Verifying privacy-type properties in a modular way

Myrto Arapinis, Vincent Cheval, and
Stéphanie Delaune

February 2012

Research report LSV-12-03

Laboratoire Spécification & Vérification

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

Verifying privacy-type properties in a modular way

Myrto Arapinis *, Vincent Cheval † and Stéphanie Delaune†
*School of Computer Science, University of Birmingham, UK
†LSV, ENS Cachan & CNRS & INRIA Saclay Île-de-France

***Abstract*—Formal methods have proved their usefulness for analysing the security of protocols. In such a setting, privacy-type security properties (*e.g.* vote-privacy, anonymity, unlinkability) that play an important role in many modern applications are formalised using a notion of equivalence.**

In this paper, we study the notion of trace equivalence and we show how to establish such an equivalence relation in a modular way. It is well-known that composition works well when the processes do not share secrets. However, there is no result allowing us to compose processes that rely on some shared secrets such as long term keys. We show that composition works even when the processes share secrets provided that they satisfy some reasonable conditions. Our composition result allows us to prove various equivalence-based properties in a modular way, and works in a quite general setting. In particular, we consider arbitrary cryptographic primitives and processes that use non-trivial else branches.

As an example, we consider the ICAO e-passport standard, and we show how the privacy guarantees of the whole application can be derived from the privacy guarantees of its sub-protocols.

I. Introduction

With the emergence of new systems and services like electronic IDs and passports, electronic payment systems and loyalty schemes, electronic tickets like the Navigo pass in Paris or the Oyster card in London, or telecommunication systems like mobile phones, new privacy and security concerns arise. Indeed, governments, financial and transport organisations, or telecommunication companies, all possess and manage important amounts of information concerning all of our everyday activities. As often reported by the media [1], [2], [3], this exposes us to a number of privacy threats. Security mechanisms should thus secure the offered services, ensuring the confidentiality of the gathered data and enhancing the privacy of users' identity and behaviour.

To this effect, many cryptographic protocols have been designed to prevent third parties from identifying messages as coming from a particular user. For example, mobile phone operators identify mobile phones using temporary identities that are periodically and securely updated to prevent mobile phones from being traceable. The electronic passports also include mechanisms that do not let the passport's chip disclose private information to external users. However, the design of protocols that meet particular security requirements is a notoriously difficult and error prone task. Indeed, numerous deployed protocols have subsequently been found to be flawed. For example, the BAC protocol of electronic passports makes it possible to recognise a previously observed passport, potentially enabling tracking passport holders [4], [5].

In this context, formal methods have proved their usefulness for precisely analysing the security guarantees provided by a protocol. Several techniques have been developed for [6], [7], and successfully applied to the analysis of cryptographic protocols [8], [5]. For example, a flaw has been discovered (see [9]) in the Single-Sign-On protocol used *e.g.* by Google Apps. It has been shown that a malicious application could very easily access to any other application (*e.g.* Gmail or Google Calendar) of their users. This flaw has been found when analysing the protocol using formal methods, abstracting messages by a term algebra and using the AVISPA platform [10]. However, existing techniques for analysing protocols with respect to privacy-type properties (*e.g.* [11], [7]), consider protocols to be executed in isolation, *i.e.* without taking into account other protocols which may be running in parallel. But in reality many applications run in parallel and the underlying protocols may interact in unexpected ways if cryptographic material is shared amongst them. This situation can arise if, for example, a user chooses the same password for two different network services, or a server uses the same key for different protocols.

Furthermore, real life protocols are usually complex and composed of several sub-protocols that rely on the same cryptographic material. For example, the UMTS standard [12], [13], [14] specifies tens of sub-protocols running in parallel in 3G mobile phone systems. And, while one may hope to automatically verify each of these sub-protocols in isolation, it is unrealistic to expect that the whole suite of protocols can be automatically checked. Indeed, due to computational constraints, existing tools and techniques do not scale up well to such large systems, and it is often the case that the sub-components have do be considered and analysed independently.

Unfortunately, security proofs of network services or protocols considered in isolation, do not carry over when they share keys or passwords. Consider for example the two naive protocols:

$$P: \ A \to S : \{A\}^r_{\mathsf{pk}(S)} \qquad Q: \ \begin{array}{l} A \to S : \{N_a\}^r_{\mathsf{pk}(S)} \\ S \to A : N_a \end{array}$$

In protocol P, the agent A simply identifies himself to the

server S by sending him his identity encrypted under S's public key (using a probabilistic encryption scheme). In protocol Q, the agent sends some fresh nonce N_a encrypted under S's public key. The server S acknowledges A's message by forwarding A's nonce. While P executed alone guarantees A's anonymity, it is not the case when the protocol Q is run in parallel. Indeed, an adversary may use Q as an oracle to decrypt any message. More realistic examples illustrating interactions between protocols can be found in *e.g.* [15].

In order to enable verification of complex real life systems, composition theorems for modular reasoning about security and privacy are therefore desirable. They may allow one to deduce security guarantees for a complex protocol, from the security guarantees of the individual sub-protocols. The goal of our paper is to study the composition of protocols with respect to privacy-type properties.

Related work: There are a number of papers studying the secure composition of security protocols in the symbolic model (*e.g.* [16], [17]) and in the computational model (*e.g.* [18], [19]). Our result clearly belongs to the first approach.

Actually, a lot of results have been established for trace-based security property, *e.g.* [16], [20], [21]. A result closely related to ours is the one of S. Ciobaca and V. Cortier [17]. Their result holds for any cryptographic primitives that can be modelled using equational theories, and their main result transforms any attack trace of the combined protocol into an attack trace of one of the individual protocols. This allows various ways of combining protocols such as sequentially or in parallel, possibly with inner replications. However, the major difference with our result is that they consider trace-based security properties, and more precisely secrecy (encoded as a reachability property).

Regarding equivalence-based properties, it has been shown that composition works for resistance against guessing attacks in the passive case without any additional hypothesis [22], and in the active case when the protocols are tagged [22], [23]. However, these composition results assume that passwords are the only shared secrets and are not well-suited to analyse privacy-type properties such as anonymity and unlinkability.

Our work is also related to those of Canetti *et al.* who, in the context of computational models, study *universal composability of protocols* [18]. This approach consists of defining for each sub-protocol an *ideal functionality* and then showing that a certain implementation securely emulates the ideal functionality. Since this initial work, the universal composability framework has been improved in several ways, *e.g.* with joint states [24], without pre-established session identifiers [19].

Our contributions: While most existing papers studying compositionality of protocols consider trace-based properties (covering confidentiality and authentication requirements), our work tackles the compositionality problem with respect to privacy-type properties which are usually expressed as equivalences between processes. Roughly, two processes P and Q are equivalent ($P \approx Q$) if no process O can observe any difference between the processes P and Q.

We identify sufficient conditions of *disjointness* under which protocols can "safely" be executed in parallel. In particular, we require protocols run in parallel not to use the same primitives. Our theorems hold for arbitrary primitives that can be modelled by a set of equations, and can thus handle composition of protocols relying on symmetric and asymmetric encryption schemes, hash functions, signatures, zero knowledge proofs, message authentication codes, designated verifier proofs, exclusive or, *etc*.

We first state a composition result that also allows the protocols considered to share the usual cryptographic primitives of symmetric and asymmetric encryption, hashing, and signing, provided that these primitives are tagged and that public and verification keys are not derivable. In this setting, we are able to establish a strong result that basically says that the disjoint scenario is equivalent to the shared one. This allows us to go back to the disjoint case (with no shared key) for which composition works unsurprisingly well.

Then, we further relax this condition. A second theorem shows that it is possible to compose protocols that share public and verification keys even if those are known by that attacker, provided that they are given to him from the beginning.

In both cases, we show that whenever processes P and Q (*resp.* P' and Q') satisfy the corresponding disjointness property, we can derive that P and Q running in parallel under the *composition context* $C[\_]$ are equivalent to P' and Q' running in parallel under the *composition context* $C'[\_]$, *i.e.*

$$C[P \mid Q] \;\approx\; C'[P' \mid Q']$$

from the equivalences $C[P] \approx C'[P']$ and $C[Q] \approx C'[Q']$. The composition context under which two processes are composed contains the shared keys possibly under some replications.

We illustrate the application of our results on a case study. We consider the protocols specified in the e-passport application [13], and show how the privacy guarantees of the whole application can be derived from the privacy guarantees of the individual e-passport protocols.

II. Models for security protocols

In this section, we introduce the cryptographic process calculus that we will use for describing protocols. This calculus is close to the applied pi calculus as defined in [25]. However, we use a slightly different syntax and we give a non-compositional semantics that is easier to manipulate than its compositional counterpart as defined in [25].

A. Messages

A protocol consists of some agents communicating on a network. The messages sent by the agents are modelled using an abstract term algebra. For this, we assume an infinite set of *names* $\mathcal{N}$ which is split into the set $\mathcal{B} = \{a, b, k, n, \ldots\}$ of names of *base type* (which are used for representing keys, nonces, ...) and the set $\mathcal{C}h = \{c, c_1, ch, ch_1, \ldots\}$ of names of *channel type* (which are used to name communication channels). We also consider a set of *variables* $\mathcal{X} = \{x, y, \ldots\}$, and a signature Σ consisting of a finite set of *function symbols*. We rely on a sort system for terms. The details of the sort system are unimportant, as long as the base type differ from the channel type. Moreover, we consider in addition the type *seed*. This is a subsort of base type, and we will assume that this set only contains atomic data, *i.e.* variables and names. As in the applied pi calculus, we suppose that function symbols only operate on and return terms of base type.

Terms are defined as names, variables, and function symbols applied to other terms. Let $\mathsf{N} \subseteq \mathcal{N}$ and $\mathsf{X} \subseteq \mathcal{X}$, the set of terms built from N and X by applying function symbols in Σ is denoted by $\mathcal{T}(\Sigma, \mathsf{N} \cup \mathsf{X})$. Of course function symbol application must respect sorts and arities. We write $fv(u)$ (resp. $fn(u)$) for the set of variables (resp. names) occurring in a term u. A term is *ground* if it does not contain any variable.

To model algebraic properties of cryptographic primitives, we define an *equational theory* by a finite set E of equations $u = v$ with $u, v \in \mathcal{T}(\Sigma, \mathcal{X})$, *i.e.* u, v do not contain names. We define $=_{\mathsf{E}}$ to be the smallest equivalence relation on terms, that contains E and that is closed under application of function symbols and substitutions of terms for variables.

Example 1: Consider the following signature Σ_0:

$$\{\mathsf{sdec}, \mathsf{senc}, \mathsf{adec}, \mathsf{aenc}, \mathsf{pk}, \langle\,\rangle, \mathsf{proj}_1, \mathsf{proj}_2, \mathsf{sign}, \mathsf{check}, \mathsf{vk}, \mathsf{h}\}$$

The function symbols sdec, senc (resp. adec and aenc) of arity 2 represent symmetric (resp. asymmetric) decryption and encryption. Pairing is modelled using a symbol of arity 2, denoted $\langle\ \rangle$, and projection functions denoted proj_1 and proj_2. We consider also signatures and hashes. A signature can be checked using check when the verification key is known and this operator also allows one to retrieve the signed message. We denote by $\mathsf{pk}(sk)$ (resp. $\mathsf{vk}(sk)$) the public key (resp. the verification key) associated to the private key sk. Moreover, we consider that the function symbols pk and vk take as argument a term of type *seed*.

Then, we consider the equational theory E_0, defined by the following equations ($i \in \{1, 2\}$):

$$\mathsf{sdec}(\mathsf{senc}(x, y), y) = x \qquad \mathsf{adec}(\mathsf{aenc}(x, \mathsf{pk}(y)), y) = x$$
$$\mathsf{proj}_i(\langle x_1, x_2\rangle) = x_i \qquad \mathsf{check}(\mathsf{sign}(x, y), \mathsf{vk}(y)) = x$$

Let $u_1 = \mathsf{senc}(\mathsf{proj}_2(\langle a, b\rangle), k)$ and $u_2 = \mathsf{senc}(b, k)$. We have that the terms u_1 and u_2 are equal modulo E_0, written $u_1 =_{\mathsf{E}_0} u_2$, while obviously the syntactic equality $u_1 = u_2$ does not hold.

B. Processes

Plain processes are built up in a similar way to plain processes in applied pi calculus. The grammar of the *plain processes* is as follows:

$$\begin{aligned} P, Q := \;& 0 \\ & P \mid Q \\ & \mathtt{new}\ n.P \\ & !P \\ & \mathtt{if}\ u_1 = u_2\ \mathtt{then}\ P\ \mathtt{else}\ Q \\ & \mathtt{in}(u, x).P \\ & \mathtt{out}(u, v).Q \end{aligned}$$

where u is a term of channel type (*i.e.* a name or a variable), u_1, u_2 are terms having the same type, x is a variable, v is a term, and n is a name. The terms u_1, u_2 and v may contain variables.

As usual, names and variables have scopes, which are delimited by restrictions and by inputs. We write $fv(P)$, $bv(P)$, $fn(P)$ and $bn(P)$ for the sets of free and bound variables, and free and bound names of a plain process P respectively.

Extended processes add a set of restricted names $\mathcal{E}$, and a sequence of messages Φ.

Definition 1: An *extended process* A is a triple $(\mathcal{E}; \mathcal{P}; \Phi)$:

- $\mathcal{E}$ is a set of names that represents the names that are restricted in $\mathcal{P}$ and Φ;
- $\mathcal{P}$ is a multiset of *plain processes* where null processes are removed and such that $fv(\mathcal{P}) = \emptyset$;
- $\Phi = \{w_1 \rhd u_1, \ldots, w_n \rhd u_n\}$ where $u_1, \ldots, u_n$ are ground terms, and $w_1, \ldots, w_n$ are variables.

We write $\mathrm{dom}(\Phi)$ the domain of Φ, *i.e.* $\mathrm{dom}(\Phi) = \{w_1, \ldots, w_n\}$. We write $fn(A)$ and $bn(A)$ for the sets of free and bound names of an extended process A. Given $A = (\mathcal{E}; \mathcal{P}; \Phi)$, we have that $fn(A) = fn(\mathcal{P}) \setminus \mathcal{E}$, and $bn(A) = bn(\mathcal{P}) \cup \mathcal{E}$.

For sake of clarity, we often omit brackets and the null process. For instance, we write k_1, $\mathtt{out}(c, u)$ instead of $\{k_1\}$ and $\{\mathtt{out}(c, u).0\}$. When there is no "else", it means "else 0"; and we sometimes write

$$\mathtt{if}\ (u_1 = u_2 \wedge u_1' = u_2')\ \mathtt{then}\ P\,\mathtt{else}\ Q$$

instead of nested conditionals. Moreover, we often write P instead of $(\emptyset; P; \emptyset)$.

Example 2: As an illustrative example, consider the process $A_i = \mathtt{new}\ sk_S.(P_i \mid Q)$ that has been informally introduced in Section I. We have that:

- $P_i = \mathtt{new}\ r.\mathtt{out}(c, \mathsf{aenc}(\langle r, id_i\rangle, \mathsf{pk}(sk_S)))$, and
- $Q = \mathtt{in}(c, x).\mathtt{out}(c, \mathsf{proj}_2(\mathsf{adec}(x, sk_S)))$.

$$(\mathcal{E}; \{\texttt{if } u = v \texttt{ then } Q_1 \texttt{ else } Q_2\} \uplus \mathcal{P}; \Phi) \xrightarrow{\tau} (\mathcal{E}; Q_1 \uplus \mathcal{P}; \Phi) \quad \text{if } u =_{\mathsf{E}} v \qquad \text{(THEN)}$$

$$(\mathcal{E}; \{\texttt{if } u = v \texttt{ then } Q_1 \texttt{ else } Q_2\} \uplus \mathcal{P}; \Phi) \xrightarrow{\tau} (\mathcal{E}; Q_2 \uplus \mathcal{P}; \Phi) \quad \text{if } u \neq_{\mathsf{E}} v \qquad \text{(ELSE)}$$

$$(\mathcal{E}; \{\texttt{out}(p,u).Q_1; \texttt{in}(p,x).Q_2\} \uplus \mathcal{P}; \Phi) \xrightarrow{\tau} (\mathcal{E}; Q_1 \uplus Q_2\{x \mapsto u\} \uplus \mathcal{P}; \Phi) \qquad \text{(COMM)}$$

$$(\mathcal{E}; \{\texttt{in}(p,x).Q\} \uplus \mathcal{P}; \Phi) \xrightarrow{\texttt{in}(p,M)} (\mathcal{E}; Q\{x \mapsto u\} \uplus \mathcal{P}; \Phi) \qquad \text{(IN)}$$
if $p \notin \mathcal{E}$, $M\Phi = u$, $fv(M) \subseteq \mathrm{dom}(\Phi)$ and $fn(M) \cap \mathcal{E} = \emptyset$

$$(\mathcal{E}; \{\texttt{out}(p,u).Q\} \uplus \mathcal{P}; \Phi) \xrightarrow{\nu w_n.\texttt{out}(p,w_n)} (\mathcal{E}; Q \uplus \mathcal{P}; \Phi \cup \{w_n \rhd u\}) \qquad \text{(OUT-T)}$$
if $p \notin \mathcal{E}$, u is a term of base type, and w_n is a variable such that $n = |\Phi| + 1$

$$(\mathcal{E}; \{\texttt{out}(p,c).Q\} \uplus \mathcal{P}; \Phi) \xrightarrow{\texttt{out}(p,c)} (\mathcal{E}; Q \uplus \mathcal{P}; \Phi) \quad \text{if } p, c \notin \mathcal{E} \qquad \text{(OUT-CH)}$$

$$(\mathcal{E}; \{\texttt{out}(p,c).Q\} \uplus \mathcal{P}; \Phi) \xrightarrow{\nu ch_n.\texttt{out}(p,ch_n)} (\mathcal{E}; (Q \uplus \mathcal{P})\{c \mapsto ch_n\}; \Phi) \qquad \text{(OPEN-CH)}$$
if $p \notin \mathcal{E}$, $c \in \mathcal{E}$, ch_n is a fresh channel name

$$(\mathcal{E}; \{\texttt{new } k.Q\} \uplus \mathcal{P}; \Phi) \xrightarrow{\tau} (\mathcal{E} \cup \{n\}; Q\{k \mapsto n\} \uplus \mathcal{P}; \Phi) \qquad \text{(NEW)}$$
if n is a fresh name with the same type as k

$$(\mathcal{E}; \{!Q\} \uplus \mathcal{P}; \Phi) \xrightarrow{\tau} (\mathcal{E}; \{!Q; Q\} \uplus \mathcal{P}; \Phi) \qquad \text{(REPL)}$$

$$(\mathcal{E}; \{P_1 \mid P_2\} \uplus \mathcal{P}; \Phi) \xrightarrow{\tau} (\mathcal{E}; \{P_1, P_2\} \uplus \mathcal{P}; \Phi) \qquad \text{(PAR)}$$

where p, c are channel names, u, v are ground terms, and x is a variable.

Figure 1. Semantics

The first component generates a fresh random number r, publishes the message $\mathsf{aenc}(\langle r, id_i\rangle, \mathsf{pk}(sk_S))$ containing its identity id_i by sending it on the public channel c. The second component receives a message on c, uses the private key sk_S to decrypt it, and sends the second part of the resulting plaintext on c.

The semantics is given by a set of labelled rules (see Figure 1) that allows one to reason about processes that interact with their environment. This defines the relation $\xrightarrow{\ell}$ where ℓ is either an input, an output, or a silent action τ. Note that the sent messages of base type are exclusively stored in the frame and not in the labels (the outputs are made by "reference").

Example 3: Let A_i be the extended process defined in Example 2. We have that:

$$\begin{aligned} &A_i \xrightarrow{\tau} \xrightarrow{\tau} \xrightarrow{\tau} \\ &\xrightarrow{\nu w_1.\texttt{out}(c,w_1)} (\{sk_S, r\}; Q; w_1 \rhd \mathsf{aenc}(\langle r, id_i\rangle, \mathsf{pk}(sk_S))) \\ &\xrightarrow{\texttt{in}(c,w_1)} (\{sk_S, r\}; \texttt{out}(c, M_i); w_1 \rhd \mathsf{aenc}(\langle r, id_i\rangle, \mathsf{pk}(sk_S))) \\ &\xrightarrow{\nu w_2.\texttt{out}(c,w_2)} (\{sk_S, r\}; 0; \Phi_i) \stackrel{\mathsf{def}}{=} A_i' \end{aligned}$$

with $M_i = \mathsf{proj}_2(\mathsf{adec}(\mathsf{aenc}(\langle r, id_i\rangle, \mathsf{pk}(sk_S)), sk_S))$ and $\Phi_i = \{w_1 \rhd \mathsf{aenc}(\langle r, id_i\rangle, \mathsf{pk}(sk_s)),\ w_2 \rhd M_i\}$. Note that $M_i =_{\mathsf{E}_0} id_i$.

The three first steps are performed using the rules NEW and PAR. Then, we used the rules OUT-T and IN. We denote by A_i' the resulting extended process.

Notations: Let $\mathcal{A}$ be the alphabet of actions (in our case this alphabet is infinite and contains the special symbol τ). For every $w \in \mathcal{A}^*$, the relation $\xrightarrow{w}$ on processes is defined in the usual way. For $s \in (\mathcal{A} \setminus \{\tau\})^*$, the relation $\stackrel{s}{\Rightarrow}$ on processes is defined by: $A \stackrel{s}{\Rightarrow} B$ if, and only if there exists $w \in \mathcal{A}^*$ such that $A \xrightarrow{w} B$ and s is obtained by erasing all occurrences of τ.

III. FORMALISING PRIVACY-TYPE SECURITY PROPERTIES

Many interesting security properties, in particular privacy-type properties such as those studied in [26], [5], [27], are formalised using behavioural equivalence. We will review some of them in Section III-B using the notion of trace equivalence.

A. Trace equivalence

Before defining trace equivalence, we introduce the notion of *static equivalence* that compares sequences of messages, a notion of intruder's knowledge that has been extensively studied (*e.g.* [28]).

To represent the knowledge of an attacker (who may have observed a sequence of messages $u_1, \ldots, u_n$, we use the concept of frame. A frame $\phi = \texttt{new } \mathcal{E}.\Phi$ consists of a finite set $\mathcal{E}$ of restricted names (those unknown to the attacker), and a substitution Φ of the form:

$$\{w_1 \rhd u_1, \ldots, w_n \rhd u_n\} \text{ with } \mathrm{dom}(\Phi) = \{w_1, \ldots, w_n\}$$

The variables enable us to refer to each u_i and we always assume that the terms u_i are ground. The names $\mathcal{E}$ are bound

in ϕ and can be renamed. Moreover names that do not appear in Φ can be added or removed from $\mathcal{E}$. In particular, we can always assume that two frames share the same set of restricted names.

Two frames are considered equivalent when the attacker cannot detect the difference between the two situations they represent, that is, his ability to distinguish whether two recipes M and N produce the same term does not depend on the frame.

Definition 2: We say that two frames $\phi_1 = \mathtt{new}\,\mathcal{E}.\Phi_1$ and $\phi_2 = \mathtt{new}\,\mathcal{E}.\Phi_2$ are *statically equivalent*, $\phi_1 \sim \phi_2$, when $\mathrm{dom}(\Phi_1) = \mathrm{dom}(\Phi_2)$, and for all terms M, N such that $fn(M,N) \cap \mathcal{E} = \emptyset$, we have that $M\Phi_1 =_{\mathsf{E}} N\Phi_1$, if and only if, $M\Phi_2 =_{\mathsf{E}} N\Phi_2$.

Example 4: Let A'_1 (resp. A'_2) be the extended process described in Example 3 and ϕ_1 (resp. ϕ_2) be its associated frame, *i.e.* $\phi_i = \mathtt{new}\ \{sk_S, r\}.\Phi_i$ with $i \in \{1,2\}$. We have that $\phi_1 \not\sim \phi_2$. Indeed, the test $w_2 \stackrel{?}{=} id_1$ can be used to distinguish the two frames. The test holds in ϕ_1 since $w_2\Phi_1 = M_1 =_{\mathsf{E}_0} id_1$, whereas it does not hold in ϕ_2 since $w_2\Phi_2 = M_2 =_{\mathsf{E}_0} id_2 \neq_{\mathsf{E}_0} id_1$. However, we have that:

$$\begin{array}{c}\mathtt{new}\ \{sk_s, r\}.\{w_1 \rhd \mathsf{aenc}(\langle r, id_1\rangle, \mathsf{pk}(sk_S))\}\\ \sim\\ \mathtt{new}\ \{sk_S, r\}.\{w_1 \rhd \mathsf{aenc}(\langle r, id_2\rangle, \mathsf{pk}(sk_S))\}.\end{array}$$

For every extended process $A = (\mathcal{E}; \mathcal{P}; \Phi)$, we define its set of traces, each trace consisting in a sequence of actions together with the sequence of sent messages:

$$\mathsf{trace}(A) = \{(\mathsf{tr}, \mathtt{new}\ \mathcal{E}'.\Phi') \mid A \stackrel{\mathsf{tr}}{\Rightarrow} (\mathcal{E}'; \mathcal{P}'; \Phi') \text{ for some process } (\mathcal{E}'; \mathcal{P}'; \Phi')\}.$$

Two processes are trace equivalent if, whatever the messages they received (built upon previously sent messages), the resulting sequences of messages are in static equivalence.

Definition 3: Let A and B be two extended processes, $A \sqsubseteq B$ if for every $(\mathsf{tr}, \phi) \in \mathsf{trace}(A)$ such that $bn(\mathsf{tr}) \cap fn(B) = \emptyset$, there exists $(\mathsf{tr}', \phi') \in \mathsf{trace}(B)$ such that $\mathsf{tr} = \mathsf{tr}'$ and $\phi \sim \phi'$. Two closed extended processes A and B are *trace equivalent*, denoted by $A \approx B$, if $A \sqsubseteq B$ and $B \sqsubseteq A$.

Example 5: Consider the following trace:

$$\mathsf{tr} = \nu w_1.\mathtt{out}(c, w_1) \cdot \mathtt{in}(c, w_1) \cdot \nu w_2.\mathtt{out}(c, w_2).$$

We have that $(\mathsf{tr}, \phi_1) \in \mathsf{trace}(A_1)$, and the only trace $(\mathsf{tr}', \phi') \in \mathsf{trace}(A_2)$ that satisfies $\mathsf{tr} = \mathsf{tr}'$ leads to the frame ϕ_2 for which we have seen that $\phi_1 \not\sim \phi_2$ (see Example 4). This allows us to conclude that $A_1 \not\approx A_2$.

B. Some examples

The definitions we present here are informal ones, and we refer the reader to [5] for detailed formal definitions. In Section VI, we will illustrate these definitions through the e-passport application.

Strong anonymity: Anonymity is informally defined by the ISO/IEC standard 15408 [29] as the property ensuring that *a user may use a service or a resource without disclosing the user's identity*. Formally, strong anonymity has been defined to hold [5] when an outside observer cannot tell the difference between a system in which the user with a public known identity id_0 executes the analysed protocol, from the system where id_0 is not present at all.

Following this formal definition of anonymity, the protocol introduced in Section I considered in isolation, *i.e.* $P = \mathtt{new}\ r.\mathtt{out}(c, \mathsf{aenc}(\langle r, id\rangle, \mathsf{pk}(sk_S)))$, is said to satisfy strong anonymity if the following equivalence holds:

$$\begin{array}{c}\mathtt{new}\ sk_S.\ ((!\mathtt{new}\ id.\ !P) \mid !P\{id_0/id\})\\ \approx\\ \mathtt{new}\ sk_S.(!\mathtt{new}\ id.\ !P)\end{array}$$

In other words, anonymity is satisfied if an observer cannot tell if the user id_0 (known to the attacker) has been executing the protocol P or not.

Strong unlinkability: Unlinkability is informally defined by the ISO/IEC standard 15408 [29] as the property ensuring that *a user may make multiple uses of a service or a resource without others being able to link these uses together*. Formally, strong unlinkability has been defined to hold [5] when a system in which the analysed protocol can be executed by each user multiple times looks the same to an outside observer that the system in which the analysed protocol can be executed by each user at most once.

Again, we can formalise this property for the protocol P when considered in isolation using an equivalence:

$$\mathtt{new}\ sk_S.\ (!\mathtt{new}\ id.\ !P) \ \approx\ \mathtt{new}\ sk_S.(!\mathtt{new}\ id.\ P)$$

In other words, unlinkability is satisfied if an observer cannot tell if the users can execute multiple or at most once the protocol P.

IV. Composition result: a simple setting

Even if a protocol is secure for an unbounded number of sessions, there is no guarantee if the protocol is executed in an environment where other protocols sharing some common keys are executed. The interaction with the other protocols may dramatically damage the security of the former protocol. This is a well-known fact that has been already observed for trace-based security properties *e.g.* [16], [17], and that remains true for privacy-type properties.

An attacker may take advantage of a protocol Q to break anonymity of another protocol P that has been proved secure

in isolation. This can happen for instance if the security of P relies on the secrecy of a particular shared key that is revealed by the protocol Q.

A. Sharing primitives

Actually, even if shared keys are not revealed, the interaction of two protocols using common primitives may compromise their security.

Example 6: Consider the processes P_i with $i \in \{1, 2\}$ as defined in Example 2. The equivalence expressing the anonymity of P (for one session) holds. We have that $\texttt{new}\ sk_S.P_1 \approx \texttt{new}\ sk_S.P_2$ whereas the equivalence expressing the anonymity of P in presence of Q does not hold anymore. We have that:

$$\texttt{new}\ sk_S.\big(P_1 \mid Q\big) \not\approx \texttt{new}\ sk_S.\big(P_2 \mid Q\big)$$

Intuitively, the security of P is ensured by the fact that its identity id is encrypted using the public key $\mathsf{pk}(sk_S)$ whose associated private key sk_S is kept secret. However, Q can be used as an oracle to decrypt a ciphertext that comes from the process P, and thus Q can be used to reveal the identity hidden in the ciphertext.

To avoid a ciphertext from a process to be decrypted by another one, we can consider processes that use disjoint primitives. However, this is an unnecessarily restrictive condition. So, we consider protocols that may share some cryptographic primitives provided they are tagged.

Tagging is a syntactic transformation that consists in assigning to each protocol an identifier (*e.g.* the protocol's name) that should appear in any encrypted message. Many relevant equational theories are not so easy to tag (*e.g.* exclusive or). So, we consider the fix common equational theory (Σ_0, E_0) defined in Example 1, and we explain how to transform any process built on a signature Σ (possibly larger that Σ_0) into a well-tagged process. For this, we define $\Sigma_{\mathsf{tag}_c} = \{\mathsf{tag}_c, \mathsf{untag}_c\}$ where tag_c and untag_c are two function symbols of arity 1 that we will use for tagging. The role of the tag_c function is to tag its argument with the tag c. The role of the untag_c function is to remove the tag. To model this interaction between tag_c and untag_c, we consider the equational theory:

$$\mathsf{E}_{\mathsf{tag}_c} = \{\mathsf{untag}_c(\mathsf{tag}_c(x)) = x\}.$$

For our composition result, we will assume that the processes P_A and P_B that we want to compose are built on $(\Sigma_a \cup \Sigma_0, \mathsf{E}_a \cup \mathsf{E}_0)$ and $(\Sigma_b \cup \Sigma_0, \mathsf{E}_b \cup \mathsf{E}_0)$, where (Σ_a, E_a), (Σ_b, E_b) and (Σ_0, E_0) are disjoint signatures that are also disjoint from $(\Sigma_{\mathsf{tag}_a}, \mathsf{E}_{\mathsf{tag}_a})$ and $(\Sigma_{\mathsf{tag}_b}, \mathsf{E}_{\mathsf{tag}_b})$. The signature Σ_0 contains the function symbols that can be used by the two processes and that have to be tagged. We denote by $\Sigma_c^+ = \Sigma_c \cup \Sigma_{\mathsf{tag}_c}$ and $\mathsf{E}_c^+ = \mathsf{E}_c \cup \mathsf{E}_{\mathsf{tag}_c}$ with $c \in \{a, b\}$.

Definition 4: Let u be a term built on $\Sigma_c \cup \Sigma_0$ ($c \in \{a, b\}$). The c-tagged version of u, denoted $[u]_c$ is defined as follows:

$$\begin{array}{rcl}
[\mathsf{senc}(u,v)]_c & \stackrel{\mathsf{def}}{=} & \mathsf{senc}(\mathsf{tag}_c([u]_c), [v]_c) \\
[\mathsf{aenc}(u,v)]_c & \stackrel{\mathsf{def}}{=} & \mathsf{aenc}(\mathsf{tag}_c([u]_c), [v]_c) \\
[\mathsf{sign}(u,v)]_c & \stackrel{\mathsf{def}}{=} & \mathsf{sign}(\mathsf{tag}_c([u]_c), [v]_c) \\
[\mathsf{h}(u)]_c & \stackrel{\mathsf{def}}{=} & \mathsf{h}(\mathsf{tag}_c([u]_c)) \\
[\mathsf{sdec}(u,v)]_c & \stackrel{\mathsf{def}}{=} & \mathsf{untag}_c(\mathsf{sdec}([u]_c, [v]_c)) \\
[\mathsf{adec}(u,v)]_c & \stackrel{\mathsf{def}}{=} & \mathsf{untag}_c(\mathsf{adec}([u]_c, [v]_c)) \\
[\mathsf{check}(u,v)]_c & \stackrel{\mathsf{def}}{=} & \mathsf{untag}_c(\mathsf{check}([u]_c, [v]_c)) \\
[\mathsf{f}(u_1, \ldots, u_n)]_c & \stackrel{\mathsf{def}}{=} & \mathsf{f}([u_1]_c, \ldots, [u_n]_c) \text{ otherwise.}
\end{array}$$

Note that we do not tag the pairing function symbol (this is actually useless), and we do not tag the pk and vk function symbols. Actually, tagging pk and vk would greatly help us to establish our results and would also avoid us to introduce some additional assumptions, but this would lead us to consider an unrealistic modelling for asymmetric keys. Some of the difficulties encountered with asymmetric keys will be discussed in Section V.

Example 7: Consider $u_i = \mathsf{aenc}(\langle r, id_i\rangle, \mathsf{pk}(sk_S))$ with $i \in \{1, 2\}$ and $v = \mathsf{proj}_2(\mathsf{adec}(x, sk_S))$. We have that $[u_i]_a = \mathsf{aenc}(\mathsf{tag}_a(\langle r, id_i\rangle), \mathsf{pk}(sk_S))$, whereas $[v]_b = \mathsf{proj}_2(\mathsf{untag}_b(\mathsf{adec}(x, sk_S)))$.

Before extending the notion of tagging to processes, we have to express the tests that are performed by an agent when he receives a message that is supposed to be tagged. This is the purpose of $\mathsf{test}_c(u)$ that represents the tests which ensure that every projection and every untagging performed by an agent during the computation of u is successful.

Definition 5: Let u be a term built on $\Sigma_c^+ \cup \Sigma_0$ with $c \in \{a, b\}$. We define $\mathsf{test}_c(u)$ as follows:

$$\begin{array}{rcl}
\mathsf{test}_c(u) & \stackrel{\mathsf{def}}{=} & \mathsf{test}_c(u_1) \wedge \mathsf{test}_c(u_2) \wedge \mathsf{tag}_c(\mathsf{untag}_c(u)) = u \\
& & \text{when } u = \mathsf{g}(u_1, u_2) \text{ with } \mathsf{g} \in \{\mathsf{sdec}, \mathsf{adec}, \mathsf{check}\} \\
\mathsf{test}_c(u) & \stackrel{\mathsf{def}}{=} & \mathsf{test}_c(u_1) \wedge u_1 = \langle \mathsf{proj}_1(u_1), \mathsf{proj}_2(u_1)\rangle \\
& & \text{when } u = \mathsf{proj}_i(u_1) \text{ with } i \in \{1, 2\} \\
\mathsf{test}_c(u) & \stackrel{\mathsf{def}}{=} & \mathsf{true} \qquad \text{when } u \text{ is a name or a variable} \\
\mathsf{test}_c(u) & \stackrel{\mathsf{def}}{=} & \mathsf{test}_c(u_1) \wedge \ldots \wedge \mathsf{test}_c(u_n) \qquad \text{otherwise.}
\end{array}$$

Example 8: Again, consider $u_i = \mathsf{aenc}(\langle r, id_i\rangle, \mathsf{pk}(sk_S))$ with $i \in \{1, 2\}$ and $v = \mathsf{proj}_2(\mathsf{adec}(x, sk_S))$. We have that:

$$\begin{array}{l}
\mathsf{test}_a([u_i]_a) = \mathsf{true} \\
\mathsf{test}_b([v]_b) = \mathsf{tag}_b(\mathsf{untag}_b(\mathsf{adec}(x, sk_S))) = \mathsf{adec}(x, sk_S) \\
\qquad\qquad \wedge\ \mathsf{proj}_1(v'), \mathsf{proj}_2(v')\rangle = v' \\
\qquad\qquad\qquad \text{where } v' = \mathsf{untag}_b(\mathsf{adec}(x, sk_S)).
\end{array}$$

Let $A = (\mathcal{E}; \mathcal{P}; \Phi)$ be a process built on $\Sigma_c \cup \Sigma_0$ with $c \in \{a, b\}$ such that $\mathcal{P} = \{P_1, \ldots, P_\ell\}$, and $\Phi = \{w_1 \rhd$

$u_1, \ldots, w_n \rhd u_n\}$. The c-tagged version of the process A, denoted $[A]_c$, is the process $(\mathcal{E}; [\mathcal{P}]_c; [\Phi]_c)$ where $[\mathcal{P}]_c = \{[P_1]_c, \ldots, [P_\ell]_c\}$, and

$$[\Phi]_c = \{w_1 \rhd [u_1]_c, \ldots, w_n \rhd [u_n]_c\}.$$

For plain processes, the transformation $[P]_c$ is defined as follows:

$$[0]_c \stackrel{\mathsf{def}}{=} 0 \qquad [!P]_c \stackrel{\mathsf{def}}{=} ![P]_c \qquad [\mathtt{new}\ k.P]_c \stackrel{\mathsf{def}}{=} \mathtt{new}\ k.[P]_c$$

$$[P \mid Q]_c \stackrel{\mathsf{def}}{=} [P]_c \mid [Q]_c \qquad [\mathtt{in}(u,x).P]_c \stackrel{\mathsf{def}}{=} \mathtt{in}(u,x).[P]_c$$

$$[\mathtt{out}(u,v).Q]_c \stackrel{\mathsf{def}}{=} \mathtt{if}\ \mathsf{test}_c([v]_c)\ \mathtt{then}\ \mathtt{out}(u,[v]_c).[Q]_c$$

$$[\mathtt{if}\ u_1 = u_2\ \mathtt{then}\ P\ \mathtt{else}\ Q]_c \stackrel{\mathsf{def}}{=} \mathtt{if}\ \varphi\ \mathtt{then}\ (\mathtt{if}\ [u_1]_c = [u_2]_c\ \mathtt{then}\ [P]_c\ \mathtt{else}\ [Q]_c)\ \mathtt{else}\ 0$$

where $\varphi = \mathsf{test}_c([u_1]_c) \wedge \mathsf{test}_c([u_2]_c)$

Roughly, instead of simply outputting a term v, a process will first performed some tests to check that the term is correctly tagged and he will output its c-tagged version $[v]_c$. For a conditional, the process will first check that the terms u_1 and u_2 are correctly tagged before checking that the test is satisfied.

Example 9: Consider the processes P_i and Q defined in Example 2.

$$[P_i]_a = \mathtt{new}\ r.\ \mathtt{out}(c, \mathsf{aenc}(\mathsf{tag}_a(\langle r, id_i\rangle), \mathsf{pk}(sk_S)))$$
$$[Q]_b = \mathtt{in}(c,x).\mathtt{if}\ \mathsf{test}_b([v]_b)\ \mathtt{then}\ \mathtt{out}(c,[v]_b)$$

where $[v]_b$ (resp. $\mathsf{test}_b([v]_b)$) have been defined in Example 7 (resp. Example 8).

Note that the tag will prevent the process Q to decrypt the ciphertext that has been output by P_i. Thus, the equivalence expressing the anonymity of $[P_i]_a$ now holds even in the presence of $[Q]_b$. We have that:

$$\mathtt{new}\ sk_S.\big([P_1]_a \mid [Q]_b\big) \approx \mathtt{new}\ sk_S.\big([P_2]_a \mid [Q]_b\big).$$

This a is non-trivial equivalence that can actually be derived from the equivalence $\mathtt{new}\ sk_S.[P_1]_a \approx \mathtt{new}\ sk_S.[P_2]_a$ using our composition result (Corollary 1).

B. Composition context

As already mentioned, we want to establish a composition result between processes that share the signature (Σ_0, E_0) and also share some keys. Thus, we introduce the notion of *composition context* that will help us to describe under which keys the composition has to be done. Note that a composition context may contain several holes, parallel operators, and nested replications. This is needed to express privacy-type properties as those described in Section III-B.

Definition 6: A *composition context* C is defined by the following grammar where n is a name of base type.

$$C, C_1, C_2 := \_ \mid \mathtt{new}\ n.\ C \mid !C \mid C_1|C_2$$

We only allow names of base type (typically keys) to be shared between processes through the composition context. In particular, they are not allowed to share a private channel even if each process can used its own private channels to communicate internally. We also suppose w.l.o.g. that names occurring in C are distinct. A composition context may contain several holes. We can index them to avoid confusion. We write $C[P_1, \ldots, P_\ell]$ (or shortly $C[\overline{P}]$) the process obtained by filling the i^{th} hole with the process P_i (or the i^{th} process of the sequence $\overline{P}$). We will also use $\overline{P} \mid \overline{Q}$ to represent the sequence of processes obtained by putting in parallel the processes of the sequences $\overline{P}$ and $\overline{Q}$ componentwise.

Example 10: In Section III-B, we have seen that unlinkability of P can be modelled using the equivalence:

$$\mathtt{new}\ sk_S.\big(!\mathtt{new}\ id.!P\big) \approx \mathtt{new}\ sk_S.\big(!\mathtt{new}\ id.P\big).$$

The composition contexts used to express this property are:

- $C[\_] = \mathtt{new}\ sk_S.\big(!\mathtt{new}\ id.\ !\ \_\big)$, and
- $C'[\_] = \mathtt{new}\ sk_S.\big(!\mathtt{new}\ id.\ \_\big)$.

Since the name id does not occur in the process Q (see Example 2), it is quite easy to see that $C[Q] \approx C'[Q]$. Unlinkability of P in presence of the process Q will be modelled as $C[P \mid Q] \approx C'[P \mid Q]$, which is equivalent to:

$$\mathtt{new}\ sk_S.!\big((\mathtt{new}\ id.!P) \mid Q\big) \approx \mathtt{new}\ sk_S.!\big((\mathtt{new}\ id.P) \mid Q\big)$$

Note that in a composition context a replication may occur in the scope of some restrictions and this is needed to express many interesting privacy-type properties. Considering composition in a simpler setting where only a bounded number of keys $\tilde{k}$ are shared (as done in *e.g.* [30]), would not allow us to establish unlinkability in a modular way, but only some results of the form:

$$\mathtt{new}\ \tilde{k}.P_1 \approx \mathtt{new}\ \tilde{k}.P_2 \ \Rightarrow\ \mathtt{new}\ \tilde{k}.\ (P_1 \mid Q) \approx \mathtt{new}\ \tilde{k}.\ (P_2 \mid Q)$$

assuming that processes P_1, P_2, and Q satisfy some additional conditions.

Now, we have introduce composition under replication, but have to formalise the notion of revealing a shared key. The names that occur in the composition context represent the names that are shared between the two processes that we want to compose. Since those names may occur under a replication, we have to consider renaming and formalise this notion of revealing accordingly.

Definition 7: Let C be a composition context, A be an extended process of the form $(\mathcal{E}; C[P_1, \ldots, P_\ell]; \Phi)$, and $key \in \{n, \mathsf{pk}(n), \mathsf{vk}(n) \mid n \in \mathcal{E} \text{ or } n \text{ occurs in } C\}$. We say that *the extended process A reveals the shared key key* when:

Either $fn(key) \in \mathcal{E}$, and

- $A \stackrel{w}{\Rightarrow} (\mathcal{E}'; \mathcal{P}'; \Phi')$ for some $(\mathcal{E}'; \mathcal{P}'; \Phi')$; and

- $M\Phi' =_{\mathsf{E}} key$ for some M such that $fv(M) \subseteq \mathrm{dom}(\Phi')$ and $fn(M) \cap \mathcal{E}' = \emptyset$.

Or, we have that $fn(key)$ occurs in C, the $i_0{}^{\text{th}}$ hole is in the scope of $\texttt{new}\ fn(key)$, and

- $(\mathcal{E} \cup \{s\}; C[P_1^+, \ldots, P_\ell^+]; \Phi) \stackrel{w}{\Rightarrow} (\mathcal{E}'; \mathcal{P}'; \Phi')$ with $P_{i_0}^+ \stackrel{\mathsf{def}}{=} P_{i_0} \mid \mathtt{in}(c,x).\mathtt{if}\ x = key\ \mathtt{then}\,\mathtt{out}(c,s)$ and $P_i^+ \stackrel{\mathsf{def}}{=} P_i$ if $i \neq i_0$; and
- $M\Phi' =_{\mathsf{E}} s$ for some M such that $fv(M) \subseteq \mathrm{dom}(\Phi')$ and $fn(M) \cap \mathcal{E}' = \emptyset$.

Example 11: Consider the composition context $C[\_] = \texttt{new}\ sk_S.\ \_$. The extended process $(\emptyset; C[P]; \emptyset)$ with P as described in Section III-B does not reveal the keys sk_S, $\mathsf{pk}(sk_S)$ and $\mathsf{vk}(sk_S)$. Indeed, let $key \in \{sk_S, \mathsf{pk}(sk_S), \mathsf{vk}(sk_S)\}$, we have that

$$(\{s\}; C[P \mid \mathtt{in}(c,x).\mathtt{if}\ x = key\ \mathtt{then}\,\mathtt{out}(c,s)]; \emptyset)$$

can not reached a configuration from which s will be derivable by the attacker.

C. Going back to the disjoint case

It is well-know that parallel composition works when processes do not share any secret, the so-called disjoint case. A first idea to establish a composition result is to see under which conditions we can go back to the disjoint case. In this section, we will see that this is indeed possible provided that processes are tagged and only share some keys that will never be revealed.

Theorem 1: Let C be a composition context, and $\overline{P_A}$ (resp. $\overline{P_B}$) be two sequences of plain processes built on the signature $\Sigma_a \cup \Sigma_0$ (resp $\Sigma_b \cup \Sigma_0$). Assume that $C[[\overline{P_A}]_a]$ and $C[[\overline{P_B}]_b]$ do not reveal any key in $\{k, \mathsf{pk}(k), \mathsf{vk}(k) \mid k \text{ occurs in } C\}$. We have that:

$$C[[\overline{P_A}]_a \mid [\overline{P_B}]_b] \approx C[[\overline{P_A}]_a] \mid C[[\overline{P_B}]_b].$$

Proof: (*sketch*) Consider $S = (\emptyset; C[[\overline{P_A}]_a \mid [\overline{P_B}]_b]; \emptyset)$ and $D = (\emptyset; C[[\overline{P_A}]_a] \mid C[[\overline{P_B}]_b]; \emptyset)$. Actually, we can show that any trace $(\mathsf{tr}, \phi_D) \in \mathsf{trace}(D)$ can be mapped to a trace $(\mathsf{tr}, \phi_S) \in \mathsf{trace}(S)$ such that $\phi_D \sim \phi_S$ and conversely. Note that even if the resulting frames ϕ_S and ϕ_D are not syntactically equal, we can show that they are in static equivalence and the computation performed by the attacker in both executions are exactly the same, namely tr.

For this, we consider the transformation δ ($c \in \{a, b\}$) on terms whose purpose is to replace the occurrences of the shared keys that are used in $\overline{P_B}$ by some fresh names in order to ensure disjointness. However, we do not want to replace any occurrence of a shared key. For instance, assume that the following term $u = \mathsf{senc}(\mathsf{tag}_b(\mathsf{senc}(\mathsf{tag}_a(n_a), k)), k)$ has been output by the process $P_B = \mathtt{in}(c,x).\mathtt{out}(c, \mathsf{senc}(\mathsf{tag}_b(x), k))$. The purpose of δ is to replace the occurrences of the shared k that "come from P_B" by a fresh key k'. Actually, we have that:

$$\delta(u) = \mathsf{senc}(\mathsf{tag}_b(\mathsf{senc}(\mathsf{tag}_a(n_a), k)), k').$$

Then the proof can go through thanks to some nice properties that are enjoyed by this transformation δ. In particular, we have that:

- this transformation preserves the equality tests performed by each process: "$\delta(u) = \delta(v) \Leftrightarrow u = v$".
- this transformation preserves deducibility in the sense that for any message u that the attacker can obtained from ϕ_S, we can show that its counterpart $\delta(u)$ can be obtained using "$\delta(\phi_S) = \phi_D$" using the same recipe (and conversely).

■

This result as well as the way we proceed to prove it are close to the one proved in [17]. However, we generalise it in several ways. First, we combine the results of [17] so that we are able to deal with disjoint equational theories together with a common equational theory. Moreover, for the common theory, we consider also pairing and asymmetric primitives. Due to the way tagging is performed, the asymmetric primitives add some difficulties. Second, since we want a composition result for trace equivalence, we have to map any trace of D to a trace of S (and conversely), and we have also to ensure that the resulting sequence of messages are in static equivalence. Third, we consider a process algebra that allows us to express disequality tests (*i.e.* non-trivial else branches).

Note that, we have to ensure that shared keys are never revealed. This is needed for symmetric keys, but as mentioned in the hypothesis of the proposition, this is also required for public keys and verification keys. As we will see in Example 14, this hypothesis is necessary for this result to hold, but we will show how to relax it and still get a composition result (see Section V).

D. A first composition result

The result stated in Theorem 1 allows us to go back to the disjoint case for which composition works quite well. Hence, as a corollary, we are now able to state our first composition result.

Corollary 1: Let C and C' be two composition contexts. Let $\overline{P_A}, \overline{P'_A}$ (resp. $\overline{P_B}, \overline{P'_B}$) be two sequences of plain processes built on the signature $\Sigma_a \cup \Sigma_0$ (resp. $\Sigma_b \cup \Sigma_0$). Assume that $C[[\overline{P_A}]_a]$ and $C[[\overline{P_B}]_b]$ (resp. $C'[[\overline{P'_A}]_a]$ and $C'[[\overline{P'_B}]_b]$) do not reveal any shared key in $\{k, \mathsf{pk}(k), \mathsf{vk}(k) \mid k \text{ occurs in } C\}$ (resp. $\{k, \mathsf{pk}(k), \mathsf{vk}(k) \mid k \text{ occurs in } C'\}$). We have that:

$$\frac{C[[\overline{P_A}]_a] \;\approx\; C'[[\overline{P'_A}]_a] \quad C[[\overline{P_B}]_b] \;\approx\; C'[[\overline{P'_B}]_b]}{C[[\overline{P_A}]_a \mid [\overline{P_B}]_b] \;\approx\; C'[[\overline{P'_A}]_a \mid [\overline{P'_B}]_b]}$$

Proof: (sketch) This composition result is proved in three main steps.

1) We have that the equivalences $C[[\overline{P_A}]_a] \approx C'[[\overline{P'_A}]_a]$ and $C[[\overline{P_B}]_b] \approx C'[[\overline{P'_B}]_b]$ hold on the signatures $(\Sigma_a^+ \cup \Sigma_0, \mathsf{E}_a^+ \cup \mathsf{E}_0)$ and $(\Sigma_b^+ \cup \Sigma_0, \mathsf{E}_b^+ \cup \mathsf{E}_0)$ respectively. It is relatively easy to show that the same equivalences also hold on the augmented signature $(\Sigma_a^+ \cup \Sigma_b^+ \cup \Sigma_0, \mathsf{E}_a^+ \cup \mathsf{E}_b^+ \cup \mathsf{E}_0)$.
2) Then, relying on these two equivalences, we can show that:
$$C[[\overline{P_A}]_a] \mid C[[\overline{P_B}]_b] \;\approx\; C'[[\overline{P'_A}]_a] \mid C'[[\overline{P'_B}]_b].$$
This corresponds to composition in the disjoint case (no shared key). This is a well-know fact that actually holds in many cryptographic calculus.
3) Then, we apply Theorem 1 on both sides of the equivalence, and we obtain the expected result:
$$C[[\overline{P_A}]_a \mid [\overline{P_B}]_b] \;\approx\; C'[[\overline{P'_A}]_a \mid [\overline{P'_B}]_b].$$

■

V. Composition in presence of processes that reveal shared keys

In the previous section, we presented a first composition result. However, this result does not hold as soon as some shared keys are revealed: such a key can be a symmetric shared key, the private part of an asymmetric key pair, but also the public part of an asymmetric key pair. In this section, we will see that we can relax this condition by allowing shared keys to be revealed from the beginning.

A. Some additional difficulties

First, as shown by the example below, we do not want public keys to be revealed (for the first time) during the execution of the protocol.

Example 12: We consider a slightly different version of the process P_i introduced in Example 2. Basically, we remove the random r inside the encryption and we consider its well-tagged version. We consider the following processes:

$$[P'_i]_a \stackrel{\mathsf{def}}{=} \mathtt{out}(c, \mathsf{aenc}(\mathsf{tag}_a(id_i), \mathsf{pk}(sk_S))) \quad i \in \{1, 2\}$$

Consider the composition context $C[\_] = \mathtt{new}\ sk_S.\ \_$. Note that, the equivalence $C[[P'_1]_a] \approx C[[P'_2]_a]$ still holds in this setting. Assume now that $[P'_i]_a$ is executed in presence of the well-tagged process $Q^{\mathsf{pk}} = \mathtt{out}(c, \mathsf{pk}(sk_S))$. Clearly, the equivalence expressing the anonymity of $[P'_i]_a$ does not hold anymore. We have that:

$$C[[P'_1]_a \mid Q^{\mathsf{pk}}] \not\approx C[[P'_2]_a \mid Q^{\mathsf{pk}}].$$

Actually, the knowledge of $\mathsf{pk}(sk_S)$ will allow the attacker to distinguish the message emitted by $[P'_1]_a$ from the one emitted by $[P'_2]_a$.

To avoid the problem mentioned above, we will assume that shared keys that are revealed have to be revealed from the very beginning. This hypothesis seems indeed reasonable since the purpose of a public key is in general to be disclosed at the beginning, or eventually never revealed to an outsider.

Note that the previous example is not a counter-example anymore if we analyse the equivalence expressing the anonymity of $[P'_i]_a$ assuming that $\mathsf{pk}(sk_S)$ is known by the attacker from the beginning. The fact that $\mathsf{pk}(sk_S)$ is revealed during the execution of Q^{pk} will not give any additional power to the attacker.

Example 13: We consider again the process P_i as presented in Example 2 with an additional output to reveal the public key $\mathsf{pk}(sk_S)$ at the very beginning. Basically, we consider the well-tagged process $P''_i \stackrel{\mathsf{def}}{=} \mathtt{out}(c, \mathsf{pk}(sk_S)).[P_i]_a$.

We have that $C[P''_1] \approx C[P''_2]$ with $C[\_] = \mathtt{new}\ sk_S.\_$. Now, the presence of Q^{pk} will not prevent this equivalence to hold. Indeed, we have that:

$$C[P''_1 \mid Q^{\mathsf{pk}}] \approx C[P''_2 \mid Q^{\mathsf{pk}}].$$

This hypothesis that states that shared keys are either known from the beginning or never revealed during the execution of the protocol is reasonable, and seems to be sufficient to establish a composition result. However, this complicates a bit the setting. In particular, as illustrated in Example 14, there is no hope to obtain a result as the one stated in Theorem 1. The situation where the processes share some keys is not equivalent in this setting to the situation where the processes do not share any key.

Example 14: Consider the processes P''_i and Q^{pk} used in Example 13. We have seen that composition works under the composition context $C = \mathtt{new}\ sk_S.\_$. However, we have that ($i \in \{1, 2\}$):

$$C[P''_i \mid Q^{\mathsf{pk}}] \;\not\approx\; C[P''_i] \mid C[Q^{\mathsf{pk}}].$$

Indeed, on the left-hand side, the same public-key will be output twice whereas the process on the right-hand side will emit two different public keys. The attacker will observe such a difference. The strong result stated in Theorem 1 allowing us to easily make the ling between the joint state case and the disjoint case does not hold anymore.

The problems encountered for composing processes that reveal shared keys are due to the fact that we do not want to tag the function symbols pk and vk that are used to model asymmetric keys: such a tagging scheme would lead us to an unrealistic modelling of asymmetric keys.

B. Composition result

We now consider public keys and verifications keys that can be made public from the beginning through an initial frame Φ_0 that will represent the initial knowledge of the attacker. As illustrated in Section V-A, we cannot rely on Theorem 1 anymore to establish our composition result. We will still go back to the disjoint case but we have to explain how a trace corresponding to the situation where processes share some keys is transformed and mapped to a trace that models the disjoint case. We cannot simply consider the identity transformation as it was done to establish the previous result. The sets of traces issued from both situations are not the same anymore.

Theorem 2: Let $\overline{P_A}, \overline{P'_A}$ (resp. $\overline{P_B}, \overline{P'_B}$) be two sequences of plain processes built $\Sigma_a \cup \Sigma_0$ (resp. $\Sigma_b \cup \Sigma_0$). Let $\mathcal{K}_0$ be a finite set of names of base type, and C and C' be two composition contexts. Let $\Phi_0 = \{w_1 \rhd \mathsf{f}_1(k_1), \ldots, w_n \rhd \mathsf{f}_n(k_n)\}$ with $\mathsf{f}_i \in \{\mathsf{pk}, \mathsf{vk}\}$, and $k_i \in \mathcal{K}_0$ for any $i \in \{1, \ldots, n\}$.

Assume that $(\mathcal{K}_0; C[[\overline{P_A}]_a]; \Phi_0)$ and $(\mathcal{K}_0; C[[\overline{P_B}]_b]; \Phi_0)$ (resp. $(\mathcal{K}_0; C[[\overline{P'_A}]_a]; \Phi_0)$, and $(\mathcal{K}_0; C[[\overline{P'_B}]_b]; \Phi_0)$):

- do not reveal any key in $\{k, \mathsf{pk}(k), \mathsf{vk}(k) \mid k \in \mathcal{K}_0\}$ unless if the key occurs explicitly in Φ_0; and
- do not reveal any shared key in C (resp. C');

Lastly, we assume that processes $\overline{P_A}, \overline{P'_A}$ and $\overline{P_B}, \overline{P'_B}$ do not use variable of channel type. We have that:

$$(\mathcal{K}_0; C[[\overline{P_A}]_a]; \Phi_0) \approx (\mathcal{K}_0; C'[[\overline{P'_A}]_a]; \Phi_0)$$
$$(\mathcal{K}_0; C[[\overline{P_B}]_b]; \Phi_0) \approx (\mathcal{K}_0; C'[[\overline{P'_B}]_b]; \Phi_0)$$

$$(\mathcal{K}_0; C[[\overline{P_A}]_a \mid [\overline{P_B}]_b]; \Phi_0) \approx (\mathcal{K}_0; C'[[\overline{P'_A}]_a \mid [\overline{P'_B}]_b]; \Phi_0)$$

Proof: (sketch) Actually, the two first steps are quite similar to the two first steps of the proof of Corollary 1, but we renamed the channel names that occur in $\overline{P_A}, \overline{P'_A}$ (resp. $\overline{P_B}, \overline{P'_B}$) before to compose these processes. This, together with our additional hypothesis on the variables of channel type, will allow us to identify easily whether a given action has been performed by $\overline{P_A}$ or $\overline{P_B}$ (resp. $\overline{P'_A}$ or $\overline{P'_B}$).

Then, consider a trace (tr, ϕ_S) issued from $S = (\mathcal{K}_0; C[[\overline{P_A}]_a \mid [\overline{P_B}]_b]; \Phi_0)$. First, we show that a similar trace (tr', ϕ_D) is also issued from $D = (\mathcal{K}_0; C[[\overline{P_A}]_a] \mid C[[\overline{P_B}]_b]; \Phi_0)$ (where channel names have been renamed). Actually, the processes along these two traces will be very similar (up to a transformation similar to the δ transformation used in the proof of Corollary 1 and a renaming on the channel names) but the labels involved in tr' have to be changed. Indeed, as soon as a message u will involved a public key in a "deducible position", the attacker will not be able to produce u and $\delta(u)$ using the same recipe. The way the recipe has to be changed depends in particular on whether the action has been made by $\overline{P_A}$ or $\overline{P_B}$. Second, relying on our hypothesis, we know that there exists (tr', ϕ'_D) issued from $D' = (\mathcal{K}_0; C'[[\overline{P'_A}]_a] \mid C'[[\overline{P'_B}]_b]; \Phi_0)$ (where again channel names have been renamed). However, to conclude, we have to go back to the process $S' = (\mathcal{K}_0; C'[[\overline{P'_A}]_a \mid [\overline{P'_B}]_b]; \Phi_0)$. This can be done by applying the reverse of the transformation δ on each process that occurs in the trace, but again the labels that occur in tr' have to be changed. Moreover, we have to ensure that this change will allow one to retrieve the original sequence tr. For this, we use the fact that the actions of $\overline{P_A}$ (resp. $\overline{P_B}$) are mimicked by $\overline{P'_A}$ (resp. $\overline{P'_B}$) (this is enforced by the way we have renamed channel names). Actually, some complications appear when an internal communication is performed on a public channel (this is indeed allowed by the semantics), but this problem can be solved by replacing such an internal step with two visible actions (an output followed by an input) having a clearly identifiable origin. ■

VI. Application: e-passport

We illustrate the usefulness of our composition results on the e-passport application. An electronic passport (or e-passport) is a paper passport with an RFID chip that stores the critical information printed on the passport. The International Civil Aviation Organisation (ICAO) standard [31] specifies the communication protocols that are used to access these information.

A. Protocols description

The information stored in the chip is organised in data groups (dg_1 to dg_{19}). For example, dg_5 contains a JPEG copy of the displayed picture, and dg_7 contains the displayed signature. The verification key $\mathsf{vk}(sk_P)$ of the passport, together with its certificate $\mathsf{sign}(\mathsf{vk}(sk_P), sk_{DS})$ issued by the Document Signer authority are stored in dg_{15}. The corresponding signing key sk_P is stored in a tamper resistant memory, and cannot be read or copied. For authentication purposes, a hash of all the dgs together with a signature on this hash value issued by the Document Signer authority are stored in a separate file, the Security Object Document:

$$sod \stackrel{\mathsf{def}}{=} \langle \mathsf{sign}(\mathsf{h}(dg_1, \ldots, dg_{19}), sk_{DS}),\ \mathsf{h}(dg_1, \ldots, dg_{19}) \rangle.$$

The ICAO standard specifies several protocols through which these information can be accessed. First, the Basic Access Control (*BAC*) protocol establishes sessions keys $ksenc$ and $ksmac$ to prevent skimming and eavesdropping on the subsequent communication with the e-passport. Once the *BAC* protocol has been successfully executed, the reader gains access to the information stored in the RFID tag through the Passive Authentication and the Active Authentication protocols that can be executed in any order (see Figure 2).

The Passive Authentication (PA) protocol is an authentication mechanism that proves that the content of the RFID chip

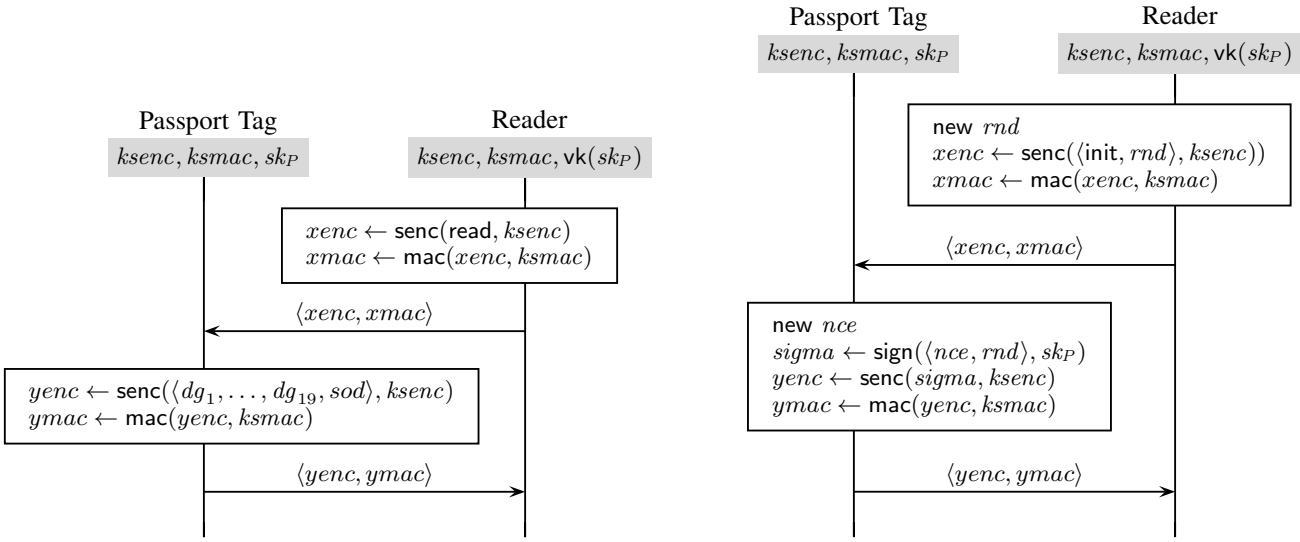

Figure 2. Passive and Active Authentication protocols

is authentic. Through *PA* the reader retrieves the information stored in the dgs and the sod. It then verifies that the hash value stored in the sod corresponds to the one signed by the Document Signer authority. It further checks that this hash value is consistant with the received dgs.

The Active Authentication (AA) protocol is an authentication mechanism that prevents cloning of the passport chip. It relies on the fact that the secret key sk_P of the passport cannot be read or copied. The reader sends a random challenge to the passport, that has to return a signature on this challenge using its private signature key sk_P. The reader can then verify using the verification key $\mathsf{vk}(sk_P)$ that the signature was built using the expected passport key.

B. Privacy analysis

Both protocols *PA* and *AA* rely on symmetric encryption, message authentication codes, signatures and the verification key generation function, to meet their security requirements. Note that $\mathsf{mac}(m, k)$ can be modelled in our setting using the hash function symbol, *i.e.* $\mathsf{mac}(m, k) \stackrel{\mathsf{def}}{=} \mathsf{h}(\langle m, k \rangle)$. Moreover, the only publicly known verification key is $\mathsf{vk}(sk_{DS})$. Thus, we can use our composition results, and in particular Theorem 2, to reason in a modular way about the privacy guarantees provided by the tagged version of the e-passport application[1].

[1]We tried to use the ProVerif tool to prove that the e-passport application as a whole (both *PA* and *AA* running in parallel) satisfies anonymity, but it failed to terminate, reinforcing the need for techniques for modular reasoning.

According to the ICAO standard, once the keys $ksenc$ and $ksmac$ have been established (using the *BAC* protocol), the reader can decide to execute *PA* and/or *AA* in any order. Formally, this corresponds to the parallel composition of *PA* and *AA*. We consider here that the keys $ksenc$ and $ksmac$ are "securely" pre-shared. We consider an arbitrary number of passports, each running an arbitrary number of times the *PA* and the *AA* protocols. This situation can be modelled in our calculus as follows:

$$P \stackrel{\mathsf{def}}{=} \texttt{new}\ sk_{DS}.\ !\texttt{new}\ sk_P.\ \texttt{new}\ id.\ \texttt{new}\ sig.\ \texttt{new}\ pic.\ \ldots\ !\texttt{new}\ ksenc.\ \texttt{new}\ ksmac.\ (PA \mid AA)$$

where id, sig, pic, ... represent the name, the signature, the displayed picture, *etc* of the e-passport owner, *i.e.* the data stored in the dgs (1-14) and (16-19). The subprocesses *PA* and *AA* model one session of the *PA* and *AA* protocol respectively. The name sk_{DS} models the signing key of the Document Signing authority used in all passports. Each passport (identified by its signing key sk_P, the owner's name, picture, signature, ...) can run multiple times and in any order the *PA* and *AA* protocols, but with different secret session keys $ksenc$ and $ksmac$, that should be established through execution of the *BAC* protocol (but that we've abstracted from).

1) Strong anonymity: To express strong anonymity as formally defined in [5] and briefly discussed at Section III-B, we will need to consider a victim's e-passport, whose name id_0, signature sig_0, picture pic_0, *etc.* are known to the attacker. The victim's e-passport follows like any other e-

passport the *PA* and *AA* protocols which can be respectively modelled by the following processes:

$$PA_0 \stackrel{\mathsf{def}}{=} PA\{id_0/id, sig_0/sig, pic_0/pic, \dots\}$$
$$AA_0 \stackrel{\mathsf{def}}{=} AA\{id_0/id, sig_0/sig, pic_0/pic, \dots\}$$

To formally express strong anonymity, we will consider the following situation:

$$C[\__1, \__2] \stackrel{\mathsf{def}}{=} !\ \mathtt{new}\ sk_P.\ \mathtt{new}\ id.\ \mathtt{new}\ sig.\ \mathtt{new}\ pic.\ \dots \\ !\ \mathtt{new}\ ksenc.\ \mathtt{new}\ ksmac.\ \__1 \\ |\ \mathtt{new}\ sk_P.\ !\mathtt{new}\ ksenc.\ \mathtt{new}\ ksmac.\ \__2$$

where the second hole will be filled with the processes modelling the victim's e-passport, while the first hole will be filled with the processes modelling any other e-passport. This system will be compared to the one where the victim's e-passport is not present at all. For this we consider the following situation:

$$C'[\_] \stackrel{\mathsf{def}}{=} !\ \mathtt{new}\ sk_P.\ \mathtt{new}\ id.\ \mathtt{new}\ sig.\ \mathtt{new}\ pic.\ \dots \\ !\ \mathtt{new}\ ksenc.\ \mathtt{new}\ ksmac.\ \_$$

whose unique hole will be filled with the processes modelling any e-passport but the victim's. In both situations, we will consider that the secret key sk_{DS} is secret whereas its associated verification key $\mathsf{vk}(sk_{DS})$ is publicly known to the attacker from the beginning, *i.e.* $\Phi_0 = \{w_1 \rhd \mathsf{vk}(sk_{DS})\}$.

To check if the tagged version of the-passport application preserves its users' strong anonymity, one thus needs to check if the following equivalence holds:

$$(sk_{DS}; C[[PA]_a \mid [AA]_b,\ [PA_0]_a \mid [AA_0]_b]; \Phi_0) \\ \approx \\ (sk_{DS}; C'[[PA]_a \mid [AA]_b]; \Phi_0)$$

Now, according to our Theorem 2, instead of checking the above equivalence, one can check *PA*'s and *AA*'s guarantees *w.r.t.* anonymity in isolation. In other words, the above equivalence can be derived from the two following equivalences that are simpler to check:

$$(sk_{DS}; C[[PA]_a, [PA_0]_a]; \Phi_0) \approx (sk_{DS}; C'[[PA]_a]; \Phi_0)$$
$$(sk_{DS}; C[[AA]_a, [AA_0]_a]; \Phi_0) \approx (sk_{DS}; C'[[AA]_a]; \Phi_0)$$

2) Strong unlinkability: To express strong unlinkability as defined in [5] and briefly discussed in Section III-B, we need on one hand to consider a system in which e-passports can execute the *PA* and *AA* protocols multiple times, and on the other hand a system in which e-passports can execute the *PA* and *AA* protocols at most once. For this we consider the two following composition contexts:

$$C[\_] \stackrel{\mathsf{def}}{=} !\mathtt{new}\ sk_P.\ \mathtt{new}\ id.\ \mathtt{new}\ sig.\ \mathtt{new}\ pic.\ \dots \\ !\mathtt{new}\ ksenc.\ \mathtt{new}\ ksmac.\ \_$$
$$C'[\_] \stackrel{\mathsf{def}}{=} !\mathtt{new}\ sk_P.\ \mathtt{new}\ id.\ \mathtt{new}\ sig.\ \mathtt{new}\ pic.\ \dots \\ \mathtt{new}\ ksenc.\ \mathtt{new}\ ksmac.\ \_$$

These two composition contexts differ on the replication before the generation of the session keys $ksenc$ and $ksmac$, modelling in the first case an unbounded number of executions of the process that will fill the unique hole, and in the second a unique session of the filling process.

To check if the tagged version of the e-passport application preserves strong unlinkability, one thus needs to check:

$$(sk_{DS}; C[[PA]_A|[AA]_b]; \Phi_0) \approx (sk_{DS}; C'[[PA]_a|[AA]_b]; \Phi_0)$$

We can instead check whether *PA* and *AA* satisfy unlinkability in isolation:

$$(sk_{DS}; C[[PA]_a]; \Phi_0) \approx (sk_{DS}; C'[[PA]_a]; \Phi_0)$$
$$(sk_{DS}; C[[AA]_b]; \Phi_0) \approx (sk_{DS}; C'[[AA]_b]; \Phi_0)$$

Then, using Theorem 2, we derive the required equivalence.

VII. Conclusion

In this paper, we investigate composition results for privacy-type properties expressed using trace equivalence. We have shown that secure protocols can be safely composed. We consider arbitrary equational theories and we assume that protocols may share some usual primitives provided they are tagged. Moreover, we have to assume that the shared keys are not revealed.

When shared keys are kept unknown during the whole execution, we transform any trace of the composition of two protocols under shared secrets into a trace on the composition under no shared secrets. This allows us to go back to the disjoint case for which composition works quite well. However, this transformation does not work anymore as soon as a shared key is revealed even if this key is the public part of an asymmetric key pair, and thus cannot be used to decrypt any ciphertext. Nevertheless, we establish a composition result in this setting by assuming that shared keys are either never revealed or known by the attacker from the beginning.

For the sake of simplicity, we only consider composition assuming that the initial knowledge of the attacker contains a bunch of names as well as some public keys and verification keys. We believe that our result can be extended to allow the attacker to have some non atomic messages in his initial provided that they are well-tagged. Our composition result allows one to consider public shared keys by giving them to the attacker initially (using the frame Φ_0). However, in our setting (and in many others) such a sequence has to be finite and thus we are only able to deal with a bounded number of public shared keys. To relax this hypothesis, we probably need to adapt our model. Lastly, for our composition result to work, we have to ensure that protocols used disjoint primitives or at least tagged them. However, real-world security protocols, typically do not use tags, at least not explicitly and not necessarily in the particular way stipulated by our composition result. Thus, it would be interesting to relax this condition. We could for instance use the implicit disjointness criterion developed in [19].

Acknowledgements. This work has been partially supported by the EPSRC projects *Verifying Interoperability Requirements in Pervasive Systems* (EP/F033540/1) and *Trust Domains* (TS/I002529/1), as well as the ANR projects PROSE and JCJC VIP n^o 11 JS02 006 01, and the grant DIGITEO API from Région Île-de-France.

REFERENCES

[1] D. Goodin, "Defects in e-passports allow real-time tracking," the Register, 26th January 2010.

[2] C. Caldwell, "A pass on privacy?" the New York Times, July 17, 2005.

[3] M. Barbaro and T. Z. Jr., "A face is exposed for AOL searcher No. 4417749," the New York Times, August 9, 2006.

[4] T. Chothia and V. Smirnov, "A traceability attack against e-passports," in *Proc. 14th International Conference on Financial Cryptography and Data Security (FC'10)*, ser. LNCS, vol. 6052. Springer, 2010.

[5] M. Arapinis, T. Chothia, E. Ritter, and M. Ryan, "Analysing unlinkability and anonymity using the applied pi calculus," in *Proc. 23rd Computer Security Foundations Symposium (CSF'10)*. IEEE Computer Society Press, 2010, pp. 107–121.

[6] J. K. Millen and V. Shmatikov, "Constraint solving for bounded-process cryptographic protocol analysis," in *Proc. 8th Conference on Computer and Communications Security (CCS'01)*. ACM Press, 2001, pp. 166–175.

[7] B. Blanchet, M. Abadi, and C. Fournet, "Automated verification of selected equivalences for security protocols," *Journal of Logic and Algebraic Programming*, vol. 75, no. 1, pp. 3–51, 2008.

[8] M. Abadi, B. Blanchet, and C. Fournet, "Just fast keying in the pi calculus," in *Proc. 13th European Symposium on Programming Languages and Systems (ESOP'04)*, ser. LNCS, vol. 2986. Springer, 2004.

[9] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, and M. L. Tobarra, "Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps," in *Proc. 6th ACM Workshop on Formal Methods in Security Engineering (FMSE 2008)*. ACM Press, 2008, pp. 1–10.

[10] A. Armando *et al.*, "The AVISPA Tool for the automated validation of internet security protocols and applications," in *Proc. 17th International Conference on Computer Aided Verification, CAV'2005*, ser. LNCS, vol. 3576. Springer, 2005, pp. 281–285.

[11] A. Tiu and J. E. Dawson, "Automating open bisimulation checking for the spi calculus," in *Proc. 23rd Computer Security Foundations Symposium (CSF'10)*. IEEE Computer Society Press, 2010, pp. 307–321.

[12] 3GPP, "Technical specification group services and system aspects; 3G security; cryptographic algorithm requirements (release 10)," 3rd Generation Partnership Project, Tech. Rep., 2011, 3GPP TS 33.105 V10.0.0.

[13] 3GPP, "Technical specification group services and system aspects; 3G security; security architecture (release 9)," 3rd Generation Partnership Project, Tech. Rep., 2010, 3GPP TS 33.102 V9.3.0.

[14] ——, "Technical specification group core network and terminals; mobile radio interface layer 3 specification; core network protocols; stage 3 (release 9)," 3rd Generation Partnership Project, Tech. Rep., 2010, 3GPP TS 24.008 V9.4.0.

[15] J. Kelsey, B. Schneier, and D. Wagner, "Protocol interactions and the chosen protocol attack." in *Proc. 5th Inter. Workshop on Security Protocols*, ser. LNCS, vol. 1361. Springer, 1997, pp. 91–104.

[16] J. D. Guttman and F. J. Thayer, "Protocol independence through disjoint encryption." in *Proc. 13th Computer Security Foundations Workshop (CSFW'00)*. IEEE Comp. Soc. Press, 2000, pp. 24–34.

[17] Ş. Ciobâcă and V. Cortier, "Protocol composition for arbitrary primitives," in *Proc. of the 23rd IEEE Computer Security Foundations Symposium (CSF'10)*. IEEE Computer Society Press, 2010, pp. 322–336.

[18] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proc. 42nd Annual Symposium on Foundations of Computer Science (FOCS'01)*. Las Vegas (Nevada, USA): IEEE Computer Society Press, 2001, pp. 136–145.

[19] R. Küsters and M. Tuengerthal, "Composition Theorems Without Pre-Established Session Identifiers," in *Proc. 18th Conference on Computer and Communications Security (CCS 2011)*. ACM Press, 2011, pp. 41–50.

[20] S. Andova, C. Cremers, K. G. Steen, S. Mauw, S. M. lsnes, and S. Radomirović, "Sufficient conditions for composing security protocols," *Information and Computation*, vol. 206, no. 2-4, pp. 425–459, 2008.

[21] S. Mödersheim and L. Viganò, "Secure pseudonymous channels," in *Proc. 14th European Symposium on Research in Computer Security (ESORICS'09)*, ser. LNCS, vol. 5789. Springer, 2009, pp. 337–354.

[22] S. Delaune, S. Kremer, and M. D. Ryan, "Composition of password-based protocols," in *Proc. 21st IEEE Computer Security Foundations Symposium (CSF'08)*. IEEE Computer Society Press, 2008, pp. 239–251.

[23] C. Chevalier, S. Delaune, and S. Kremer, "Transforming password protocols to compose," in *Proc. 31st Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'11)*, ser. Leibniz International Proceedings in Informatics. Leibniz-Zentrum für Informatik, 2011, pp. 204–216.

[24] B. Barak, R. Canetti, J. Nielsen, and R. Pass, "Universally composable protocols with relaxed set-up assumptions," in *Proc. 45th Symposium on Foundations of Computer Science (FOCS'04)*. IEEE Computer Society Press, 2004, pp. 186–195.

[25] M. Abadi and C. Fournet, "Mobile values, new names, and secure communication," in *Proc. 28th Symposium on Principles of Programming Languages (POPL'01)*. ACM Press, 2001, pp. 104–115.

[26] S. Delaune, S. Kremer, and M. D. Ryan, "Verifying privacy-type properties of electronic voting protocols," *Journal of Computer Security*, no. 4, pp. 435–487, Jul. 2008.

[27] M. Bruso, K. Chatzikokolakis, and J. den Hartog, "Formal verification of privacy for RFID systems," in *Proc. 23rd Computer Security Foundations Symposium (CSF'10)*. IEEE Computer Society Press, 2010.

[28] M. Abadi and V. Cortier, "Deciding knowledge in security protocols under equational theories," *Theoretical Computer Science*, vol. 387, no. 1-2, pp. 2–32, 2006.

[29] "ISO 15408-2: Common Criteria for Information Technology Security Evaluation - Part 2: Security functional components," ISO/IEC, Final draft, July 2009.

[30] V. Cortier and S. Delaune, "Safely composing security protocols," *Formal Methods in System Design*, vol. 34, no. 1, pp. 1–36, Feb. 2009.

[31] "PKI for machine readable travel documents offering ICC read-only access," International Civil Aviation Organization, Tech. Rep., 2004.

[32] Y. Chevalier and M. Rusinowitch, "Combining intruder theories," in *Proc. 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, ser. LNCS, vol. 3580. Springer, 2005, pp. 639–651.

[33] ——, "Combining intruder theories," INRIA, Tech. Rep. 5495, 2005.

[34] V. Cortier and S. Delaune, "Decidability and combination results for two notions of knowledge in security protocols," *Journal of Automated Reasoning*, 2012, to appear.

[35] N. Dershowitz and J.-P. Jouannaud, "Rewrite systems," in *Handbook of Theoretical Computer Science*. Elsevier, 1990, vol. B, ch. 6.

[36] F. Baader and K. U. Schulz, "Unification in the union of disjoint equational theories: Combining decision procedures." *Journal of Symbolic Computation*, vol. 21, no. 2, pp. 211–243, 1996.

APPENDIX

We consider several equational presentations $\mathcal{H}_1, \ldots, \mathcal{H}_n$ where $\mathcal{H}_i = (\Sigma_i, \mathsf{E}_i)$, for all $i \in \{1, \ldots, n\}$. Furthermore we assume that they are disjoint equational presentations (*i.e.* for all $i, j \in \{1, \ldots, n\}$, $\Sigma_i \cap \Sigma_j = \emptyset$) and consistent. Note that $\mathcal{T}(\Sigma_i, \mathcal{N} \cup \mathcal{X})$ and $\mathcal{T}(\Sigma_j, \mathcal{N} \cup \mathcal{X})$ share symbols, namely names and variables. Names are used to represent agent identities, keys or nonces, for $i, j \in \{1, \ldots, n\}$. We define $\Sigma = \bigcup_{i=1}^{n} \Sigma_i$ and $\mathsf{E} = \bigcup_{i=1}^{n} \mathsf{E}_i$. The *union* of the equational presentations $\mathcal{H}_1, \ldots, \mathcal{H}_n$ is the equational presentation defined by (Σ, E).

A. Factors

We denote by $\mathsf{root}(\cdot)$ the function that associates to each term $M \in \mathcal{T}(\Sigma, \mathcal{N} \cup \mathcal{X})$ the function symbol at position ϵ (root position) in M. For $M \in \mathcal{N} \cup \mathcal{X}$, we define $\mathsf{root}(M) = \bot$, where $\bot$ is a new symbol. The term N is *alien* to M if $\mathsf{root}(N) \in \Sigma_i$, $\mathsf{root}(M) \in \Sigma_j$ and $i \neq j$. We now introduce our notion of *factors*. A similar notion is also used in [32].

Definition 8 (factors): Let $M \in \mathcal{T}(\Sigma, \mathcal{N} \cup \mathcal{X})$. The *factors* of M, denoted $Fct(M)$, are the maximal syntactic subterms of M that are alien to M

Example 15: Let Σ_+ be the signature made up of the constant symbol 0 and the binary function $+$ and E_+ be the following set of equations:

$$\begin{array}{rclcrcl} x + (y + z) & = & (x + y) + z & & x + 0 & = & x \\ x + y & = & y + x & & x + x & = & 0 \end{array}$$

Consider the theories (Σ_0, E_0) and (Σ_+, E_+). Let M be the term $\mathsf{sdec}(\langle n_1 + \langle n_2, n_3 \rangle, \mathsf{proj}_1(n_1 + n_2) \rangle, n_3)$. The term $n_1 + \langle n_2, n_3 \rangle$ is a syntactic subterm of M alien to M since $\mathsf{root}(n_1 + \langle n_2, n_3 \rangle) \in \Sigma_+$ and $\mathsf{root}(M) \in \Sigma_0$. We have that

$$Fct(M) = \{n_1 + \langle n_2, n_3 \rangle,\ n_1 + n_2,\ n_3\}$$

B. Ordered rewriting

Most of the definitions and results in this subsection are borrowed from [33] and [34] since we use similar techniques. We consider the notion of *ordered rewriting* defined in [35], which is a useful tool that has been used (*e.g.* [36]) for proving correctness of combination of unification algorithms. Let $\prec$ be a simplification ordering[2] on ground terms assumed to be total and such that the minimum for $\prec$ is a name n_{min} and the constants in Σ are smaller that any ground term that is neither a constant nor a name. We define Σ^+ to be the set of the function symbols of Σ_1, …, Σ_n plus the name n_{min}, *i.e.* $\Sigma^+ = \Sigma_1 \cup \cdots \cup \Sigma_n \cup \{n_{min}\}$. In what follows, we furthermore assume that n_{min} is never used under restriction in frames.

[2]By definition $\prec$ satisfies that for all ground terms M, N_1, N_2, and for any position $p \neq \epsilon$ in M, we have $N_1 \prec M[N_1]_p$ and $N_1 \prec N_2$ implies $M[N_1]_p \prec M[N_2]_p$.

Given a possibly infinite set of equations $\mathcal{O}$ we define the ordered rewriting relation $\to_{\mathcal{O}}$ by $M \to_{\mathcal{O}} M'$ if and only if there exists an equation $N_1 = N_2 \in \mathcal{O}$, a position p in M and a substitution τ such that:

$$M = M[N_1\tau]_p, \;\; M' = M[N_2\tau]_p \text{ and } N_2\tau \prec N_1\tau.$$

It has been shown (see [35]) that by applying the *unfailing completion procedure* to a set of equations E we can derive a (possibly infinite) set of equations $\mathcal{O}$ such that on ground terms:

1) the relations $=_{\mathcal{O}}$ and $=_{\mathsf{E}}$ are equal,
2) the rewriting system $\to_{\mathcal{O}}$ is convergent.

It was showed ([36]) that applying unfailing completion to two disjoints sets of equations $\mathsf{E} = \mathsf{E}_1 \cup \mathsf{E}_2$, yields the set of generated equations $\mathcal{O}$ that is the disjoint union of the two systems $\mathcal{O}_1$ and $\mathcal{O}_2$ obtained by applying unfailing completion procedures to E_1 and to E_2 respectively. Thus, we can easily extend this result to $\mathsf{E} = \mathsf{E}_1 \cup \ldots \cup \mathsf{E}_n$ since $\mathsf{E}_1, \ldots, \mathsf{E}_n$ are all disjoint two at a time.

Thus, applying unfailing completion to $\mathsf{E} = \mathsf{E}_1 \cup \ldots \cup \mathsf{E}_n$ yields the set of generated equations $\mathcal{O}$ that is the disjoint union of $\mathcal{O}_1, \ldots, \mathcal{O}_n$ obtained by applying unfailing completion procedures respectively to $\mathsf{E}_1, \ldots, \mathsf{E}_n$.

Since the relation $\to_{\mathcal{O}}$ is convergent on ground terms, we define $M{\downarrow_{\mathsf{E}}}$ (or briefly $M{\downarrow}$) as the unique normal form of the ground term M for $\to_{\mathcal{O}}$. We denote by $M{\downarrow_{\mathsf{E}_i}}$ ($i \in \{1, \ldots, n\}$) the unique normal form of the ground term M for $\to_{\mathcal{O}_i}$. These notations are extended as expected to sets of terms.

Lemma 1: Let M be a ground term such that all its factors are in normal form and $\mathsf{root}(M) \in \Sigma_i$. Then

- either $M{\downarrow} \in Fct(M) \cup \{n_{min}\}$,
- or $\mathsf{root}(M{\downarrow}) \in \Sigma_i$ and $Fct(M{\downarrow}) \subseteq Fct(M) \cup \{n_{min}\}$.

Example 16: Consider the equational theory (Σ_+, E_+) described in Example 15. Let $\Sigma_{\mathsf{f}} = \{\mathsf{f}\}$ and $\mathsf{E}_{\mathsf{f}} = \{\mathsf{f}(x) = \mathsf{f}(y)\}$. We have that the theories E_+, E_{f} and $\mathsf{E}_+ \cup \mathsf{E}_{\mathsf{f}}$ are consistent. Let $M = \mathsf{f}(n_1 + n_2)$. We have that $M{\downarrow} = \mathsf{f}(n_{min})$. Hence $Fct(M{\downarrow})$ contains n_{min} whereas $Fct(M)$ does not contain this term.

Lemma 2: Let t be a ground term such that $t = C[u_1, \ldots, u_n]$ where C is a context built on Σ_i, $i \in \{1, \ldots, n\}$ and $u_1, \ldots, u_n$ are the factors of t in normal form. Furthermore, let D be the context built on Σ_i (possibly a hole) such that $t{\downarrow} = D[u_{j_1}, \ldots, u_{j_k}]$ with $j_1, \ldots, j_k \in \{0 \ldots n\}$ and $u_0 = n_{min}$ (the existence is given by Lemma 1). We have that for all ground terms $v_1, \ldots, v_n$ in normal form and alien to t, if

$$\forall (p, q) \in \{1 \ldots n\}, u_p = u_q \Leftrightarrow v_p = v_q$$

then $C[v_1, \ldots, v_n]{\downarrow} = D[v_{j_1}, \ldots, v_{j_k}]$ with $v_0 = n_{min}$

C. Name replacement

Let ρ be a bijective renaming of name of base type such that for all $k \in \mathrm{dom}(\rho)$, k is a "fresh name". Let δ_c^ρ ($c \in \{a, b\}$) be functions on terms that is defined as follows:

- $\delta_a^\rho(u) = u$ when u is a name or a variable;
- $\delta_b^\rho(u) = k$ when $u{\downarrow} = k\rho$ for some $k \in \mathrm{dom}(\rho)$ and $\mathsf{root}(u) \notin \Sigma_b \cup \Sigma_{\mathsf{tag}_b} \cup \Sigma_0$; otherwise $\delta_b^\rho(u) = u$ when u is a name or a variable;
- $\delta_c^\rho(\mathsf{f}(t_1, \ldots, t_k)) = \mathsf{f}(\delta_d^\rho(t_1), \ldots, \delta_d^\rho(t_k))$ if $\mathsf{f} \in \Sigma_d \cup \Sigma_{\mathsf{tag}_d}$ with $d \in \{a, b\}$.
- $\delta_c^\rho(\mathsf{f}(\mathsf{tag}_d(t_1), t_2)) = \mathsf{f}(\mathsf{tag}_d(\delta_d^\rho(t_1)), \delta_d^\rho(t_2))$ if $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$ and $d \in \{a, b\}$
- $\delta_c^\rho(\mathsf{h}(\mathsf{tag}_d(t_1))) = \mathsf{h}(\mathsf{tag}_d(\delta_d^\rho(t_1)))$ if $d \in \{a, b\}$
- $\delta_c^\rho(\mathsf{f}(t_1, \ldots, t_k)) = \mathsf{f}(\delta_c^\rho(t_1), \ldots, \delta_c^\rho(t_k))$ otherwise

The purpose of δ_b^ρ is to replace the keys used by B but created by A (*i.e.* $\mathrm{img}(\rho)$) with fresh names (*i.e.* $\mathrm{dom}(\rho)$).

Lemma 3: If t_1, t_2 are terms (that do not use $\mathrm{dom}(\rho)$) in normal form. For all $i \in \{a, b\}$, we have

$$t_1 = t_2 \text{ is equivalent to } \delta_i^\rho(t_1) = \delta_i^\rho(t_2)$$

Proof: The right implication of the lemma is trivial thus, we focus on the left implication: for all $i \in \{a, b\}$,$\delta_i^\rho(t_1) = \delta_i^\rho(t_2)$ implies $t_1 = t_2$. We prove this result by induction on $\max(|t_1|, |t_2|)$.

Base case $\max(|t_1|, |t_2|) = 1$*:* In such a case, we have that $t_1, t_2 \in \mathcal{X} \cup \mathcal{N}$. By definition of δ_a^ρ, we know that $\delta_a^\rho(t_1) = t_1$ and $\delta_a^\rho(t_2) = t_2$. Thus, we can conclude that $\delta_a^\rho(t_1) = \delta_a^\rho(t_2)$ implies $t_1 = t_2$. For the case $i = b$, we do a case analysis on whether $\delta_b^\rho(t_1) = \delta_b^\rho(t_2) = k$ for some $k \in \mathrm{dom}(\rho)$ or not.

Case $\delta_b^\rho(t_1) = k \in \mathrm{dom}(\rho)$*:* By hypothesis, we know that t_2 and t_1 do not use $\mathrm{dom}(\rho)$. Therefore, by definition of δ_b^ρ, we can deduce that, $t_2{\downarrow} = k\rho$ and $t_1{\downarrow} = k\rho$. Since t_1 and t_2 are in normal form, we can conclude that $t_1 = t_2 = k\rho$.

Case $\delta_b^\rho(t_1) \neq k$ *for every* $k \in \mathrm{dom}(\rho)$*:* By definition of δ_b^ρ, we have that $\delta_b^\rho(t_1) = t_1$ and $\delta_b^\rho(t_2) = t_2$, and thus $t_1 = t_2$.

Inductive step $\max(|t_1|, |t_2|) > 1$*:* Assume w.l.o.g. that $|t_1| > 1$. Thus, there exists a symbol function f and terms $u_1, \ldots u_n$ such that $t_1 = \mathsf{f}(u_1, \ldots u_n)$. Since t_1 is in normal form, we can deduce that $t_1{\downarrow} \neq k\rho$, for every $k \in \mathrm{dom}(\rho)$. We do a case analysis on t_1 :

Case $\mathsf{f} \in \Sigma_d \cup \Sigma_{\mathsf{tag}_d}$ *and* $d \in \{a, b\}$*:* In such a case, we have that $\delta_i^\rho(t_1) = \mathsf{f}(\delta_d^\rho(u_1), \ldots, \delta_d^\rho(u_n))$. But $\delta_i^\rho(t_2) = \delta_i^\rho(t_1)$ and by definition of δ_i^ρ, we know that it implies that there exists $v_1, \ldots, v_n$ such that $t_2 = \mathsf{f}(v_1, \ldots, v_n)$ and $\delta_i^\rho(t_2) = \mathsf{f}(\delta_d^\rho(v_1), \ldots, \delta_d^\rho(v_n))$. Thus we have that $\delta_d^\rho(v_j) = \delta_d^\rho(u_j)$ for all $j \in \{1, \ldots, n\}$. Furthermore, since t_1 and t_2 are in normal form, we also know that u_j and v_j are in normal form, for every j. But, $\max(|t_1|, |t_2|) > \max(|u_j|, |v_j|)$, for any j, thus by our

inductive hypothesis, we can deduce that $u_j = v_j$, for all j and so $t_1 = \mathsf{f}(u_1, \ldots, u_n) = \mathsf{f}(v_1, \ldots, v_n) = t_2$.

Case $t_1 = \mathsf{f}(\mathsf{tag}_d(w_1), w_2)$, $d \in \{a, b\}$ *and* $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$*:* In this case, we know that $\delta_i^\rho(t_1) = \mathsf{f}(\mathsf{tag}_d(\delta_d^\rho(w_1)), \delta_d^\rho(w_2))$. But we know that $\delta_i^\rho(t_2) = \delta_i^\rho(t_1) = \mathsf{f}(\mathsf{tag}_d(\delta_d^\rho(w_1)), \delta_d^\rho(w_2))$. Thus thanks to t_2 being in normal form and by definition of δ_i^ρ, it implies that there exists v_1 and v_2 such that $t_2 = \mathsf{f}(\mathsf{tag}_d(v_1), v_2)$ and so $\delta_i^\rho(t_2) = \mathsf{f}(\mathsf{tag}_d(\delta_d^\rho(v_1)), \delta_d^\rho(v_2))$. Thus, we have that $\delta_d^\rho(v_1) = \delta_d^\rho(u_1)$ and $\delta_d^\rho(v_2) = \delta_d^\rho(u_2)$. Moreover, t_1 and t_2 being in normal form and not using $\mathrm{dom}(\rho)$, so are u_j and v_j for $j \in \{1, 2\}$, so we can apply inductive hypothesis and conclude that $v_1 = u_1$ and $v_2 = u_2$ and so $t_1 = t_2$.

Case $t_1 = \mathsf{h}(\mathsf{tag}_d(w_1))$ *and* $d \in \{a, b\}$*:* This case is analogous to the previous one and can be handled in a similar way.

Else case: Otherwise, we have that $\mathsf{f} \in \Sigma_0$ but the root symbol of u_1 is not tag_a or tag_b. By definition of δ_i^ρ, we can deduce that $\delta_i^\rho(t_1) = \mathsf{f}(\delta_i^\rho(u_1), \ldots, \delta_i^\rho(u_n))$. Since $\delta_i^\rho(t_1) = \delta_i^\rho(t_2)$, we can deduce that the top symbol of t_2 is also f and so there exists $v_1, \ldots, v_n$ such that $t_2 = \mathsf{f}(v_1, \ldots, v_n)$. In the previous cases, we showed that if $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}, \mathsf{h}\}$ and the top symbol of v_1 is tag_a or tag_b, then $\delta_i^\rho(t_1) = \delta_i^\rho(t_2)$ implies that the top symbol of u_1 is also tag_a or tag_b. Thus, thanks to our hypothesis, we can deduce that either $\mathsf{f} \notin \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}, \mathsf{h}\}$ or the top symbol of v_1 is different from tag_a and tag_b. Hence by definition of δ_i^ρ, we can deduce that $\delta_i^\rho(t_2) = \mathsf{f}(\delta_i^\rho(v_1), \ldots, \delta_i^\rho(v_n))$ and so $\delta_i^\rho(v_j) = \delta_i^\rho(u_j)$ for all $j \in \{1, \ldots, n\}$. Moreover, t_1 and t_2 being in normal form and not using names in $\mathrm{dom}(\rho)$, implies that so are u_j and v_j for all $j \in \{1, \ldots, n\}$. We can thus invoke our inductive hypothesis and conclude that $u_j = v_j$ for all $j \in \{1, \ldots, n\}$ and so $t_1 = t_2$. ■

Lemma 4: Let t_1, t_2 two terms (that do not use $\mathrm{dom}(\rho)$) in normal form. If $\delta_a^\rho(t_1) = \delta_b^\rho(t_2)$ then $t_1 = t_2$.

Proof: We prove the result by induction on $|\delta_a^\rho(t_1)|$.

Base case $|\delta_a^\rho(t_1)| = 1$*:* In such a case, we have that $\delta_a^\rho(t_1) \in \mathcal{N} \cup \mathcal{X}$. Assume first that $\delta_a^\rho(t_1) \in \mathcal{X}$ and so $\delta_b^\rho(t_2) \in \mathcal{X}$: By definition of δ_a^ρ and δ_b^ρ, we can deduce that $\delta_a^\rho(t_1) = t_1$ and $\delta_b^\rho(t_2) = t_2$. Thus we conclude that $t_1 = t_2$. Assume now that $\delta_a^\rho(t_1) \in \mathcal{N}$. We need to distinguish two cases :

Case $\delta_a^\rho(t_1) \in \mathrm{dom}(\rho)$*:* By definition of δ_a^ρ, $\delta_a^\rho(t_1) \in \mathcal{N}$ implies that $\delta_a^\rho(t_1) = t_1$. But we assumed that t_1, t_2 do not use $\mathrm{dom}(\rho)$. Thus this case is impossible.

Case $\delta_a^\rho(t_1) \notin \mathrm{dom}(\rho)$*:* Once again by definition of δ_a^ρ, we have $\delta_a^\rho(t_1) = t_1$. Furthermore, since t_2 do not use $\mathrm{dom}(\rho)$ and $\delta_b^\rho(t_2) \notin \mathrm{dom}(\rho)$, we also have that $\delta_b^\rho(t_2) = t_2$ by definition of δ_b^ρ. Thus we conclude that $t_1 = t_2$.

Inductive step $|\delta_a^\rho(t_1)| > 1$*:* In that case, we have that $\delta_a^\rho(t_1) = \mathsf{f}(u_1, \ldots, u_n) = \delta_b^\rho(t_2)$. Assume that $\mathsf{f} \in \Sigma_d \cup \Sigma_{\mathsf{tag}_d}$ with $d \in \{a, b\}$. By definition of δ_a^ρ and δ_b^ρ, we can deduce that $\mathsf{root}(t_1) = \mathsf{f} = \mathsf{root}(t_2)$. Furthermore, if we assume that $t_1 = \mathsf{f}(v_1, \ldots, v_n)$ and $t_2 = \mathsf{f}(w_1, \ldots, w_n)$, we would have $\delta_d^\rho(v_j) = \delta_d^\rho(w_j)$ for all $j \in \{1, \ldots, n\}$. By Lemma 3, we deduce that $v_j = w_j$ for all $j \in \{1, \ldots, n\}$. Hence, we conclude that $t_1 = t_2$. Assume now that $\mathsf{f} \in \Sigma_0$. According to the definition of δ_a^ρ and δ_b^ρ, there exists $v_1, \ldots, v_n$ and $w_1, \ldots, w_n$ such that $t_1 = \mathsf{f}(v_1, \ldots, v_n)$, $t_2 = \mathsf{f}(w_1, \ldots, w_n)$ and $\delta_k^\rho(v_j) = \delta_\ell^\rho(w_j)$, for some $k, \ell \in \{a, b\}$. Moreover, t_1 and t_2 being in normal form and not using names in $\mathrm{dom}(\rho)$ implies that so are v_j and w_j for all $j \in \{1, \ldots, n\}$. Now, either $k = \ell$ and so by Lemma 3, we have that $v_j = w_j$, else $k \neq \ell$ but then by our inductive hypothesis, we also have $v_j = w_j$. Hence we conclude that $t_1 = t_2$. ■

Definition 9 (Factor for Σ_0*):* Let u be a term. We define $Fct_{\Sigma_0}(u)$ the factors of a term u for Σ_0 as the maximal syntactic subterms of u of the form $\mathsf{f}(\mathsf{tag}_i(u_1), u_2)$ with $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$ and $i \in \{a, b\}$; or of the form $\mathsf{h}(\mathsf{tag}_i(u_1))$ with $i \in \{a, b\}$; or which root symbol is not in Σ_0.

Note that the factors for Σ_0 of a u are slightly different from the factors of u defined at Definition 8. Typically, the difference comes from the fact that we need to differentiate terms that use properly the tags from those that don't.

Example 17: Consider the theories (Σ_a, E_a), $(\Sigma_{\mathsf{tag}_a}, \mathsf{E}_{\mathsf{tag}_a})$ and (Σ_+, E_+) given in Example 15. Let u be the term $\langle \mathsf{senc}(\mathsf{tag}_a(n_1), \langle n_1 + n_2, n_3 \rangle), \mathsf{sign}(n_4, n_5) \rangle$. We have:

- $Fct(u) = \{\mathsf{tag}_a(n_1);\ n_1 + n_2;\ n_3;\ n_4;\ n_5\}$
- $Fct_{\Sigma_0}(u) = \{\mathsf{senc}(\mathsf{tag}_a(n_1), \langle n_1 + n_2, n_3 \rangle);\ n_4;\ n_5\}$
- $Fct_{\Sigma_0}(n_1 + n_2) = \{n_1 + n_2\}$

Lemma 5: Let u be a term, C a context (possibly a hole) built over Σ_0 and $v_1, \ldots, v_m$ terms such that $u = C[v_1, \ldots, v_m]$ and $\{v_1, \ldots, v_m\} = Fct_{\Sigma_0}(u)$. If for all $i \in \{1, \ldots, m\}$, $v_i{\downarrow} \neq k\rho$ for any $k \in \mathrm{dom}(\rho)$, then we have that $\delta_a^\rho(u) = \delta_b^\rho(u)$.

Proof: We prove this lemma by induction on the syntactic size of $|C|$.

Base case $|C| = 0$ *:* In this case, C is a hole which means that either $\mathsf{root}(u) \notin \Sigma_0$, or u is of the form $\mathsf{f}(\mathsf{tag}_i(u_1), u_2)$ with $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$ and $i \in \{a, b\}$, or u is of the form $\mathsf{h}(\mathsf{tag}_i(u_1))$ with $i \in \{a, b\}$. Remember that $u{\downarrow} \neq k\rho$ for any $k \in \mathrm{dom}(\rho)$. We do a case analysis on u :

Case $u \in \mathcal{N} \cup \mathcal{X}$*:* In such a case, since we assume that $u{\downarrow} \neq k\rho$ for any $k \in \mathrm{dom}(\rho)$, then by definition of δ_a^ρ and δ_b^ρ, we have that $\delta_a^\rho(u) = u = \delta_b^\rho(u)$.

Case $u = \mathsf{f}(u_1, \ldots, u_n)$ with $\mathsf{f} \notin \Sigma_0$: $\mathsf{f} \notin \Sigma_0$ implies that $\mathsf{f} \in \Sigma_d \cup \Sigma_{\mathsf{tag}_d}$ with $d \in \{a, b\}$. Thus, by definition of δ_a^ρ and δ_b^ρ, we have that $\delta_a^\rho(u) = \mathsf{f}(\delta_d^\rho(u_1), \ldots, \delta_d^\rho(u_n)) = \delta_b^\rho(u)$.

Case $u = \mathsf{f}(\mathsf{tag}_d(u_1), u_2)$ *with* $d \in \{a, b\}$ *and* $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$*:* In that case, we have by definition of δ_a^ρ and δ_b^ρ that $\delta_a^\rho(u) = \mathsf{f}(\mathsf{tag}_d(\delta_d^\rho(u_1)), \delta_d^\rho(u_2)) = \delta_b^\rho(u)$.

Case $u = \mathsf{h}(\mathsf{tag}_d(u_1))$ *with* $d \in \{a, b\}$*:* This case is analogous to the previous one and can be handled in a similar way.

Inductive step $|C| > 1$ *:* In this case, we know that $u = \mathsf{f}(u_1, \ldots, u_n)$ with $\mathsf{f} \in \Sigma_0$. But $|C| > 1$ also implies that for all $i \in \{1, \ldots, n\}$, there exists a sub context C_i (possibly a hole) of C such that $u_i = C_i[v_1^i, \ldots, v_{m_i}^i]$ and $\{v_1^i, \ldots, v_{m_i}^i\} = Fct_{\Sigma_0}(u_i)$. Note that $Fct_{\Sigma_0}(u_i) \subseteq Fct_{\Sigma_0}(u)$. Since C_i is a sub context of C, then thanks to our hypothesis, we have that for $j \in \{1, \ldots, m_i\}$, $v_j^i{\downarrow} \neq k\rho$ for any $k \in \mathrm{dom}(\rho)$. Thus we can apply our inductive hypothesis and deduce that $\delta_a^\rho(u_i) = \delta_b^\rho(u_i)$. Since $C \neq [\_]$ and $\mathsf{f} \in \Sigma_0$, it must be the case that $\mathsf{root}(u_1) \neq \mathsf{tag}_k$ for any $k \in \{a, b\}$, thus by definition of δ_a^ρ and δ_b^ρ, since $\mathsf{root}(u) \in \Sigma_0$, we have that $\delta_a^\rho(u) = \mathsf{f}(\delta_a^\rho(u_1), \ldots, \delta_a^\rho(u_n))$ and $\delta_b^\rho(u) = \mathsf{f}(\delta_b^\rho(u_1), \ldots, \delta_b^\rho(u_n))$. Thanks to the previously established equalities $\delta_a^\rho(u_i) = \delta_b^\rho(u_i)$ for all $i \in \{1, \ldots, n\}$, we can conclude that $\delta_a^\rho(u) = \delta_b^\rho(u)$. ■

Lemma 6: Let u be a term in normal form that do no use $\mathrm{dom}(\rho)$. We have that for all $i \in \{a, b\}$, $\delta_i^\rho(u)$ is in normal form and $\mathsf{root}(\delta_i^\rho(u)) = \mathsf{root}(u)$.

Proof: We prove this result by induction on $|u|$.
Base case $|u| = 1$*:* In such a case, $u \in \mathcal{X} \cup \mathcal{N}$. If $u \in \mathcal{X}$, then for all $i \in \{a, b\}$, $\delta_i^\rho(u) = u$. Since u is in normal form then the result holds. If $u \in \mathcal{N}$, by definition, we also have that $\delta_i^\rho(u) \in \mathcal{N}$ and so $\delta_i^\rho(u)$ is in normal form with the same root as u, namely $\bot$.

Inductive $|u| > 1$*:* In this case, we have that $u = C[u_1, \ldots, u_n]$ with $u_1, \ldots, u_n$ factors of u. Assume first that C is built upon $\Sigma_j \cup \Sigma_{\mathsf{tag}_j}$, for some $j \in \{a, b\}$. Since u is in normal form, then for all position p of C, we have that $u_{|p}{\downarrow} \notin \mathrm{img}(\rho)$ and so for all $k \in \{a, b\}$, $\delta_k^\rho(u) = C[\delta_j^\rho(u_1), \ldots, \delta_j^\rho(u_n)]$. By inductive hypothesis on $u_1, \ldots, u_n$. We have that $\delta_j^\rho(u_1), \ldots, \delta_j^\rho(u_n)$ are in normal form and $\mathsf{root}(\delta_j^\rho(u_1)) = \mathsf{root}(u_1)$, $\ldots \mathsf{root}(\delta_j^\rho(u_n)) = \mathsf{root}(u_n)$, thus $\delta_j^\rho(u_1), \ldots, \delta_j^\rho(u_n)$ are factors of $\delta_i^\rho(u)$. Thus, since u is in normal form, by Lemmas 3 and 2 , we have that $\delta_i^\rho(u){\downarrow} = C[\delta_j^\rho(u_1), \ldots, \delta_j^\rho(u_n)] = \delta_i^\rho(u)$. Furthermore, we also have that $\mathsf{root}(\delta_i^\rho(u)) = \mathsf{root}(u)$. Assume now that C is built upon Σ_0. Thus, assume that $u = \mathsf{f}(v_1, \ldots, v_m)$. By definition of δ_a^ρ and δ_b^ρ, there exists $j \in \{a, b\}$ such that $\delta_i^\rho(u) = \mathsf{f}(\delta_j^\rho(v_1), \ldots, \delta_j^\rho(v_m))$. We do a case analysis on f:

Case $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{pk}, \mathsf{sign}, \mathsf{vk}, \mathsf{h}, \langle\ \rangle\}$*:* In this case, by the equational theory E_0, we have that $\delta_i^\rho(u){\downarrow} = \mathsf{f}(\delta_j^\rho(v_1){\downarrow}, \ldots, \delta_j^\rho(v_m){\downarrow})$. Since by inductive hypothesis, $\delta_j^\rho(v_k)$ is in normal form, for all $k \in \{1, \ldots, m\}$, we can deduce that $\delta_i^\rho(u)$ is also in normal form and $\mathsf{root}(\delta_i^\rho(u)) = \mathsf{f} = \mathsf{root}(u)$.

Case $\mathsf{f} = \mathsf{sdec}$*:* Then $m = 2$, and by definition of δ_a^ρ and δ_b^ρ, we have that $\delta_i^\rho(u) = \mathsf{sdec}(\delta_j^\rho(v_1), \delta_j^\rho(v_2))$, with $j \in \{a, b\}$. By inductive hypothesis, we have that $\delta_j^\rho(v_1)$ and $\delta_j^\rho(v_2)$ are both in normal form and have the same root as v_1 and v_2 respectively.

Assume first that sdec cannot be reduced, *i.e.* $\delta_i^\rho(u){\downarrow} = \mathsf{sdec}(\delta_j^\rho(v_1){\downarrow}, \delta_j^\rho(v_2){\downarrow}) = \mathsf{sdec}(\delta_j^\rho(v_1), \delta_j^\rho(v_2))$. Thus the result holds. Otherwise, if sdec can be reduced, it implies that there exists w_1, w_2 such that $\delta_j^\rho(v_1) = \mathsf{senc}(w_1, w_2)$ and $\delta_j^\rho(v_2) = w_2$. But by definition of δ_j^ρ, there must exist $k \in \{a, b\}$, and w_1', w_2' such that $\delta_j^\rho(v_1) = \mathsf{senc}(\delta_k^\rho(w_1'), \delta_k^\rho(w_2'))$, $v_1 = \mathsf{senc}(w_1', w_2')$, $w_1 = \delta_k^\rho(w_1')$ and $w_2 = \delta_k^\rho(w_2')$. Thus, we have that $\delta_j^\rho(v_2) = \delta_k^\rho(w_2')$. Thanks to Lemmas 3 and 4, we can conclude that $v_2 = w_2'$ and so $u = \mathsf{sdec}(\mathsf{senc}(w_1', w_2'), w_2')$. But in such a case, we would have that u is not in normal form which contradicts our hypothesis.

Case $\mathsf{f} = \mathsf{check}$*:* Then $m = 2$, and by definition of δ_a^ρ and δ_b^ρ, we have that $\delta_i^\rho(u) = \mathsf{check}(\delta_j^\rho(v_1), \delta_j^\rho(v_2))$, with $j \in \{a, b\}$. By inductive hypothesis, we have that $\delta_j^\rho(v_1)$ and $\delta_j^\rho(v_2)$ are both in normal form and have the same root as v_1 and v_2 respectively.

Assume first that check is cannot be reduced: this case is analogous to the sdec one and can be handled similarly. Otherwise, if check can be reduced, it implies that there exist w_1, w_2 such that $\delta_j^\rho(v_1) = \mathsf{sign}(w_1, w_2)$ and $\delta_j^\rho(v_2) = \mathsf{vk}(w_2)$. But by definition of δ_j^ρ, there must exist $k \in \{a, b\}$, and w_1', w_2' such that $\delta_j^\rho(v_1) = \mathsf{sign}(\delta_k^\rho(w_1'), \delta_k^\rho(w_2'))$, $v_1 = \mathsf{sign}(w_1', w_2')$, $w_1 = \delta_k^\rho(w_1')$ and $w_2 = \delta_k^\rho(w_2')$. Thus, we have that $\delta_j^\rho(v_2) = \mathsf{vk}(\delta_k^\rho(w_2')) = \delta_k^\rho(\mathsf{vk}(w_2'))$. Thanks to Lemmas 3 and 4, we can conclude that $v_2 = \mathsf{vk}(w_2')$ and so $u = \mathsf{check}(\mathsf{sign}(w_1', w_2'), \mathsf{vk}(w_2')$. But in such a case, we would have that u is not in normal form which contradicts our hypothesis.

Case $\mathsf{f} = \mathsf{adec}$*:* Then $m = 2$, and by definition of δ_a^ρ and δ_b^ρ, we have that $\delta_i^\rho(u) = \mathsf{adec}(\delta_j^\rho(v_1), \delta_j^\rho(v_2))$, with $j \in \{a, b\}$. By inductive hypothesis, we have that $\delta_j^\rho(v_1)$ and $\delta_j^\rho(v_2)$ are both in normal form and have the same root as v_1 and v_2 respectively.

Assume first that adec cannot be reduced: this case is analogous to the sdec one and can be handled similarly. Otherwise, if adec can be reduced, it implies that there exist w_1, w_2 such that $\delta_j^\rho(v_1) = \mathsf{aenc}(w_1, \mathsf{pk}(w_2))$ and $\delta_j^\rho(v_2) = w_2$. But by definition of δ_j^ρ, there must exist $k \in \{a, b\}$, and w_1', w_2' such that $\delta_j^\rho(v_1) = \mathsf{aenc}(\delta_k^\rho(w_1'), \mathsf{pk}(\delta_k^\rho(w_2')))$, $v_1 = \mathsf{aenc}(w_1', \mathsf{pk}(w_2'))$, $w_1 = \delta_k^\rho(w_1')$ and $w_2 = \delta_k^\rho(w_2')$. Thus, we have that $\delta_j^\rho(v_2) = \delta_k^\rho(w_2')$. Thanks to Lemmas 3 and 4, we can conclude that $v_2 = w_2'$ and so $u = \mathsf{adec}(\mathsf{aenc}(w_1', \mathsf{pk}(w_2')), w_2')$. But in such a case, we would have that u is not in normal form which contradicts our hypothesis. ■

D. δ_i^ρ and tagged term

Let $i \in \{a, b\}$. Let $u \in \mathcal{T}(\Sigma_i \cup \Sigma_{\mathsf{tag}_i} \cup \Sigma_0, \mathcal{N} \cup \mathcal{X})$. As defined in Section IV, $\mathsf{test}_i(u)$ is a conjunction of elementary formulas (equalities between term). Let α be

a ground substitution such that $fv(u) \subseteq \mathrm{dom}(\alpha)$. We say that α satisfies $t_1 = t_2$, denoted $\alpha \vDash t_1 = t_2$, if $t_1\alpha\downarrow = t_2\alpha\downarrow$. Typically, in an execution of a protocol, the substitution α will represent the value of the inputs that are inside the term u. In this subsection, we will assume the existence of a renaming ρ. Furthermore, we will assume that all terms we consider do not use $\mathrm{dom}(\rho)$ (they might once we apply δ^ρ_a or δ^ρ_b on them). For a term u, we will denote $st(u)$ the set of subterms of u. At last, for a term substitution α, we will consider for all $i \in \{a, b\}$, $\delta^\rho_i(\alpha)$ the substitution such that $\mathrm{dom}(\alpha) = \mathrm{dom}(\delta^\rho_i(\alpha))$ and for all $x \in \mathrm{dom}(\alpha)$, $x\delta^\rho_i(\alpha) = \delta^\rho_i(x\alpha)$

Lemma 7: Let $i \in \{a, b\}$. Let $u \in \mathcal{T}(\Sigma_i \cup \Sigma_0, \mathcal{N} \cup \mathcal{X})$. Let α be a ground substitution such that $fv(u) \subseteq \mathrm{dom}(\alpha)$ and for all $x \in \mathrm{dom}(\alpha)$, $x\alpha$ is in normal form. We have that:

- $\delta^\rho_i([u]_i\alpha) = \delta^\rho_i([u]_i)\delta^\rho_i(\alpha)$; and
- If $\alpha \vDash \mathsf{test}_i([u]_i)$ then $\delta^\rho_i([u]_i\alpha)\downarrow = \delta^\rho_i([u]_i\alpha\downarrow)$.

Proof: We prove the two results separately. First of all, we show by induction on $|u|$ that $\delta^\rho_i([u]_i\alpha) = \delta^\rho_i([u]_i)\delta^\rho_i(\alpha)$:

Base case $|u| = 1$*:* In this case, $u \in \mathcal{N} \cup \mathcal{X}$. If $u \in \mathcal{N}$ then we have that $[u]_i = u$ and so $[u]_i\alpha = u$ and $\delta^\rho_i(u) \in \mathcal{N}$. Thus, we have that $\delta^\rho_i([u]_i\alpha) = \delta^\rho_i(u) = \delta^\rho_i(u)\delta^\rho_i(\alpha) = \delta^\rho_i([u]_i)\delta^\rho_i(\alpha)$. Else if $u \in \mathcal{X}$, then we also have that $[u]_i = u$ and $\delta^\rho_i(u) = u$. Thus, $\delta^\rho_i(u)\delta^\rho_i(\alpha) = u\delta^\rho_i(\alpha)$. Since $u \in \mathcal{X}$ and $fv(u) \subseteq \mathrm{dom}(\alpha)$, we have that $u\delta^\rho_i(\alpha) = \delta^\rho_i(u\alpha)$, thus $\delta^\rho_i([u]_i\alpha) = \delta^\rho_i(u\alpha) = u\delta^\rho_i(\alpha) = \delta^\rho_i(u)\delta^\rho_i(\alpha) = \delta^\rho_i([u]_i)\delta^\rho_i(\alpha)$.

Inductive step $|u| > 1|$*:* In this case, $u = \mathsf{f}(u_1, \ldots, u_n)$. We do a case analysis on f.

Case $\mathsf{f} \in \Sigma_i$*:* In such a case, we have that $[u]_i = \mathsf{f}([u_1]_i, \ldots, [u_n]_i)$. But by definition of δ^ρ_i, we have that $\delta^\rho_i([u]_i\alpha) = \mathsf{f}(\delta^\rho_i([u_1]_i\alpha), \ldots, \delta^\rho_i([u_n]_i\alpha))$ and $\delta^\rho_i([u]_i) = \mathsf{f}(\delta^\rho_i([u_1]_i), \ldots, \delta^\rho_i([u_n]_i))$. By our inductive hypothesis, we can deduce that for all $k \in \{1, \ldots, n\}$, we have that $\delta^\rho_i([u_k]_i\alpha) = \delta^\rho_i([u_k]_i)\delta^\rho_i(\alpha)$. Thus, we can deduce that $\delta^\rho_i([u]_i\alpha) = \mathsf{f}(\delta^\rho_i([u_1]_i), \ldots, \delta^\rho_i([u_n]_i))\delta^\rho_i(\alpha) = \delta^\rho_i([u]_i)\delta^\rho_i(\alpha)$.

Case $\mathsf{f} \in \{\mathsf{aenc}, \mathsf{sign}, \mathsf{senc}\}$*:* In this case $n = 2$, and by definition of $[u]_i$, we have that $[u]_i = \mathsf{f}(\mathsf{tag}_i([u_1]_i), [u_2]_i)$. Thus, we have that $\delta^\rho_i([u]_i) = \mathsf{f}(\mathsf{tag}_i(\delta^\rho_i([u_1]_i)), \delta^\rho_i([u_2]_i))$ and $\delta^\rho_i([u]_i\alpha) = \mathsf{f}(\mathsf{tag}_i(\delta^\rho_i([u_1]_i\alpha)), \delta^\rho_i([u_2]_i\alpha))$. But by our inductive hypothesis, we can deduce that $\delta^\rho_i([u_k]_i\alpha) = \delta^\rho_i([u_k]_i)\delta^\rho_i(\alpha)$, for all $k \in \{1, 2\}$. Thus, we can deduce that $\delta^\rho_i([u]_i\alpha) = \mathsf{f}(\mathsf{tag}_i(\delta^\rho_i([u_1]_i\alpha)), \delta^\rho_i([u_2]_i\alpha))$ and so we conclude that $\delta^\rho_i([u]_i\alpha) = \mathsf{f}(\mathsf{tag}_i(\delta^\rho_i([u_1]_i)\delta^\rho_i(\alpha)), \delta^\rho_i([u_2]_i)\delta^\rho_i(\alpha)) = \delta^\rho_i([u]_i)\delta^\rho_i(\alpha)$.

Case $\mathsf{f} = \mathsf{h}$*:* This case is analogous to the previous one and can be handled in a similar way.

Case $\mathsf{f} \in \{\mathsf{sdec}, \mathsf{adec}, \mathsf{check}\}$*:* In this case $n = 2$, and by definition of $[u]_i$, we have that $[u]_i = \mathsf{untag}_i(\mathsf{f}([u_1]_i, [u_2]_i))$. Thus, we have that $\delta^\rho_i([u]_i) = \mathsf{untag}_i(\mathsf{f}(\delta^\rho_i([u_1]_i), \delta^\rho_i([u_2]_i)))$ and $\delta^\rho_i([u]_i\alpha) = \mathsf{untag}_i(\mathsf{f}(\delta^\rho_i([u_1]_i\alpha), \delta^\rho_i([u_2]_i\alpha)))$. Once again, with our inductive hypothesis, we can deduce that $\delta^\rho_i([u_k]_i\alpha) = \delta^\rho_i([u_k]_i)\delta^\rho_i(\alpha)$, for $k \in \{1, 2\}$, and so we can conclude that $\delta^\rho_i([u]_i\alpha) = \mathsf{untag}_i(\mathsf{f}(\delta^\rho_i([u_1]_i), \delta^\rho_i([u_2]_i)))\delta^\rho_i(\alpha) = \delta^\rho_i([u]_i)\delta^\rho_i(\alpha)$.

Else case: In this case, by definition of $[u]_i$, we have that $[u]_i = \mathsf{f}([u_1]_i, \ldots, [u_n]_i)$ and $\delta^\rho_i([u]_i) = \mathsf{f}(\delta^\rho_i([u_1]_i), \ldots, \delta^\rho_i([u_n]_i))$. Thus, this case is similar to the case $\mathsf{f} \in \Sigma_i$. Hence the result holds.

We now prove the second property, *i.e.* if $\alpha \vDash \mathsf{test}_i([u]_i)$, then $\delta^\rho_i([u]_i\alpha)\downarrow = \delta^\rho_i([u]_i\alpha\downarrow)$. Once again, we prove the results by induction on $|u|$:

Base case $|u| = 1$*:* In this case, $u \in \mathcal{N} \cup \mathcal{X}$. In both cases, we have that $[u]_i = u$ and $\mathsf{test}_i(u) = \mathsf{true}$. If $u \in \mathcal{N}$, we know that $\delta^\rho_i(u) \in \mathcal{N}$ and so $\delta^\rho_i(u)\downarrow = \delta^\rho_i(u)$. But we also have that $u\alpha\downarrow = u\alpha = u$. Thus, we conclude that $\delta^\rho_i(u\alpha)\downarrow = \delta^\rho_i(u)\downarrow = \delta^\rho_i(u) = \delta^\rho_i(u\alpha\downarrow)$. Else, if $u \in \mathcal{X}$, by hypothesis on α, we deduce that $u\alpha\downarrow = u\alpha$. Furthermore, by Lemma 6, we also know that $\delta^\rho_i(u\alpha\downarrow)\downarrow = \delta^\rho_i(u\alpha\downarrow)$. Thus, we conclude that $\delta^\rho_i(u\alpha\downarrow) = \delta^\rho_i(u\alpha\downarrow)\downarrow = \delta^\rho_i(u\alpha)\downarrow$.

Inductive step $|u| > 1$*:* In this case, we have $u = \mathsf{f}(u_1, \ldots, u_n)$. We do a case analysis on f.

Case $\mathsf{f} \in \Sigma_i$*:* In such a case, we have that $[u]_i = \mathsf{f}([u_1]_i, \ldots, [u_n]_i)$. Hence, we deduce that $\delta^\rho_i([u]_i\alpha) = \mathsf{f}(\delta^\rho_i([u_1]_i\alpha), \ldots, \delta^\rho_i([u_n]_i\alpha))$ and so $\delta^\rho_i([u]_i\alpha)\downarrow = \mathsf{f}(\delta^\rho_i([u_1]_i\alpha)\downarrow, \ldots, \delta^\rho_i([u_n]_i\alpha)\downarrow)\downarrow$. But $\mathsf{test}_i([u]_i) = \bigwedge_{j=1}^n \mathsf{test}_i([u_j]_i)$ which means that $\alpha \vDash \mathsf{test}_i([u_j]_i)$, for all j. Thus, By our inductive hypothesis on $u_1, \ldots, u_n$, we deduce that $\delta^\rho_i([u]_i\alpha)\downarrow = \mathsf{f}(\delta^\rho_i([u_1]_i\alpha\downarrow), \ldots, \delta^\rho_i([u_n]_i\alpha\downarrow))\downarrow = \delta^\rho_i(\mathsf{f}([u_1]_i\alpha\downarrow, \ldots, [u_n]_i\alpha\downarrow))\downarrow$.

Let's denote $t = \mathsf{f}([u_1]_i\alpha\downarrow, \ldots, [u_n]_i\alpha\downarrow)$. We can assume that there exists a context C built on Σ_i such that $t = C[t_1, \ldots, t_m]$ with $Fct(t) = \{t_1, \ldots, t_m\}$ and $t_1, \ldots, t_m$ are in normal form. Thus, by Lemma 1, there exists a context D (possibly a hole) such that $t\downarrow = D[t_{j_1}, \ldots, t_{j_k}]$ with $j_1, \ldots, j_k \in \{0, \ldots, m\}$ and $t_0 = n_{min}$. But since $t_1, \ldots, t_m$ are all in normal form and thanks to Lemma 6, we know that for all $k \in \{0, \ldots, m\}$, $\delta^\rho_i(t_k)$ is also in normal form and its root is not in Σ_i. Hence, we can apply Lemma 2 such that $C[\delta^\rho_i(t_1), \ldots, \delta^\rho_i(t_n)]\downarrow = D[\delta^\rho_i(t_{j_1}), \ldots, \delta^\rho_i(t_{j_k})]$. But since C and D are both built upon Σ_i, we have that $C[\delta^\rho_i(t_1), \ldots, \delta^\rho_i(t_n)]\downarrow = \delta^\rho_i(C[t_1, \ldots, t_n])\downarrow$ and $D[\delta^\rho_i(t_{j_1}), \ldots, \delta^\rho_i(t_{j_k})] = \delta^\rho_i(D[t_{j_1}, \ldots, t_{j_k}])$. Hence, we can deduce that $\delta^\rho_i(t)\downarrow = \delta^\rho_i(t\downarrow)$. But we already know that $t\downarrow = [u]_i\alpha\downarrow$ and $\delta^\rho_i(t)\downarrow = \delta^\rho_i([u]_i\alpha)\downarrow$. Thus, we can conclude that $\delta^\rho_i([u]_i\alpha)\downarrow = \delta^\rho_i([u]_i\alpha\downarrow)$.

Case $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$*:* In such a case, we have that $[u]_i = \mathsf{f}(\mathsf{tag}_i([u_1]_i), [u_2]_i)$ and $\mathsf{test}_i([u]_i) = \mathsf{test}_i([u_1]_i) \wedge \mathsf{test}_i([u_2]_i)$. By definition of E_0, we have that $[u]_i\alpha\downarrow =$

$\mathsf{f}(\mathsf{tag}_i([u_1]_i\alpha\downarrow), [u_2]_i\alpha\downarrow)$. Furthermore, we also have that $\delta_i^\rho([u]_i\alpha)\downarrow = \mathsf{f}(\mathsf{tag}_i(\delta_i^\rho([u_1]_i\alpha)\downarrow), \delta_i^\rho([u_2]_i\alpha)\downarrow)$. By our inductive hypothesis on u_1 and u_2, we have that $\delta_i^\rho([u_k]_i\alpha)\downarrow = \delta_i^\rho([u_k]_i\alpha\downarrow)$, for $k \in \{1,2\}$. Hence, we can deduce that $\delta_i^\rho([u]_i\alpha)\downarrow = \mathsf{f}(\mathsf{tag}_i(\delta_i^\rho([u_1]_i\alpha\downarrow)), \delta_i^\rho([u_2]_i\alpha\downarrow)) = \delta_i^\rho(\mathsf{f}(\mathsf{tag}_i([u_1]_i\alpha\downarrow), [u_2]_i\alpha\downarrow))$. Thus, we can conclude that $\delta_i^\rho([u]_i\alpha)\downarrow = \delta_i^\rho([u]_i\alpha\downarrow)$.

Case $\mathsf{f} = \mathsf{h}$*:* This case is analogous to the previous one and can be handled in a similar way.

Case $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}, \langle\ \rangle\}$*:* In this case, we have that $[u]_i = \mathsf{f}([u_1]_i, \ldots, [u_n]_i)$ with $n \in \{1,2\}$, and $\mathsf{test}_i([u]_i) = \wedge_{j=1}^n \mathsf{test}_i([u_j]_i)$. By definition of E_0, we have that $[u]_i\alpha\downarrow = \mathsf{f}([u_1]_i\alpha\downarrow, [u_2]_i\alpha\downarrow)$. Thus, this case is similar to the senc case and can be handled similarly.

Case $\mathsf{f} \in \{\mathsf{sdec}, \mathsf{adec}, \mathsf{check}\}$*:* In such a case, we have that $[u]_i = \mathsf{untag}_i(\mathsf{f}([u_1]_i, [u_2]_i))$ and $\mathsf{test}_i([u]_i) = (\mathsf{tag}_i(\mathsf{untag}_i(\mathsf{f}([u_1]_i, [u_2]_i))) = \mathsf{f}([u_1]_i, [u_2]_i)) \wedge \mathsf{test}_i([u_1]_i) \wedge \mathsf{test}_i([u_2]_i)$. But by hypothesis, we know that $\alpha \vDash \mathsf{test}_i([u]_i)$, thus, we have that $\mathsf{tag}_i(\mathsf{untag}_i(\mathsf{f}([u_1]_i, [u_2]_i)))\alpha\downarrow = \mathsf{f}([u_1]_i, [u_2]_i)\alpha\downarrow$. Hence, we deduce that the root function symbol f can be reduced and the root and the plain text is tag_i, *i.e.*: there exists v_1, v_2 such that

- $\mathsf{f} = \mathsf{sdec}$: $[u_1]_i\alpha\downarrow = \mathsf{senc}(\mathsf{tag}_i(v_1), v_2)$, $[u_2]_i\alpha\downarrow = v_2$ and $[u]_i\alpha\downarrow = v_1$. It implies that $\delta_i^\rho([u_1]_i\alpha\downarrow) = \mathsf{senc}(\mathsf{tag}_i(\delta_i^\rho(v_1)), \delta_i^\rho(v_2))$ and so we can deduce that $\mathsf{untag}_i(\mathsf{sdec}(\delta_i^\rho([u_1]_i\alpha\downarrow), \delta_i^\rho([u_2]_i\alpha\downarrow)))\downarrow = \delta_i^\rho(v_1) = \delta_i^\rho([u]_i\alpha\downarrow)$
- $\mathsf{f} = \mathsf{adec}$: $[u_1]_i\alpha\downarrow = \mathsf{aenc}(\mathsf{tag}_i(v_1), \mathsf{pk}(v_2))$ and $[u_2]_i\alpha\downarrow = v_2$
- $\mathsf{f} = \mathsf{check}$: $[u_1]_i\alpha\downarrow = \mathsf{sign}(\mathsf{tag}_i(v_1), v_2)$ and $[u_2]_i\alpha\downarrow = \mathsf{vk}(v_2)$.

In all cases, the following equality holds: $\mathsf{untag}_i(\mathsf{f}(\delta_i^\rho([u_1]_i\alpha\downarrow), \delta_i^\rho([u_2]_i\alpha\downarrow)))\downarrow = \delta_i^\rho([u]_i\alpha\downarrow)$. But by inductive hypothesis, we know that $\delta_i^\rho([u_k]_i\alpha\downarrow) = \delta_i^\rho([u_k]_i\alpha)\downarrow$, for $k \in \{1,2\}$. Thus, since we also have that $\delta_i^\rho([u]_i\alpha)\downarrow = \mathsf{untag}_i(\mathsf{f}(\delta_i^\rho([u_1]_i\alpha)\downarrow, \delta_i^\rho([u_2]_i\alpha)\downarrow))\downarrow$, we can conclude that $\delta_i^\rho([u]_i\alpha)\downarrow = \mathsf{untag}_i(\mathsf{f}(\delta_i^\rho([u_1]_i\alpha\downarrow), \delta_i^\rho([u_2]_i\alpha\downarrow)))\downarrow = \delta_i^\rho([u]_i\alpha\downarrow)$.

Case $\mathsf{f} = \mathsf{proj}_j$*,* $j = 1,2$*:* In this case $n = 1$, and $[u]_i = \mathsf{f}([u_1]_i)$. Since $\alpha \vDash \mathsf{test}_i([u]_i)$, we have that there exists v_1, v_2 such that $[u_1]_i\alpha\downarrow = \langle v_1, v_2\rangle$ and so $\delta_i^\rho([u]_i\alpha\downarrow) = \delta_i^\rho(v_j)$. But by inductive hypothesis, we have that $\delta_i^\rho([u_1]_i\alpha)\downarrow = \delta_i^\rho([u_1]_i\alpha\downarrow) = \langle\delta_i^\rho(v_1), \delta_i^\rho(v_2)\rangle$. Hence, $\delta_i^\rho([u]_i\alpha)\downarrow = \mathsf{f}(\delta_i^\rho([u_1]_i\alpha))\downarrow = \mathsf{f}(\delta_i^\rho([u_1]_i\alpha)\downarrow)\downarrow = \delta_i^\rho(v_j)\downarrow$. But we showed that $\delta_i^\rho(v_j) = \delta_i^\rho([u]_i\alpha\downarrow)$, thus by Lemma 6, $\delta_i^\rho(v_j)$ is in normal form and which allows us to conclude. ■

Corollary 2: Let $i \in \{a,b\}$. Let $u, v \in \mathcal{T}(\Sigma_i \cup \Sigma_0, \mathcal{N} \cup \mathcal{X})$. Let α be a ground substitution such that $fv(u) \subseteq \mathrm{dom}(\alpha)$ and for all $x \in \mathrm{dom}(\alpha)$, $x\alpha$ is in normal form. We have that : If $\alpha \vDash \mathsf{test}_i([u]_i) \wedge \mathsf{test}_i([v]_i)$ then $[u]_i\alpha\downarrow = [v]_i\alpha\downarrow$ is equivalent to $\delta_i^\rho([u]_i)\delta_i^\rho(\alpha)\downarrow = \delta_i^\rho([v]_i)\delta_i^\rho(\alpha)\downarrow$.

Proof: Thanks to Lemma 3, $[u]_i\alpha\downarrow = [v]_i\alpha\downarrow$ is equivalent to $\delta_i^\rho([u]_i\alpha\downarrow) = \delta_i^\rho([v]_i\alpha\downarrow)$. But thanks to Lemma 7, we have that $\delta_i^\rho([u]_i\alpha\downarrow) = \delta_i^\rho([u]_i\alpha)\downarrow = \delta_i^\rho([u]_i)\delta_i^\rho(\alpha)\downarrow$ and $\delta_i^\rho([v]_i\alpha\downarrow) = \delta_i^\rho([v]_i\alpha)\downarrow = \delta_i^\rho([v]_i)\delta_i^\rho(\alpha)\downarrow$. Thus, the result holds. ■

Lemma 8: Let $i \in \{a,b\}$. Let $u, v \in \mathcal{T}(\Sigma_i \cup \Sigma_0, \mathcal{N} \cup \mathcal{X})$. Let α be a ground substitution such that $fv(u) \subseteq \mathrm{dom}(\alpha)$ and for all $x \in \mathrm{dom}(\alpha)$, $x\alpha$ is in normal form. We have that :

$$\alpha \vDash \mathsf{test}_i([u]_i) \text{ is equivalent to } \delta_i^\rho(\alpha) \vDash \mathsf{test}_i(\delta_i^\rho([u]_i))$$

Proof: We prove this result by induction on $|u|$:

Base case $|u| = 1$*:* In this case, we have that $u \in \mathcal{N} \cup \mathcal{X}$, and thus $[u]_i, \delta_i^\rho([u]_i) \in \mathcal{N} \cup \mathcal{X}$. But then by definition, $\mathsf{test}_i([u]_i) = \mathsf{true}$ and $\mathsf{test}_i(\delta_i^\rho([u]_i)) = \mathsf{true}$. Thus the result trivially holds.

Inductive step $|u| > 1$*:* Then, we have that $u = \mathsf{f}(u_1, \ldots, u_n)$. We do a case analysis on f:

Case $\mathsf{f} \in \Sigma_i \cup \{\mathsf{pk}, \mathsf{vk}, \langle\ \rangle\}$*:* In this case, we have that $[u]_i = \mathsf{f}([u_1]_i, \ldots, [u_n]_i)$ and $\delta_i^\rho([u]_i) = \mathsf{f}(\delta_i^\rho([u_1]_i), \ldots, \delta_i^\rho([u_n]_i))$. Thus, we deduce that $\mathsf{test}_i([u]_i) = \bigwedge_{j=1}^n \mathsf{test}_i([u_j]_i)$ and $\mathsf{test}_i(\delta_i^\rho([u]_i)) = \bigwedge_{j=1}^n \mathsf{test}_i(\delta_i^\rho([u_j]_i))$. By inductive hypothesis on $u_1, \ldots, u_n$, the result holds.

Case $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$*:* In this case, we have that $[u]_i = \mathsf{f}(\mathsf{tag}_i([u_1]_i), [u_2]_i)$ and $\delta_i^\rho([u]_i) = \mathsf{f}(\mathsf{tag}_i(\delta_i^\rho([u_1]_i)), \delta_i^\rho([u_2]_i))$. Thus, we deduce that $\mathsf{test}_i([u]_i) = \mathsf{test}_i([u_1]_i) \wedge \mathsf{test}_i([u_2]_i)$ and $\mathsf{test}_i(\delta_i^\rho([u]_i)) = \mathsf{test}_i(\delta_i^\rho([u_1]_i)) \wedge \mathsf{test}_i(\delta_i^\rho([u_2]_i))$. By inductive hypothesis on u_1, u_2, the result holds.

Case $\mathsf{f} = \mathsf{h}$*:* This case is analogous to de previous one and can be handled in a similar way.

Case $\mathsf{f} \in \{\mathsf{sdec}, \mathsf{adec}, \mathsf{check}\}$*:* In this case, we have that $[u]_i = \mathsf{untag}_i(\mathsf{f}([u_1]_i, [u_2]_i))$ and $\delta_i^\rho([u]_i) = \mathsf{untag}_i(\mathsf{f}(\delta_i^\rho([u_1]_i), \delta_i^\rho([u_2]_i)))$. Thus, we deduce that:

- $\mathsf{test}_i([u]_i) = \mathsf{test}_i([u_1]_i) \wedge \mathsf{test}_i([u_2]_i) \wedge \mathsf{tag}_i(\mathsf{untag}_i(\mathsf{f}([u_1]_i, [u_2]_i))) = \mathsf{f}([u_1]_i, [u_2]_i)$
- $\mathsf{test}_i(\delta_i^\rho([u]_i)) = \mathsf{test}_i(\delta_i^\rho([u_1]_i)) \wedge \mathsf{test}_i(\delta_i^\rho([u_2]_i)) \wedge \mathsf{tag}_i(\mathsf{untag}_i(\mathsf{f}(\delta_i^\rho([u_1]_i), \delta_i^\rho([u_2]_i)))) = \mathsf{f}(\delta_i^\rho([u_1]_i), \delta_i^\rho([u_2]_i))$

Whether we assume that $\alpha \vDash \mathsf{test}_i([u]_i)$ or $\delta_i^\rho(\alpha) \vDash \mathsf{test}_i(\delta_i^\rho([u]_i))$, we have by inductive hypothesis that $\alpha \vDash \mathsf{test}_i([u_k]_i)$ for $k \in \{1,2\}$. Thus by Lemma 7, it implies that $\delta_i^\rho([u_k]_i\alpha\downarrow) = \delta_i^\rho([u_k]_i)\delta_i^\rho(\alpha)\downarrow$, for $k \in \{1,2\}$. We do a case analysis on f :

- $\mathsf{f} = \mathsf{sdec}$: $\alpha \vDash \mathsf{tag}_i(\mathsf{untag}_i(\mathsf{f}([u_1]_i, [u_2]_i))) = \mathsf{f}([u_1]_i, [u_2]_i)$ is equivalent to there exists v_1, v_2 such that $[u_1]_i\alpha\downarrow = \mathsf{senc}(\mathsf{tag}_i(v_1), v_2)$ and $[u_2]_i\alpha\downarrow = v_2$. But by Lemma 3, it is equivalent to $\delta_i^\rho([u_1]_i\alpha\downarrow) = \mathsf{senc}(\mathsf{tag}_i(\delta_i^\rho(v_1)), \delta_i^\rho(v_2))$ and

$\delta_i^\rho([u_2]_i\alpha\downarrow) = \delta_i^\rho(v_2)$. Thus, it is equivalent to $\delta_i^\rho([u_1]_i)\delta_i^\rho(\alpha)\downarrow = \mathsf{senc}(\mathsf{tag}_i(\delta_i^\rho(v_1)), \delta_i^\rho(v_2))$ and $\delta_i^\rho([u_2]_i)\delta_i^\rho(\alpha)\downarrow = \delta_i^\rho(v_2)$. Hence it is equivalent to $\delta_i^\rho(\alpha) \vDash \mathsf{tag}_i(\mathsf{untag}_i(\mathsf{f}(\delta_i^\rho([u_1]_i), \delta_i^\rho([u_2]_i)))) = \mathsf{f}(\delta_i^\rho([u_1]_i), \delta_i^\rho([u_2]_i))$

- $\mathsf{f} = \mathsf{adec}$ and $\mathsf{f} = \mathsf{check}$: Similar to case $\mathsf{f} = \mathsf{sdec}$.

Case $\mathsf{f} \in \{\mathsf{proj}_1, \mathsf{proj}_2\}$*:* In this case, we have that $[u]_i = \mathsf{f}([u_1]_i)$ and $\delta_i^\rho([u]_i) = \mathsf{f}(\delta_i^\rho([u_1]_i))$. Thus, we deduce that :

- $\mathsf{test}_i([u]_i) = \mathsf{test}_i([u_1]_i) \wedge \langle \mathsf{proj}_1([u_1]_i), \mathsf{proj}_2([u_1]_i)\rangle = [u_1]_i$
- $\mathsf{test}_i(\delta_i^\rho([u]_i)) = \mathsf{test}_i(\delta_i^\rho([u_1]_i)) \wedge \langle \mathsf{proj}_1(\delta_i^\rho([u_1]_i)), \mathsf{proj}_2(\delta_i^\rho([u_1]_i))\rangle = \delta_i^\rho([u_1]_i)$

Whether we assume that $\alpha \vDash \mathsf{test}_i([u]_i)$ or $\delta_i^\rho(\alpha) \vDash \mathsf{test}_i(\delta_i^\rho([u]_i))$, we have by inductive hypothesis that $\alpha \vDash \mathsf{test}_i([u_1]_i)$. Thus by Lemma 7, it implies that $\delta_i^\rho([u_1]_i\alpha\downarrow) = \delta_i^\rho([u_1]_i)\delta_i^\rho(\alpha)\downarrow$.

But $\alpha \vDash \langle \mathsf{proj}_1([u_1]_i), \mathsf{proj}_2([u_1]_i)\rangle = [u_1]_i$ is equivalent to there exists v_1, v_2 such that $[u_1]_i\alpha\downarrow = \langle v_1, v_2\rangle$, which is equivalent to $\delta_i^\rho([u_1]_i\alpha\downarrow) = \langle \delta_i^\rho(v_1), \delta_i^\rho(v_2)\rangle$ thanks to Lemma 3. We showed that it is equivalent to $\delta_i^\rho([u_1]_i)\delta_i^\rho(\alpha)\downarrow = \langle \delta_i^\rho(v_1), \delta_i^\rho(v_2)\rangle$, which allows us to conclude that $\alpha \vDash \langle \mathsf{proj}_1([u_1]_i), \mathsf{proj}_2([u_1]_i)\rangle = [u_1]_i$ is equivalent $\delta_i^\rho(\alpha) \vDash \langle \mathsf{proj}_1(\delta_i^\rho([u_1]_i)), \mathsf{proj}_2(\delta_i^\rho([u_1]_i))\rangle = \delta_i^\rho([u_1]_i)$. ■

For a term u that does not contain any tag, we defined in Section IV, a way to construct a term that is properly tagged (*i.e.* $[u]_i$). Hence, for a term properly tagged, we would never have $\mathsf{senc}(n, k)$ where n and k are both nonces, for example. Instead, we would have $\mathsf{senc}(\mathsf{tag}_i(n), k)$. However, even if we can force the processes to properly tag their term, we do not have any control on what the intruder can build. Typically, if the intruder is able to deduce n and k, he is allowed to send to a process the term $\mathsf{senc}(n, k)$. Thus, we want to define the notion of *flawed tagged term*.

Definition 10: Let u be a ground term in normal form. We define the flawed tagged subterm of u, denoted $\mathsf{Flawed}(u)$, recursively on u :

- $u = \mathsf{f}(\mathsf{tag}_i(u_1), u_2)$, for $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$ and $i \in \{a, b\}$: $\mathsf{Flawed}(u) = \mathsf{Flawed}(u_1) \cup \mathsf{Flawed}(u_2)$.
- $u = \mathsf{h}(\mathsf{tag}_i(v))$, for $i \in \{a, b\}$: $\mathsf{Flawed}(u) = \mathsf{Flawed}(v)$.
- $u = \mathsf{f}(u_1, \ldots, u_n)$ for $\mathsf{f} \in \Sigma_a \cup \Sigma_b \cup \Sigma_{\mathsf{tag}_a} \cup \Sigma_{\mathsf{tag}_b} \cup \{\langle\ \rangle\}$: $\mathsf{Flawed}(u) = \mathsf{Flawed}(u_1) \cup \cdots \cup \mathsf{Flawed}(u_n)$.
- $u = \mathsf{f}(v)$ for $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}\}$: If $v \in \mathcal{N}$ then $\mathsf{Flawed}(u) = \emptyset$ else $\mathsf{Flawed}(u) = \{u\} \cup \mathsf{Flawed}(v)$.
- $u \in \mathcal{N}$: $\mathsf{Flawed}(u) = \emptyset$.
- else we have that $u = \mathsf{f}(u_1, \ldots, u_n)$ and $\mathsf{Flawed}(u) = \{u\} \cup \bigcup_{i=1}^n \mathsf{Flawed}(u_i)$.

Lemma 9: Let $i \in \{a, b\}$. Let $u \in \mathcal{T}(\Sigma_i \cup \Sigma_0, \mathcal{N} \cup \mathcal{X})$ such that for all $v \in st(u)$, $\mathsf{root}(v) = \mathsf{f}$ with $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}\}$ implies that there exists $v' \in \mathcal{N}$ such that $v = \mathsf{f}(v')$. Let α be a ground substitution such that $fv(u) \subseteq \mathrm{dom}(\alpha)$ and for all $x \in \mathrm{dom}(\alpha)$, $x\alpha$ is in normal form. We have that if $\alpha \vDash \mathsf{test}_i([u]_i)$ then for all $t \in \mathsf{Flawed}([u]_i\alpha\downarrow)$, there exists $x \in fv([u]_i)$ such that $t \in \mathsf{Flawed}(x\alpha)$.

Proof: We prove the result by induction on $|u|$.

Base case $|u| = 1$*:* In this case, we have that $u \in \mathcal{X} \cup \mathcal{N}$ and so $[u]_i = u$. If $u \in \mathcal{N}$, then $u\alpha \in \mathcal{N}$ and $[u]_i\alpha\downarrow \in \mathcal{N}$, which means that $\mathsf{Flawed}([u]_i\alpha\downarrow) = \emptyset$. Thus the results holds. Else $u \in \mathcal{X}$ and so $[u]_i = u \in \mathrm{dom}(\alpha)$ which means that the result trivially holds.

Inductive step $|u| > 1$*:* Then, $u = \mathsf{f}(u_1, \ldots, u_n)$. We do a case analysis on f.

Case $\mathsf{f} \in \Sigma_i$*:* In this case, $[u]_i = \mathsf{f}([u_1]_i, \ldots, [u_n]_i)$ and $[u]_i\alpha\downarrow = \mathsf{f}([u_1]_i\alpha\downarrow, \ldots, [u_n]_i\alpha\downarrow)\downarrow$. By definition, we know that for all $t \in \mathsf{Flawed}([u]_i\alpha\downarrow)$, $\mathsf{root}(t) \notin \Sigma_a \cup \Sigma_b$. Thus, thanks to Lemma 1, we can deduce that for all $t \in \mathsf{Flawed}([u]_i\alpha\downarrow)$, there exists $k \in \{1, \ldots, n\}$ such that $t \in \mathsf{Flawed}([u_k]_i\alpha\downarrow)$. By hypothesis, $\alpha \vDash \mathsf{test}_i([u]_i)$ and so $\alpha \vDash \mathsf{test}_i([u_k]_i)$. Thus, by inductive hypothesis, we know that there exists $x \in fv([u_k]_i)$ such that $t \in st(x\alpha)$. But $x \in fv([u_k]_i)$ implies $x \in fv([u]_i)$, thus the result holds.

Case $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$*:* In this case, $[u]_i = \mathsf{f}(\mathsf{tag}_i([u_1]_i), [u_2]_i)$. Furthermore, $[u]_i\alpha\downarrow = \mathsf{f}(\mathsf{tag}_i([u_1]_i\alpha\downarrow), [u_2]_i\alpha\downarrow)$. At last, $\alpha \vDash \mathsf{test}_i([u]_i)$ implies that $\alpha \vDash \mathsf{test}_i([u_k]_i)$, for all $k = 1, 2$. But, by definition, $\mathsf{Flawed}([u]_i\alpha\downarrow) = \mathsf{Flawed}([u_1]_i\alpha\downarrow) \cup \mathsf{Flawed}([u]_i\alpha\downarrow)$ and so by our inductive hypothesis on u_1 and u_2, the result holds.

Case $\mathsf{f} = \mathsf{h}$*:* This case is analogous to the previous one and can be handled in a similar way.

Case $\mathsf{f} = \langle\ \rangle$*:* In this case, $[u]_i = \mathsf{f}([u_1]_i, [u_2]_i)$. Furthermore, $[u]_i\alpha\downarrow = \mathsf{f}([u_1]_i\alpha\downarrow, [u_2]_i\alpha\downarrow)$. At last, $\alpha \vDash \mathsf{test}_i([u]_i)$ implies that $\alpha \vDash \mathsf{test}_i([u_k]_i)$, for all $k = 1, 2$. But, by definition, $\mathsf{Flawed}([u]_i\alpha\downarrow) = \mathsf{Flawed}([u_1]_i\alpha\downarrow) \cup \mathsf{Flawed}([u]_i\alpha\downarrow)$ and so by our inductive hypothesis on u_1 and u_2, the result holds.

Case $\mathsf{f} = \{\mathsf{vk}, \mathsf{pk}\}$*:* In this case, we know by hypothesis that $u = \mathsf{f}(v)$ with $v \in \mathcal{N}$. Thus $[u]_i = u$ and $\mathsf{Flawed}(u) = \emptyset$. Thus the result trivially holds.

Case $\mathsf{f} \in \{\mathsf{sdec}, \mathsf{adec}, \mathsf{check}\}$*:* In this case, we have that $[u]_i = \mathsf{untag}_i(\mathsf{f}([u_1]_i, [u_2]_i))$ and $\mathsf{test}_i([u]_i) = \mathsf{test}_i([u_1]_i) \wedge \mathsf{test}_i([u_2]_i) \wedge \mathsf{tag}_i(\mathsf{untag}_i([u]_i)) = [u]_i$. But by hypothesis, we know that $\alpha \vDash \mathsf{test}_i([u]_i)$ and more specifically $\mathsf{tag}_i(\mathsf{untag}_i([u]_i))\alpha\downarrow = [u]_i\alpha\downarrow$. It implies that there exists v_1, v_2 such that $[u_1]_i\alpha\downarrow = \mathsf{g}(\mathsf{tag}_i(v_1), v_2)$ and $[u]_i\alpha\downarrow = v_1$, with $\mathsf{g} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$. Thus, for all $t \in \mathsf{Flawed}([u]_i\alpha\downarrow)$, $t \in \mathsf{Flawed}([u_1]_i\alpha\downarrow)$. Since $\alpha \vDash \mathsf{test}_i([u_1]_i)$, the result holds by inductive hypothesis.

Case $\mathsf{f} = \mathsf{proj}_j$, $j \in \{1, 2\}$*:* In this case, we have that $[u]_i = \mathsf{f}([u_1]_i)$ and $\mathsf{test}_i([u]_i) = \mathsf{test}_i([u_1]_i) \wedge \langle \mathsf{proj}_1([u_1]_i), \mathsf{proj}_2([u_1]_i)\rangle = [u_1]_i$. Hence, $\alpha \vDash \mathsf{test}_i([u]_i)$ implies that there exist v_1, v_2 such that $[u_1]_i\alpha\downarrow = \langle v_1, v_2\rangle$ and $[u]_i\alpha\downarrow = v_j$. Thus, for all $t \in \mathsf{Flawed}([u]_i\alpha\downarrow)$,

$t \in \mathsf{Flawed}([u_1]_i\alpha\downarrow)$. Since $\alpha \vDash \mathsf{test}_i([u_1]_i)$ by hypothesis, our inductive hypothesis allows us to conclude. ■

Corollary 3: Let $i \in \{a, b\}$. Let $u \in \mathcal{T}(\Sigma_i \cup \Sigma_0, \mathcal{N} \cup \mathcal{X})$ such that for all $v \in st(u)$, $\mathsf{root}(v) = \mathsf{f}$ with $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}\}$ implies that there exists $v' \in \mathcal{N}$ such that $v = \mathsf{f}(v')$. Let α be a ground substitution such that $fv(u) \subseteq \mathrm{dom}(\alpha)$ and for all $x \in \mathrm{dom}(\alpha)$, $x\alpha$ is in normal form. We have that if $\delta_i^\rho(\alpha) \vDash \mathsf{test}_i(\delta_i^\rho([u]_i))$, then for all $t \in \mathsf{Flawed}(\delta_i^\rho([u]_i)\delta_i^\rho(\alpha)\downarrow)$, there exists $x \in fv(\delta_i^\rho([u]_i)$ such that $t \in \mathsf{Flawed}(x\delta_i^\rho(\alpha))$.

Proof: (sketch) This result is a corollary of Lemma 9. Actually, this can be proved in a similar way since the transformation δ_i^ρ does not change the structure of the term. ■

E. Frame of a tagged process

In this subsection, we will focus on the frame of a trace for a tagged process. Let $(\mathcal{E}; \mathcal{P}; \Phi)$ be an extended process such that $\Phi = \{w_1 \rhd u_1, \ldots, w_n \rhd u_n\}$. We define a lexicographic measurement on terms M, denoted $\mathcal{M}(M)$, where $fv(M) \subseteq \mathrm{dom}(\Phi)$ and $fn(M) \cap \mathcal{E} = \emptyset$ such that : $\mathcal{M}(M) = (\max\{i \mid w_i \in fv(M)\}, |M|)$.

Definition 11: Let $(\mathcal{E}; \mathcal{P}; \Phi)$ be a closed extended process. Let's denote $\Phi = \{w_1 \rhd u_1, \ \ldots, \ w_n \rhd u_n\}$. We say that $\mathtt{new}\ \mathcal{E}.\Phi$ is *well-tagged* if for all $i \in \{1, \ldots, n\}$, u_i is well-tagged for i, *i.e.* there exists a term v_i, a substitution α and $c \in \{a, b\}$ such that:

- for all $\mathsf{vk}(t), \mathsf{pk}(t') \in st(v_i)$, $t, t' \in \mathcal{N}$
- $u_i = [v_i]_c\alpha$; and
- $\alpha \vDash \mathsf{test}_c([v_i]_c)$; and
- for all $x \in \mathrm{dom}(\alpha)$, either there exists M such that $fv(M) \subseteq \{w_1, \ldots, w_{i-1}\}$, $fn(M) \cap \mathcal{E} = \emptyset$ and $M\Phi = x\alpha$; or v_i is not a variable and $x\alpha$ is well-tagged for i.

Lemma 10: Let $(\mathcal{E}; \mathcal{P}; \Phi)$ be an extended process such that $\mathtt{new}\ \mathcal{E}.\Phi$ is well-tagged. Let's denote $\Phi = \{w_1 \rhd u_1; \ \ldots; \ w_n \rhd u_n\}$. We have that for all $i \in \{1, \ldots, n\}$, for all $t \in \mathsf{Flawed}(u_i\downarrow)$, there exists M such that $fv(M) \subseteq \{w_1, \ldots, w_{i-1}\}$, $fn(M) \cap \mathcal{E} = \emptyset$ and $t \in \mathsf{Flawed}(M\Phi\downarrow)$.

Proof: We prove this result for any ground term u well-tagged for $i \in \{1, \ldots, n\}$. By definition, u being well-tagged for $i \in \{1, \ldots, n\}$ implies that there exists a term v, a substitution α and $c \in \{a, b\}$ such that:

- for all $\mathsf{vk}(t), \mathsf{pk}(t') \in st(v)$, $t, t' \in \mathcal{N}$
- $u = [v]_c\alpha$; and
- $\alpha \vDash \mathsf{test}_c([v]_c)$; and
- for all $x \in \mathrm{dom}(\alpha)$, either there exists M such that $fv(M) \subseteq \{w_1, \ldots, w_{i-1}\}$, $fn(M) \cap \mathcal{E} = \emptyset$ and $M\Phi = x\alpha$; or v is not a variable and $x\alpha$ is well-tagged for i.

The proof is done by induction $|u|$.

Base case $|u| = 1$*:* In this case, $u \in \mathcal{N}$ which implies $u\downarrow = u$, and thus by definition $\mathsf{Flawed}(u\downarrow) = \emptyset$. Hence the result trivially holds.

Inductive step $|u| > 1$*:* Let $t \in \mathsf{Flawed}(u\downarrow)$. Since for all $\mathsf{vk}(t), \mathsf{pk}(t') \in st(v)$, $t, t' \in \mathcal{N}$, and $u = [v]_c\alpha$ and $\alpha \vDash \mathsf{test}_c([v]_c)$, we can apply Lemma 9 to v and $\alpha\downarrow$. Thus, we have that there exists $x \in fv([v]_c)$ such that $t \in \mathsf{Flawed}(x\alpha\downarrow)$. But since u is well-tagged for i, we know that either there exists M such that $fv(M) \subseteq \{w_1, \ldots, w_{i-1}\}$, $fn(M) \cap \mathcal{E} = \emptyset$ and $M\Phi = x\alpha$. Hence $t \in \mathsf{Flawed}(x\alpha\downarrow) = \mathsf{Flawed}(M\Phi\downarrow)$ and so the result holds in such a case. Or, v is not a variable and thus nor is $[v]_c$. Moreover, $x\alpha$ is well-tagged for i. Now because v is not a variable and $x \in fv([v]_c)$, we have that $|x\alpha| < |[v]_c\alpha| = |u|$, we conclude by applying our inductive hypothesis. ■

Lemma 11: Let $(\mathcal{E}, \mathcal{P}, \Phi)$ be a closed process such that $\mathtt{new}\ \mathcal{E}.\Phi$ is well-tagged. We have that for all M such that $fn(M) \cap \mathcal{E} = \emptyset$ and $fv(M) \subseteq \mathrm{dom}(\Phi)$, for all $\mathsf{f}(u_1, \ldots, u_n) \in \mathsf{Flawed}(M\Phi\downarrow)$, there exists $M_1, \ldots, M_n$ such that $fv(M_i) \subseteq \mathrm{dom}(\Phi)$, $fn(M_i) \cap \mathcal{E} = \emptyset$, $M_i\Phi\downarrow = u_i$, and $\mathcal{M}(M_i) < \mathcal{M}(M)$, for all $i \in \{1, \ldots, n\}$.

Proof: We prove this result by induction on $\mathcal{M}(M)$.
Base case $\mathcal{M}(M) = (0, 0)$*:* A term with $|M| = 0$ is impossible so the result trivially holds.

Inductive step $\mathcal{M}(M) = (i, 1)$*:* In this case, either we have that $M \in \mathcal{N}$ or $M = w_i$. If $M \in \mathcal{N}$, then we have $M\Phi\downarrow = M \in \mathcal{N}$ and $\mathsf{Flawed}(M\Phi\downarrow) = \emptyset$. Thus the result holds. If $M = w_i$, then by Lemma 10, for all $\mathsf{f}(t_1, \ldots, t_m) \in \mathsf{Flawed}(w_i\Phi\downarrow)$, there exists M' such that $fv(M') \subseteq \{w_1, \ldots, w_{i-1}\}$, $fn(M) \cap \mathcal{E} = \emptyset$ and $\mathsf{f}(t_1, \ldots, t_m) \in \mathsf{Flawed}(M'\Phi\downarrow)$. But $\mathcal{M}(M') < \mathcal{M}(M)$, thus by our inductive hypothesis, we can deduce that there exists $M_1, \ldots, M_m$ such that $fv(M_i) \subseteq \mathrm{dom}(\Phi)$, $fn(M_i) \cap \mathcal{E} = \emptyset$, $M_i\Phi\downarrow = t_i$ and $\mathcal{M}(M_i) < \mathcal{M}(M') < \mathcal{M}(M)$, for $i \in \{1, \ldots, m\}$.

Inductive step $\mathcal{M}(M) > (i, 1)$*:* We have that $M = \mathsf{f}(M_1, \ldots, M_n)$. Let $t = \mathsf{g}(t_1, \ldots, t_m) \in \mathsf{Flawed}(M\Phi\downarrow)$. We do a case analysis on f.

Case $\mathsf{f} \in \Sigma_\ell \cup \Sigma_{\mathsf{tag}_\ell}$, $\ell \in \{a, b\}$*:* In this case, $M\Phi\downarrow = \mathsf{f}(M_1\Phi\downarrow, \ldots, M_n\Phi\downarrow)\downarrow$. By definition, we know that for all $t \in \mathsf{Flawed}(M\Phi\downarrow)$, $\mathsf{root}(t) \notin \Sigma_\ell \cup \Sigma_{\mathsf{tag}_\ell}$. Thus, thanks to Lemma 1, we can deduce that for all $t \in \mathsf{Flawed}(M\Phi\downarrow)$, there exists $k \in \{1, \ldots, n\}$ such that $t \in \mathsf{Flawed}(M_k\Phi\downarrow)$. But $\mathcal{M}(M_k) < \mathcal{M}(M)$, thus, by inductive hypothesis, we know that there exists $M'_1, \ldots, M'_m$ such that $fv(M'_j) \subseteq \Phi$, $fn(M'_j) \cap \mathcal{E} = \emptyset$, $M'_j\Phi\downarrow = t_i$ and $\mathcal{M}(M'_j) < \mathcal{M}(M_k) < \mathcal{M}(M)$, for $j \in \{1, \ldots, m\}$. Hence the result holds.

Case $\mathsf{f} = \langle\ \rangle$*:* In such a case, $M\Phi\downarrow = \mathsf{f}(M_1\Phi\downarrow, M_2\Phi\downarrow)$. Furthermore by definition, we have $\mathsf{Flawed}(M\Phi\downarrow) = \mathsf{Flawed}(M_1\Phi\downarrow) \cup \mathsf{Flawed}(M_2\Phi\downarrow)$. Since $\mathcal{M}(M_1) < \mathcal{M}(M)$, $\mathcal{M}(M_2) < \mathcal{M}(M)$ and $t \in \mathsf{Flawed}(M_1\Phi\downarrow) \cup \mathsf{Flawed}(M_2\Phi\downarrow)$, we conclude by applying our inductive hypothesis on M_1 (or M_2).

Case $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}\}$*:* In this case, $M\Phi\downarrow = \mathsf{f}(M_1\Phi\downarrow)$. If $M_1\Phi\downarrow \in \mathcal{N}$, then we have that $\mathsf{Flawed}(M\Phi\downarrow) = \emptyset$, else

$\mathsf{Flawed}(M\Phi\downarrow) = \{M\Phi\downarrow\} \cup \mathsf{Flawed}(M_1\Phi\downarrow)$. If $t = M\Phi\downarrow$, then we have $t_1 = M_1\Phi\downarrow$. Since $\mathcal{M}(M_1) < \mathcal{M}(M)$, then the result holds; else we conclude by applying our inductive hypothesis on M_1.

Case $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$*:* In such a case, $M\Phi\downarrow = \mathsf{f}(M_1\Phi\downarrow, M_2\Phi\downarrow)$. We need to distinguish if $\mathsf{root}(M_1\Phi\downarrow) \in \{\mathsf{tag}_a, \mathsf{tag}_b\}$ or not.

If $\mathsf{root}(M_1\Phi\downarrow) \in \{\mathsf{tag}_a, \mathsf{tag}_b\}$, then there exists $\ell \in \{a, b\}$ and u_1 such that $M_1\Phi\downarrow = \mathsf{tag}_\ell(u_1)$. Thus, $\mathsf{Flawed}(M_1\Phi\downarrow) = \mathsf{Flawed}(u_1)$. But by definition, we have that $\mathsf{Flawed}(M\Phi\downarrow) = \mathsf{Flawed}(u_1) \cup \mathsf{Flawed}(M_2\Phi\downarrow)$. Thus, $t \in \mathsf{Flawed}(M\Phi\downarrow)$ implies that $t \in \mathsf{Flawed}(M_1\Phi\downarrow)$ or $t \in \mathsf{Flawed}(M_2\Phi\downarrow)$. Since $\mathcal{M}(M_1) < \mathcal{M}(M)$ and $\mathcal{M}(M_2) < \mathcal{M}(M)$, we conclude by applying our inductive hypothesis on M_1 or M_2.

Else $\mathsf{root}(M_1\Phi\downarrow) \notin \{\mathsf{tag}_a, \mathsf{tag}_b\}$. In such a case, $\mathsf{Flawed}(M\Phi\downarrow) = \mathsf{Flawed}(M_1\Phi\downarrow) \cup \mathsf{Flawed}(M_2\Phi\downarrow) \cup \{M\Phi\downarrow\}$. If $t = M\Phi\downarrow$, we have that $t_1 = M_1\Phi\downarrow$, $t_2 = M_2\Phi\downarrow$ and $\mathcal{M}(M_1) < \mathcal{M}(M)$, $\mathcal{M}(M_2) < \mathcal{M}(M)$. Thus the result holds. If $t \in \mathsf{Flawed}(M_1\Phi\downarrow) \cup \mathsf{Flawed}(M_2\Phi\downarrow)$, we conclude by applying our inductive hypothesis on M_1 or M_2.

Case $\mathsf{f} = \mathsf{h}$*:* This case is analogous to the previous one and can be handled similarly.

Case $\mathsf{f} \in \{\mathsf{sdec}, \mathsf{adec}, \mathsf{check}\}$*:* For all those functions, we have to distinguish two cases: Either f is reduced in $M\Phi\downarrow$, or not.

If f is not reduced, then we have that $M\Phi\downarrow = \mathsf{f}(M_1\downarrow, M_2\downarrow)$. Thus, by definition we have that $\mathsf{Flawed}(M\Phi\downarrow) = \{M\Phi\downarrow\} \cup \mathsf{Flawed}(M_1\Phi\downarrow) \cup \mathsf{Flawed}(M_2\Phi\downarrow)$. Thus if $t = M\Phi\downarrow$, we have that $t_1 = M_1\Phi\downarrow$, $t_2 = M_2\Phi\downarrow$ and $\mathcal{M}(M_1) < \mathcal{M}(M)$, $\mathcal{M}(M_2) < \mathcal{M}(M)$. Thus the result holds. Else if $t \in \mathsf{Flawed}(M_1\Phi\downarrow)$ or $t \in \mathsf{Flawed}(M_2\Phi\downarrow)$, since $\mathcal{M}(M_1) < \mathcal{M}(M)$, $\mathcal{M}(M_2) < \mathcal{M}(M)$, we can conclude by applying our inductive hypothesis on M_1 or M_2.

If f is reduced, then we have that $M_1\Phi\downarrow = \mathsf{f}'(u_1, u_2)$ with $M\Phi\downarrow = u_1$ and $\mathsf{f}' \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}\}$. If $\mathsf{root}(u_1) = \mathsf{tag}_\ell$, with $\ell \in \{a, b\}$ then we have that there exists u_1' such that $u_1 = \mathsf{tag}_\ell(u_1')$, $\mathsf{Flawed}(M\Phi\downarrow) = \mathsf{Flawed}(u_1')$ and $\mathsf{Flawed}(M_1\Phi\downarrow) = \mathsf{Flawed}(u_1') \cup \mathsf{Flawed}(u_2)$. Thus $\mathsf{Flawed}(M\Phi\downarrow) \subseteq \mathsf{Flawed}(M_1\Phi\downarrow)$. If $\mathsf{root}(u_1) \neq \mathsf{tag}_\ell$, for all $\ell \in \{a, b\}$, then we have that $\mathsf{Flawed}(M_1\Phi\downarrow) = \{M_1\Phi\downarrow\} \cup \mathsf{Flawed}(u_1) \cup \mathsf{Flawed}(u_2)$ and $\mathsf{Flawed}(M\Phi\downarrow) = \mathsf{Flawed}(u_1)$. Thus, we also have that $\mathsf{Flawed}(M\Phi\downarrow) \subseteq \mathsf{Flawed}(M_1\Phi\downarrow)$. Since in both cases, we have that $\mathsf{Flawed}(M\Phi\downarrow) \subseteq \mathsf{Flawed}(M_1\Phi\downarrow)$ and $\mathcal{M}(M_1) < \mathcal{M}(M)$, we can conclude by applying our inductive hypothesis on M_1. ■

Lemma 12: Let u be a ground term in normal form. We have that there exists a context C (possibly a hole) built on $\{\langle\ \rangle\}$, and $u_1, \ldots, u_m$ such that $u = C[u_1, \ldots, u_m]$, and for all $i \in \{1, \ldots, m\}$,

- either $u_i \in \mathsf{Flawed}(u)$;
- or $u_i \in Fct_{\Sigma_0}(u)$ and $\delta_a^\rho(u_i) = \delta_b^\rho(u_i)$,
- or $u_i = \mathsf{f}(n)$ for some $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}\}$ and $n \in \mathcal{N}$,
- or $u_i \in \mathcal{N}$.

Proof: Let u a ground term in normal form and let $\{v_1, \ldots, v_n\} = Fct_{\Sigma_0}(u)$. Thus there exists a context D (possibly a hole) built on Σ_0 such that $u = D[v_1, \ldots, v_n]$. We prove the result by induction on $|D|$.

Base case $|D| = 0$*:* We show that the result holds with $C = \_$. $|D| = 0$ implies that $Fct_{\Sigma_0}(u) = u$. But by Lemma 5, $u\downarrow \notin \mathrm{dom}(\rho)$ implies that $\delta_a^\rho(u) = \delta_b^\rho(u)$. But u is in normal form which means that $u\downarrow \in \mathrm{dom}(\rho)$ implies that $u \in \mathcal{N}$. Hence we have that $u \in Fct_{\Sigma_0}(u)$ and either $u \in \mathcal{N}$ or $\delta_a^\rho(u) = \delta_b^\rho(u)$ which allows us to conclude.

Inductive step $|D| > 0$*:* There exist $\mathsf{f} \in \Sigma_0$, and $v_1, \ldots, v_k$ such that $u = \mathsf{f}(u_1, \ldots, u_k)$. We do a case analysis on f.

Case $\mathsf{f} = \langle\ \rangle$*:* In such a case, there exists D_1, D_2 context (possibly holes) built on Σ_0 such that $D = \langle D_1, D_2\rangle$, $u_i = D_i[v_1^i, \ldots, v_{n_i}^i]$ and $\{v_1^i, \ldots, v_{n_i}^i\} = Fct_{\Sigma_0}(u_i)$ and $|D_i| < |D|$, for all $i \in \{1, 2\}$. By inductive hypothesis on u_1 and u_2, we have that there exists C_1 and C_2 context built on $\{\langle\ \rangle\}$ such that $u_1 = C_1[u_1^1, \ldots, u_{m_1}^1]$, $u_2 = C_2[u_1^2, \ldots, u_{m_2}^2]$ and for all i, j,

- either $u_j^i \in \mathsf{Flawed}(u_i)$, but we have that $\mathsf{Flawed}(u) = \mathsf{Flawed}(u_1) \cup \mathsf{Flawed}(u_2)$ so $u_j^i \in \mathsf{Flawed}(u)$;
- or $u_j^i \in Fct_{\Sigma_0}(u_i)$ and $\delta_a^\rho(u_j^i) = \delta_b^\rho(u_j^i)$, but we have that $Fct_{\Sigma_0}(u) = Fct_{\Sigma_0}(u_1) \uplus Fct_{\Sigma_0}(u_2)$ thus $u_j^i \in Fct_{\Sigma_0}(u)$
- or $u_j^i = \mathsf{f}(n)$ for some $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}\}$ and $n \in \mathcal{N}$,
- or $u_j^i \in \mathcal{N}$.

Thus, with the context $C = \langle C_1, C_2\rangle$, and $u = C[u_1^1, \ldots, u_{m_1}^1, u_1^2, \ldots, u_{m_2}^2]$, the result holds.

Case $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}\}$ *and* $u = \mathsf{f}(n)$ *for some* $n \in N$*:* With $C = \_$ as context, the result trivially holds.

Otherwise: By definition, we have that $\mathsf{Flawed}(u) = \{u\} \cup \bigcup_{i=1}^k \mathsf{Flawed}(u_i)$. Thus, since $u \in \mathsf{Flawed}(u)$, then with $C = \_$ as context, the result trivially holds. ■

F. Proof of Theorem 1

In this subsection, we will focus on the proof of Theorem 1. Typically, all previous subsection of the Appendix are useful for the proofs of both Theorems. Thus, this subsection is independent of subsection G. We will assume in this subsection that processes and frames are colored by a or b. Intuitively, coloring a process by a means that this process was derived from the original process A. The same way, we say that a frame element $(w \rhd u)$ of a frame is colored by a if u was output by a process derived from the original process A. We denote $\mathsf{col}(w)$ to represent the color of a frame element $(w \rhd u)$ in a frame, and $\mathsf{col}(P)$ for the color of a process. We will assume that processes of different colors use different channels.

Lemma 13: let $\mathcal{E}$ be a set of names and $\Phi = \{w_1 \rhd u_1, \ldots, w_n \rhd u_n\}$ such that $\texttt{new}\ \mathcal{E}.\Phi$ is a well-tagged frame in normal form. Let renaming ρ such that $\mathrm{dom}(\rho) \cup \mathrm{img}(\rho) \subseteq \mathcal{E}$ and $\mathrm{dom}(\rho) \cap fn(\Phi) = \emptyset$. If one of the two following conditions is satisfied: (a) for all $k \in \mathrm{img}(\rho)$, $\Phi \not\vdash k$, $\Phi \not\vdash \mathsf{pk}(k)$ and $\Phi \not\vdash \mathsf{vk}(k)$, (b) for all $k \in \mathrm{img}(\rho) \cup \mathrm{dom}(\rho)$, $\delta^\rho(\Phi) \not\vdash k$, $\delta^\rho(\Phi) \not\vdash \mathsf{pk}(k)$ and $\delta^\rho(\Phi) \not\vdash \mathsf{vk}(k)$; then we have for all M such that $fv(M) \subseteq \mathrm{dom}(\Phi)$ and $fn(M) \cap \mathcal{E} = \emptyset$, for all $i \in \{a, b\}$, $\delta_i^\rho(M\Phi\downarrow) = M\delta^\rho(\Phi)\downarrow$.

Proof: We prove this result by induction on $\mathcal{M}(M)$:

Base case $\mathcal{M}(M) = (0, 0)$*:* There exists no term M such that $|M| = 0$, thus the result holds.

Inductive step $\mathcal{M}(M) > (0, 0)$*:* We first prove there exists $i \in \{a, b\}$ such that $\delta_i^\rho(M\Phi\downarrow) = M\delta^\rho(\Phi)\downarrow$ and then we show that $\delta_a^\rho(M\Phi\downarrow) = \delta_b^\rho(M\Phi\downarrow)$

Assume first that $|M| = 1$, i.e. either $M \in \mathcal{N}$ or there exists $j \in \{1, \ldots, n\}$ such that $M = w_j$.

Let us first suppose $M \in \mathcal{N}$. In that case, $M\Xi\downarrow = M$ for any susbsitution. Now, because by hypotheses $\mathrm{dom}(\rho) \cup \mathrm{img}(\rho) \subseteq \mathcal{E}$, and $fn(M) \cap \mathcal{E} = \emptyset$, we necessarily have $M\downarrow \neq k\rho$ for any k. Thus, by definition $\delta_j^\rho(M\Phi\downarrow) = \delta_j^\rho(M) = M = M\delta^\rho(M\Phi)\downarrow$ for any $d \in \{a, b\}$.

Let's now assume that there exists $j \in \{1, \ldots, n\}$ such that $M = w_j$, and suppose that $\mathsf{col}(w_j) = i \in \{a, b\}$. According to the definition of $\delta^\rho(\Phi)$, we have that $w_j\delta^\rho(\Phi) = \delta_i^\rho(w_j\Phi)$. Since u_j is in normal form, then by Lemma 6, we know that $\delta_i^\rho(w_j\Phi)$ is also in normal form. Thus, we have that $\delta_i^\rho(M\Phi\downarrow) = M\delta^\rho(\Phi)\downarrow$.

Otherwise, if $|M| > 1$, then there exists a symbol f and $M_1, \ldots, M_n$ such that $M = \mathsf{f}(M_1, \ldots, M_n)$. We do a case analysis on f.

Case $\mathsf{f} \in \Sigma_\ell \cup \Sigma_{\mathsf{tag}_\ell}$, $\ell \in \{a, b\}$*:* In this case, let $t = \mathsf{f}(M_1\Phi\downarrow, \ldots, M_n\Phi\downarrow)$. Since $\mathsf{f} \in \Sigma_\ell$, then there exists a context C built upon Σ_ℓ such that $t = C[u_1, \ldots, u_m]$ and $u_1, \ldots, u_m$ are factor of t in normal form. By Lemma 1, we know that there exists a context D (possibly a hole) over Σ_0 such that $t\downarrow = D[u_{i_1}, \ldots, u_{i_k}]$ with $i_1, \ldots, i_k \in \{0, \ldots, m\}$ and $u_0 = n_{min}$. But thanks to Lemma 2, 3 and 6, we also that $C[\delta_\ell^\rho(u_1), \ldots, \delta_\ell^\rho(u_m)]\downarrow = D[\delta_\ell^\rho(u_{i_1}), \ldots, \delta_\ell^\rho(u_{i_k})]$. But C and D are both built on $\Sigma_\ell \cup \Sigma_{\mathsf{tag}_\ell}$, thus by definition of δ_ℓ^ρ, we have that $\delta_\ell^\rho(t)\downarrow = C[\delta_\ell^\rho(u_1), \ldots, \delta_\ell^\rho(u_m)]$ and $\delta_\ell^\rho(t\downarrow) = D[\delta_\ell^\rho(u_{i_1}), \ldots, \delta_\ell^\rho(u_{i_k})]$. Hence, the equality, $\delta_\ell^\rho(t\downarrow) = \delta_\ell^\rho(t)\downarrow$, holds. But $t\downarrow = M\Phi\downarrow$ which means that $\delta_\ell^\rho(M\Phi\downarrow) = \delta_\ell^\rho(t)\downarrow$.

At last $\delta_\ell^\rho(t)\downarrow = \delta_\ell^\rho(\mathsf{f}(M_1\Phi\downarrow, \ldots, M_n\Phi\downarrow))\downarrow = \mathsf{f}(\delta_\ell^\rho(M_1\Phi\downarrow), \ldots, \delta_\ell^\rho(M_n\Phi\downarrow))\downarrow$. Thanks to $\mathcal{M}(M_1) < \mathcal{M}(M)$, ..., $\mathcal{M}(M_n) < \mathcal{M}(M)$, applying our inductive hypothesis on $M_1, \ldots, M_n$ gives us $\delta_\ell^\rho(t)\downarrow = \mathsf{f}(M_1\delta^\rho(\Phi)\downarrow, \ldots, M_n\delta^\rho(\Phi)\downarrow)\downarrow = \mathsf{f}(M_1, \ldots, M_n)\delta^\rho(\Phi)\downarrow$. Thus we can conclude that $\delta_\ell^\rho(M\Phi\downarrow) = \delta_\ell^\rho(t)\downarrow = M\delta^\rho(\Phi)\downarrow$.

Case $\mathsf{f} \in \Sigma_0 \backslash \{\mathsf{sdec}, \mathsf{adec}, \mathsf{check}\}$*:* In this case, we have that $M\Phi\downarrow = \mathsf{f}(M_1\Phi\downarrow, \ldots, M_n\Phi\downarrow)$. By applying our inductive hypothesis on $M_1, \ldots, M_n$, we have $\delta_a^\rho(M_k\Phi\downarrow) = \delta_b^\rho(M_k\Phi\downarrow)$, for all $k \in \{1, \ldots, n\}$. Thus we have that $\delta_i^\rho(M\Phi\downarrow) = \mathsf{f}(\delta_j^\rho(M_1\Phi\downarrow), \ldots, \delta_j^\rho(M_n\Phi\downarrow))$ with $j \in \{a, b\}$, for all $i \in \{a, b\}$. Thus by applying again our inductive hypothesis on $M_1, \ldots, M_n$, we have that $\delta_i^\rho(M\Phi\downarrow) = \mathsf{f}(M_1\delta^\rho(\Phi)\downarrow, \ldots, M_n\delta^\rho(\Phi)\downarrow) = M\delta^\rho(\Phi)\downarrow$.

Case h*:* This case is analogous to the previous one and can be handled similarly.

Case $\mathsf{f} \in \{\mathsf{sdec}, \mathsf{adec}, \mathsf{check}\}$*:* If we first assume that the root occurence f is not reduced in $M\Phi\downarrow$ then the proof is similar to the previous case. Thus, we focus on the case where the root occurence of f is reduced. In this case, there exists v_1, v_2 such that

- $M_1\Phi\downarrow = \mathsf{senc}(v_1, v_2)$, $M_2\Phi\downarrow = v_2$ and $M\Phi\downarrow = v_1$. According to the definition of δ^ρ, we know that there exists $i \in \{a, b\}$ such that $\delta_i^\rho(\mathsf{senc}(v_1, v_2)) = \mathsf{senc}(\delta_i^\rho(v_1), \delta_i^\rho(v_2))$. For such i, we have that $\mathsf{sdec}(\delta_i^\rho(M_1\Phi\downarrow), \delta_i^\rho(M_2\Phi\downarrow))\downarrow = \delta_i^\rho(M\Phi\downarrow)$. But by applying our inductive hypothesis on M_1 and M_2, we obtain $\delta_i^\rho(M\Phi\downarrow) = \mathsf{sdec}(M_1\delta^\rho(\Phi)\downarrow, M_2\delta^\rho(\Phi)\downarrow)\downarrow = M\delta^\rho(\Phi)\downarrow$.
- $M_1\Phi\downarrow = \mathsf{aenc}(v_1, \mathsf{pk}(v_2))$, $M_2\Phi\downarrow = v_2$ and $M\Phi\downarrow = v_1$: According to the definition of δ^ρ, we know that there exists $i \in \{a, b\}$ such that $\delta_i^\rho(\mathsf{aenc}(v_1, \mathsf{pk}(v_2))) = \mathsf{aenc}(\delta_i^\rho(v_1), \mathsf{pk}(\delta_i^\rho(v_2)))$. For such i, we have that $\mathsf{adec}(\delta_i^\rho(M_1\Phi\downarrow), \delta_i^\rho(M_2\Phi\downarrow))\downarrow = \delta_i^\rho(M\Phi\downarrow)$. But by applying our inductive hypothesis on M_1 and M_2, we obtain $\delta_i^\rho(M\Phi\downarrow) = \mathsf{adec}(M_1\delta^\rho(\Phi)\downarrow, M_2\delta^\rho(\Phi)\downarrow)\downarrow = M\delta^\rho(\Phi)\downarrow$.
- $M_1\Phi\downarrow = \mathsf{sign}(v_1, v_2)$, $M_2\Phi\downarrow = \mathsf{vk}(v_2)$ and $M\Phi\downarrow = v_1$: According to the definition of δ^ρ, we know that there exists $i \in \{a, b\}$ such that $\delta_i^\rho(\mathsf{sign}(v_1, v_2)) = \mathsf{sign}(\delta_i^\rho(v_1), \delta_i^\rho(v_2))$. For such i, we have that $\mathsf{adec}(\delta_i^\rho(M_1\Phi\downarrow), \delta_i^\rho(M_2\Phi\downarrow))\downarrow = \delta_i^\rho(M\Phi\downarrow)$. But by applying our inductive hypothesis on M_1 and M_2, we obtain $\delta_i^\rho(M\Phi\downarrow) = \mathsf{adec}(M_1\delta^\rho(\Phi)\downarrow, M_2\delta^\rho(\Phi)\downarrow)\downarrow = M\delta^\rho(\Phi)\downarrow$.

It remains to prove that $\delta_a^\rho(M\Phi\downarrow) = \delta_b^\rho(M\Phi\downarrow)$. We proved that there exists $i_0 \in \{a, b\}$ such that $\delta_{i_0}^\rho(M\Phi\downarrow) = M\delta^\rho(\Phi)\downarrow$. But by Lemma 12, we know that there exists a context C built over $\{\langle\rangle\}$, and $v_1, \ldots, v_m$ terms such that $M\Phi\downarrow = C[v_1, \ldots, v_m]$ and for all $i \in \{1, \ldots, m\}$:

- either $v_i \in \mathsf{Flawed}(M\Phi\downarrow)$
- or $v_i \in Fct_{\Sigma_0}(M\Phi\downarrow)$ and $\delta_a^\rho(v_i) = \delta_b^\rho(v_i)$.
- or $v_i = \mathsf{f}(n)$ for some $f \in \{\mathsf{pk}, \mathsf{vk}\}$ and $n \in \mathcal{N}$,
- or $v_i \in \mathcal{N}$.

First of all, note that C being built upon $\{\langle\rangle\}$ means that v_i is deducible in Φ, for all $i \in \{1, \ldots, m\}$. Furthermore, since $C[v_1, \ldots, v_m]$ is in normal form, $\delta_{i_0}^\rho(M\Phi\downarrow) = C[\delta_{i_0}^\rho(v_1), \delta_{i_0}^\rho(v_m)]$. But we previously proved that $\delta_{i_0}^\rho(M\Phi\downarrow) = M\delta^\rho(\Phi)\downarrow$, thus $\delta_{i_0}^\rho(v_i)$ is deducible from $\delta^\rho(\Phi)$, for all $i \in \{1, \ldots, m\}$.

Case $v_i \in \mathsf{Flawed}(M\Phi\downarrow)$*:* There exists $w_1, \ldots, w_\ell$ terms and a function symbol f such that $v_i = \mathsf{f}(w_1, \ldots, w_\ell)$. By

Lemma 11, there exists $N_1, \ldots, N_\ell$ such that for all $k \in \{1, \ldots, \ell\}$, $\mathcal{M}(N_k) < \mathcal{M}(M)$ and $N_k\Phi\downarrow = u_k$. Hence, by applying inductive hypothesis on $N_1, \ldots, N_\ell$, we obtain that $\delta_a^\rho(N_k\Phi\downarrow) = \delta_b^\rho(N_k\Phi\downarrow)$, for all $k \in \{1, \ldots, \ell\}$. Thus, thanks to v_i being in normal form, we can conclude that $\delta_a^\rho(v_i) = \delta_b^\rho(v_i)$.

Case $v_i \in Fct_{\Sigma_0}(M\Phi\downarrow)$*:* we also have $\delta_a^\rho(v_i) = \delta_b^\rho(v_i)$.

Case $v_i = \mathsf{f}(n)$ *for some* $f \in \{\mathsf{pk}, \mathsf{vk}\}$ *and* $n \in \mathcal{N}$*:* By hypothesis, we know that either $\Phi \not\vdash \mathsf{f}(k)$, for all $k \in \mathrm{img}(\rho)$; or $\delta^\rho(\Phi) \not\vdash \mathsf{f}(k)$, for all $k \in \mathrm{img}(\rho) \cup \mathrm{dom}(\rho)$. Since we showed that v_i is deducible from Φ and $\delta_{i_0}^\rho(v_i)$ is deducible from $\delta^\rho(\Phi)$, both hypotheses imply that $n \notin \mathrm{img}(\rho)$ and so $\delta_a^\rho(v_i) = \delta_b^\rho(v_i)$.

Case $v_i \in \mathcal{N}$*:* By hypothesis we know that either $\Phi \not\vdash k$, for all $k \in \mathrm{img}(\rho)$; or $\delta^\rho(\Phi) \not\vdash k$, for all $k \in \mathrm{img}(\rho) \cup \mathrm{dom}(\rho)$. Since we showed that v_i is deducible from Φ and $\delta_{i_0}^\rho(v_i)$ is deducible from $\delta^\rho(\Phi)$, both hypotheses imply that $v_i \notin \mathrm{img}(\rho)$ and so $\delta_a^\rho(v_i) = \delta_b^\rho(v_i)$. ■

Corollary 4: Let $\mathcal{E}$ be a set of names. Let Φ such that $\mathtt{new}\ \mathcal{E}.\Phi$ is a well-tagged frame in normal form. Let ρ a renaming such that $\mathrm{dom}(\rho) \cup \mathrm{img}(\rho) \subseteq \mathcal{E}$ and $\mathrm{dom}(\rho) \cap fn(\Phi) = \emptyset$. The two following properties are equivalent:

- for all $k \in \mathrm{img}(\rho) \cup \mathrm{dom}(\rho)$, $\mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi) \not\vdash k$, $\mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi) \not\vdash \mathsf{pk}(k)$, $\mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi) \not\vdash \mathsf{vk}(k)$
- for all $k \in \mathrm{img}(\rho)$, $\mathtt{new}\ \mathcal{E}.\Phi \not\vdash k$, $\mathtt{new}\ \mathcal{E}.\Phi \not\vdash \mathsf{pk}(k)$, $\mathtt{new}\ \mathcal{E}.\Phi \not\vdash \mathsf{vk}(k)$

Corollary 5: Let $\mathcal{E}$ be a set of names. Let Φ such that $\mathtt{new}\ \mathcal{E}.\Phi$ is a well-tagged frame in normal form and let $\mathcal{E}$ be a set of names. Let ρ be a renaming such that $\mathrm{dom}(\rho) \cup \mathrm{img}(\rho) \subseteq \mathcal{E}$ and $\mathrm{dom}(\rho) \cap fn(\Phi) = \emptyset$. If for all $k \in \mathrm{img}(\rho)$, $\mathtt{new}\ \mathcal{E}.\Phi \not\vdash k$, $\mathtt{new}\ \mathcal{E}.\Phi \not\vdash \mathsf{pk}(k)$ and $\mathtt{new}\ \mathcal{E}.\Phi \not\vdash \mathsf{vk}(k)$, then we have $\mathtt{new}\ \mathcal{E}.\Phi \sim \mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi)$.

Proof: The proof directly follows Lemmas 3 and 13. Indeed, $M\Phi\downarrow = N\Phi\downarrow$ is equivalent to $\delta_i^\rho(M\Phi\downarrow) = \delta_i^\rho(N\Phi\downarrow)$ (thanks to Lemma 3), which is equivalent to $M\delta^\rho(\Phi)\downarrow = N\delta^\rho(\Phi)\downarrow$ (thanks to Lemma 13). ■

In the following Lemma, we will consider processes without replication. Hence, we also assume that all nonces are already created, i.e. no $\mathtt{new}\ k$. Intuitively, all the nonces will be in $\mathcal{E}$, in the extended process $(\mathcal{E}; \mathcal{P}; \Phi)$.

Definition 12: Let P a colored process and let's denote $i = \mathsf{col}(P)$. Let α be a ground substitution in normal form such that $\mathrm{dom}(\alpha) \subseteq fv(P)$. We will says that (P, α) is an original well-tagged process if

- either $P = [Q]_i$;
- or $P = \mathtt{out}(u, [v]_i).[Q]_i$, $\alpha \vDash \mathsf{test}_i([v]_i)$;
- or $P = \mathtt{if}\ [u]_i = [v]_i\ \mathtt{then}\ [Q_1]_i\ \mathtt{else}\ [Q_2]_i$ with $\alpha \vDash \mathsf{test}_i([u]_i) \wedge \mathsf{test}_i([v]_i)$
- else $P = \mathtt{if}\ \mathsf{test}_i([u]_i)\ \mathtt{then}\ P'\ \mathtt{else}\ 0$

where (P', α) is an original well-tagged process, Q, Q_1, Q_2 are processes built on $\Sigma_i \cup \Sigma_0$, and u, v are some terms.

For a colored multi-set of processes $\mathcal{P}$, we say that $(\mathcal{P}, \alpha)$ is an original well-tagged multi-set of processes if for all $P \in \mathcal{P}$, (P, α) is an original well-tagged process.

Lemma 14: Let $S = (\mathcal{E}_S; \mathcal{P}_S; \Phi_S)$, $S' = (\mathcal{E}'_S; \mathcal{P}'_S; \Phi'_S)$ and $D = (\mathcal{E}_D; \mathcal{P}_D; \Phi_D)$ be three extended processes. Assume that $S \xrightarrow{\ell} S'$ and there exists an original well-tagged process $(\mathcal{P}_0, \alpha)$ and a renaming ρ, such that

- $\mathrm{dom}(\rho) \cup \mathrm{img}(\rho) \subseteq \mathcal{E}_S$, $\mathrm{dom}(\rho) \cap fn(\mathcal{P}_S, \Phi_S) = \emptyset$;
- $\mathcal{E}_S = \mathcal{E}_D$, $\Phi_D\downarrow = \delta^\rho(\Phi_S\downarrow)$;
- $\mathcal{P}_S = \mathcal{P}_0\alpha$ and $\mathcal{P}_D\downarrow = \delta^\rho(\mathcal{P}_0)\delta^\rho(\alpha\downarrow)\downarrow$.
- for all trace (tr, Φ) of S, for all $k \in \mathrm{img}(\rho)$, $\Phi \not\vdash k$, $\Phi \not\vdash \mathsf{pk}(k)$ and $\Phi \not\vdash \mathsf{vk}(k)$

We have that there exists an intermediate process $D' = (\mathcal{E}'_D; \mathcal{P}'_D; \Phi'_D)$, an original well tagged process $(\mathcal{P}'_0,\ \alpha')$ such that $D \xrightarrow{\ell} D'$, $\mathcal{E}'_S = \mathcal{E}_S = \mathcal{E}'_D$, $\Phi'_D\downarrow = \delta^\rho(\Phi'_S\downarrow)$, $\mathcal{P}'_S\downarrow = \mathcal{P}'_0\alpha'\downarrow$ and $\mathcal{P}'_D\downarrow = \delta^\rho(\mathcal{P}'_0)\delta^\rho(\alpha'\downarrow)\downarrow$.

Proof: We show this result by case-by-case analysis on the rule:

Case of the rule THEN*:* In this case, there exists ϕ formula and $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}$ processes such that $\mathcal{P}_S = \{\mathtt{if}\ \phi\ \mathtt{then}\ \mathcal{Q}_1\ \mathtt{else}\ \mathcal{Q}_2\} \uplus \mathcal{Q}$, $\mathcal{P}'_S = \mathcal{Q}_1 \uplus \mathcal{Q}$, $\mathcal{E}_S = \mathcal{E}'_S$ and $\Phi_S = \Phi'_S$ with $i = \mathsf{col}(\mathcal{Q}_1) = \mathsf{col}(\mathcal{Q}_2)$ and ϕ a conjunction of equation $u = v$. Furthermore, we have that for all equation $u = v$ of ϕ, $u =_{\mathsf{E}} v$.

Since $\mathcal{P}_S = \mathcal{P}_0\alpha$, then there exists $\mathcal{Q}_1^0$, $\mathcal{Q}_2^0$ and $\mathcal{Q}^0$ processes such that $\mathcal{Q}_1^0\alpha = \mathcal{Q}_1$, $\mathcal{Q}_2^0\alpha = \mathcal{Q}_2$ and $\mathcal{Q}^0\alpha = \mathcal{Q}$. Furthermore, either (a) there exists u such that ϕ is the formula $\mathsf{test}_i([u]_i)\alpha$, or (b) there exists u_1, u_2 such that ϕ is the formula $[u_1]_i\alpha = [u_2]_i\alpha$ and $\alpha \vDash \mathsf{test}_i([u_1]_i) \wedge \mathsf{test}_i([u_2]_i)$.

But we also have $\mathcal{P}_D\downarrow = \delta^\rho(\mathcal{P}_0)\delta^\rho(\alpha\downarrow)\downarrow$, which means that there exists ϕ' formula and $\mathcal{Q}'_1$, $\mathcal{Q}'_2$, and $\mathcal{P}'$ processes such that $\mathcal{P}_D = \{\mathtt{if}\ \phi'\ \mathtt{then}\ \mathcal{Q}'_1\ \mathtt{else}\ \mathcal{Q}'_2)\} \uplus \mathcal{Q}'$ with $\mathcal{Q}'_1\downarrow = \delta^\rho(\mathcal{Q}_1^0)\delta^\rho(\alpha\downarrow)\downarrow$, $\mathcal{Q}'_2\downarrow = \delta^\rho(\mathcal{Q}_2^0)\delta^\rho(\alpha\downarrow)\downarrow$, $\mathcal{Q}'\downarrow = \delta^\rho(\mathcal{Q}^0)\delta^\rho(\alpha\downarrow)\downarrow$ and $\Phi_D\downarrow = \delta^\rho(\Phi_S\downarrow)$. Furthermore, in case (a) $\phi'\downarrow$ is the formula $\mathsf{test}_i(\delta_i^\rho([u]_i))\delta_i^\rho(\alpha\downarrow)\downarrow$; and in case (b) ϕ' is the formula $u' = v'$ where $u\downarrow = \delta_i^\rho([u_1]_i)\delta^\rho(\alpha\downarrow)\downarrow$ and $v'\downarrow = \delta_i^\rho([u_2]_i)\delta^\rho(\alpha\downarrow)\downarrow$.

In case (a), since for all equation in $u = v$ in ϕ, $u =_{\mathsf{E}} v$, then $u\downarrow = v\downarrow$. Thus it is equivalent to $(\alpha\downarrow) \vDash \mathsf{test}_i([u]_i)$. But by Lemma 8, we know that this is equivalent to $\delta_i^\rho(\alpha\downarrow) \vDash \mathsf{test}_i(\delta_i^\rho([u]_i))$. Thus we have that for all equation $u = v$ of $\mathsf{test}_i(\delta_i^\rho([u]_i))$, $u\delta_i^\rho(\alpha\downarrow)\downarrow = v\delta_i^\rho(\alpha\downarrow)\downarrow$. Since $\phi'\downarrow$ is the formula $\mathsf{test}_i(\delta_i^\rho([u]_i))\delta_i^\rho(\alpha\downarrow)\downarrow$, we can conclude that for all equation $u = v$ of ϕ', $u\downarrow = v\downarrow$ and so $u =_{\mathsf{E}} v$.

In case (b), we know that $\alpha\downarrow \vDash \mathsf{test}_i([u_1]_i) \wedge \mathsf{test}_i([u_2]_i)$ and $[u_1]_i\alpha\downarrow = [u_2]_i\alpha\downarrow$. Thus by Corollary 2, we can deduce that $\delta_i^\rho([u_1]_i)\delta_i^\rho(\alpha\downarrow)\downarrow = \delta_i^\rho([u_2]_i)\delta_i^\rho(\alpha\downarrow)\downarrow$ which means that $u'\downarrow = v'\downarrow$ and so $u' =_{\mathsf{E}} v'$.

ϕ' being satisfied allows us to deduce that $D \xrightarrow{\ell} (\mathcal{E}_D; \mathcal{Q}'_1 \uplus \mathcal{Q}'; \Phi_D)$. But we know that on one hand $\mathcal{Q}_1^0\alpha = \mathcal{Q}_1$ and $\mathcal{Q}^0\alpha = \mathcal{Q}$, and on the other hand, $\mathcal{Q}'_1\downarrow = \delta^\rho(\mathcal{Q}_1^0)\delta^\rho(\alpha\downarrow)\downarrow$,

$\mathcal{Q}'\downarrow = \delta^\rho(\mathcal{Q}^0)\delta^\rho(\alpha\downarrow)\downarrow$ and $\Phi_D\downarrow = \delta^\rho(\Phi_S\downarrow)$. Thus with $\alpha' = \alpha$ and $\mathcal{P}'_0 = \mathcal{Q}^0_1 \uplus \mathcal{Q}^0$, the result holds.

Case of the rule ELSE*:* This case similar to the rule THEN.

Case of the rule COMM*:* In this case, there exists p, u, x terms, and $\mathcal{Q}_1$, $\mathcal{Q}_2$, $\mathcal{Q}$ processes such that $\mathcal{P}_S = \{\mathtt{out}(p,u).\mathcal{Q}_1; \mathtt{in}(p,x).\mathcal{Q}_2\} \uplus \mathcal{Q}$, $\mathcal{P}'_S = \mathcal{Q}_1 \uplus \mathcal{Q}_2\{x \mapsto u\} \uplus \mathcal{Q}$, $\mathcal{E}_S = \mathcal{E}'_S$ and $\Phi_S = \Phi'_S$. Assume that $\mathsf{col}(\mathtt{out}(p,u).\mathcal{Q}_1) = i$ and $\mathsf{col}(\mathtt{in}(p,x).\mathcal{Q}_2) = j$. We distinguish two cases:

Case $p \in \mathcal{E}_s$*:* Because processes of different color do not share private channel, it is necessarily the case that $i = j$,

Since $\mathcal{P}_S = \mathcal{P}_0\alpha$, then there exists $\mathcal{Q}^0_1$, $\mathcal{Q}^0_2$ and $\mathcal{Q}^0$ processes, v term such that $\mathcal{Q}^0_1\alpha = \mathcal{Q}_1$, $\mathcal{Q}^0_2\alpha = \mathcal{Q}_2$, $\mathcal{Q}^0\alpha = \mathcal{Q}$ and $u = [v]_i\alpha$. Furthermore, since $(\mathcal{P}_0, \alpha)$ is an original well tagged process, we have that $\alpha \vDash \mathsf{test}_i([v]_i)$.

But we also have that $\mathcal{P}_D\downarrow = \delta^\rho(\mathcal{P}_0)\delta^\rho(\alpha\downarrow)\downarrow$, which means that there exists p'', p', u', x' terms and $\mathcal{Q}'_1$, $\mathcal{Q}'_2$, $\mathcal{Q}'$ processes such that $\mathcal{P}_D = \{\mathtt{out}(p',u').\mathcal{Q}'_1; \mathtt{in}(p'',x).\mathcal{Q}'_2\} \uplus \mathcal{Q}'$, $\mathcal{E}_D = \mathcal{E}_S$ with $\mathcal{Q}'_1\downarrow = \delta^\rho(\mathcal{Q}^0_1)\delta^\rho(\alpha\downarrow)\downarrow$, $\mathcal{Q}'_2\downarrow = \delta^\rho(\mathcal{Q}^0_2)\delta^\rho(\alpha\downarrow)\downarrow$, $\mathcal{Q}'\downarrow = \delta^\rho(\mathcal{Q}^0)\delta^\rho(\alpha\downarrow)\downarrow$, $\Phi_D\downarrow = \delta^\rho(\Phi_S\downarrow)$, $p'' = \delta^\rho_i(p) = p'$ and $u'\downarrow = \delta^\rho_i([v]_i)\delta^\rho_i(\alpha\downarrow)\downarrow$.

Therefore, we have that $(\mathcal{E}_D; \mathcal{P}_D; \Phi_D) \xrightarrow{\ell} (\mathcal{E}_D; \mathcal{Q}'_1 \uplus \mathcal{Q}'_2\{x \mapsto u'\} \uplus \mathcal{P}'; \Phi_D)$.

Let's denote $\alpha' = \alpha \cup \{x \mapsto u\}$. Since $\mathcal{Q}_2 = \mathcal{Q}^0_2\alpha$, we have that $\mathcal{Q}_2\{x \mapsto u\} = \mathcal{Q}^0_2\alpha'$. Furthermore, $\mathcal{Q}'_2\{x \mapsto u'\}\downarrow = (\mathcal{Q}'_2\downarrow)\{x \mapsto (u'\downarrow)\}\downarrow$. But thanks to Lemma 7 and $\alpha \vDash \mathsf{test}_i([v]_i)$, $\delta^\rho_i([v]_i)\delta^\rho_i(\alpha\downarrow)\downarrow = \delta^\rho_i([v]_i(\alpha\downarrow))\downarrow = \delta^\rho_i([v]_i\alpha\downarrow) = \delta^\rho_i(u\downarrow)$. Thus, we deduce that $\mathcal{Q}'_2\{x \mapsto u'\}\downarrow = \delta^\rho(\mathcal{Q}^0_2)\delta^\rho(\alpha\downarrow)\{x \mapsto \delta^\rho_i(u\downarrow)\}\downarrow = \delta^\rho(\mathcal{Q}^0_2)\delta^\rho(\alpha'\downarrow)\downarrow$.

Let $\mathcal{P}'_0 = \mathcal{Q}^0_1 \uplus \mathcal{Q}^0_2 \uplus \mathcal{Q}^0$. Since x doesn't appears in $\mathcal{Q}^0_1$ and $\mathcal{Q}^0$, we conclude that $\mathcal{P}'_0\alpha' = \mathcal{P}'_S$ and $\delta^\rho(\mathcal{P}'_0)\delta^\rho(\alpha'\downarrow)\downarrow = \mathcal{P}'_D\downarrow$. Hence the result holds.

Case $p \notin \mathcal{E}_s$*:* First of all, note that $\delta^\rho_a(u\downarrow) = \delta^\rho_b(u\downarrow)$. Indeed by hypothesis, we know that for all trace (tr, Φ) of S, $\Phi \not\vdash k$, $\Phi \not\vdash \mathsf{pk}(k)$ and $\Phi \not\vdash \mathsf{vk}(k)$, for all $k \in \mathrm{img}(\rho)$. But $S \xrightarrow{\mathtt{new}\ w_n.\mathtt{out}(p,w_n)} (\mathcal{E}_S; \{\mathcal{Q}_1; \mathtt{in}(p,x).\mathcal{Q}_2\}; \Phi \cup \{w_n \rhd u\})$. Thus since $w_n\Phi\downarrow = u\downarrow$, then by Lemma 13, $\delta^\rho_a(u\downarrow) = \delta^\rho_b(u\downarrow)$.

Since $\mathcal{P}_S = \mathcal{P}_0\alpha$, then there exists $\mathcal{Q}^0_1$, $\mathcal{Q}^0_2$ and $\mathcal{Q}^0$ processes, v term such that $\mathcal{Q}^0_1\alpha = \mathcal{Q}_1$, $\mathcal{Q}^0_2\alpha = \mathcal{Q}_2$, $\mathcal{Q}^0\alpha = \mathcal{Q}$ and $u = [v]_i\alpha$. Furthermore, since $(\mathcal{P}_0, \alpha)$ is an original well tagged process, we have that $\alpha \vDash \mathsf{test}_i([v]_i)$.

But we also have that $\mathcal{P}_D\downarrow = \delta^\rho(\mathcal{P}_0)\delta^\rho(\alpha\downarrow)\downarrow$, which means that there exists p'', p', u', x' terms and $\mathcal{Q}'_1$, $\mathcal{Q}'_2$, $\mathcal{Q}'$ processes such that $\mathcal{P}_D = \{\mathtt{out}(p',u').\mathcal{Q}'_1; \mathtt{in}(p'',x).\mathcal{Q}'_2\} \uplus \mathcal{Q}'$, $\mathcal{E}_D = \mathcal{E}_S$ with $\mathcal{Q}'_1\downarrow = \delta^\rho(\mathcal{Q}^0_1)\delta^\rho(\alpha\downarrow)\downarrow$, $\mathcal{Q}'_2\downarrow = \delta^\rho(\mathcal{Q}^0_2)\delta^\rho(\alpha\downarrow)\downarrow$, $\mathcal{Q}'\downarrow = \delta^\rho(\mathcal{Q}^0)\delta^\rho(\alpha\downarrow)\downarrow$, $\Phi_D\downarrow = \delta^\rho(\Phi_S\downarrow)$, $p'' = \delta^\rho_j(p)$, $p' = \delta^\rho_i(p)$ and $u'\downarrow = \delta^\rho_i([v]_i)\delta^\rho_i(\alpha\downarrow)\downarrow$.

We assumed that all names in $\mathrm{dom}(\rho) \cup \mathrm{img}(\rho)$ are of base type. Thus, we can deduce that $\delta_i(p) = \delta_j(p) = p$ and so $p'' = p'$. Therefore, we have that $(\mathcal{E}_D; \mathcal{P}_D; \Phi_D) \xrightarrow{\ell} (\mathcal{E}_D; \mathcal{Q}'_1 \uplus \mathcal{Q}'_2\{x \mapsto u'\} \uplus \mathcal{P}'; \Phi_D)$.

Let's denote $\alpha' = \alpha \cup \{x \mapsto u\}$. Since $\mathcal{Q}_2 = \mathcal{Q}^0_2\alpha$, we have that $\mathcal{Q}_2\{x \mapsto u\} = \mathcal{Q}^0_2\alpha'$. Furthermore, $\mathcal{Q}'_2\{x \mapsto u'\}\downarrow = (\mathcal{Q}'_2\downarrow)\{x \mapsto (u'\downarrow)\}\downarrow$. But thanks to Lemma 7 and $\alpha \vDash \mathsf{test}_i([v]_i)$, $\delta^\rho_i([v]_i)\delta^\rho_i(\alpha\downarrow)\downarrow = \delta^\rho_i([v]_i(\alpha\downarrow))\downarrow = \delta^\rho_i([v]_i\alpha\downarrow) = \delta^\rho_i(u\downarrow)$. Thus, we deduce that $\mathcal{Q}'_2\{x \mapsto u'\}\downarrow = \delta^\rho(\mathcal{Q}^0_2)\delta^\rho(\alpha\downarrow)\{x \mapsto \delta^\rho_i(u\downarrow)\}\downarrow = \delta^\rho(\mathcal{Q}^0_2)\delta^\rho(\alpha'\downarrow)\downarrow$.

Let $\mathcal{P}'_0 = \mathcal{Q}^0_1 \uplus \mathcal{Q}^0_2 \uplus \mathcal{Q}^0$. Since x doesn't appears in $\mathcal{Q}^0_1$ and $\mathcal{Q}^0$, we conclude that $\mathcal{P}'_0\alpha' = \mathcal{P}'_S$ and $\delta^\rho(\mathcal{P}'_0)\delta^\rho(\alpha'\downarrow)\downarrow = \mathcal{P}'_D\downarrow$. Hence the result holds.

Case of the rule IN*:* In this case, there exists p, x, u, M terms, and $\mathcal{Q}$, $\mathcal{P}$ processes such that $p \notin \mathcal{E}_S$, $M\Phi_S\downarrow = u$, $fv(M) \subseteq \mathrm{dom}(\Phi_S)$ and $fn(M) \cap \mathcal{E}_S = \emptyset$. Furthermore, we have that $\mathcal{P}_S = \{\mathtt{in}(p,x).\mathcal{Q}_1\} \uplus \mathcal{Q}$, $\mathcal{P}'_S = \mathcal{Q}_1\{x \mapsto u\} \uplus \mathcal{Q}$, $\Phi_S = \Phi'_S$, $\mathcal{E}_S = \mathcal{E}'_S$ and $\ell = \mathtt{in}(p, M)$. Since $\mathcal{P}_S = \mathcal{P}_0\alpha$, then there exists $\mathcal{Q}^0_1$ and $\mathcal{Q}^0$ processes such that $\mathcal{Q}^0_1\alpha = \mathcal{Q}_1$ and $\mathcal{Q}^0\alpha = \mathcal{Q}$.

But we also have that $\mathcal{P}_D\downarrow = \delta^\rho(\mathcal{P}_0)\delta^\rho(\alpha\downarrow)\downarrow$, which means that there exists p' term and $\mathcal{Q}'_1$, $\mathcal{Q}'$ processes such that $\mathcal{P}_D = \{\mathtt{in}(p',x).\mathcal{Q}'_1\} \uplus \mathcal{Q}'$, $\mathcal{E}_D = \mathcal{E}_S$ with $\mathcal{Q}'_1\downarrow = \delta^\rho(\mathcal{Q}^0_1)\delta^\rho(\alpha\downarrow)\downarrow$, $\mathcal{Q}'\downarrow = \delta^\rho(\mathcal{Q}^0)\delta^\rho(\alpha\downarrow)\downarrow$, $p' = \delta^\rho_i(p)$ and $\Phi_D\downarrow = \delta^\rho(\Phi_S\downarrow)$.

p' and p are both channel type term and we assumed that all the names in $\mathrm{img}(\rho) \cup \mathrm{dom}(\rho)$ are names of base type. Thus we have that $p' = p$. Furthermore, $\mathcal{E}_D = \mathcal{E}_S$ which means that $p' \notin \mathcal{E}_D$ and $fn(M) \cap \mathcal{E}_D = \emptyset$. We also have that $\Phi_D\downarrow = \delta(\Phi_S\downarrow)$ which means that $\mathrm{dom}(\Phi_D) = \mathrm{dom}(\Phi_S)$ and so $fv(M) \subseteq \mathrm{dom}(\Phi_D)$. Thus, we can deduce that $(\mathcal{E}_D; \mathcal{P}_D; \Phi_D) \xrightarrow{\ell} (\mathcal{E}_D; \mathcal{Q}'_1\{x \mapsto u'\} \uplus \mathcal{Q}'; \Phi_D)$ where $u' = M\Phi_D$.

By hypothesis, we assumed that for all $k \in \mathrm{img}(\rho)$, $\mathtt{new}\ \mathcal{E}.\Phi'_S \not\vdash k$, $\mathtt{new}\ \mathcal{E}.\Phi'_S \not\vdash \mathsf{pk}(k)$, $\mathtt{new}\ \mathcal{E}.\Phi'_S \not\vdash \mathsf{vk}(k)$. Thus, thanks to Lemma 13, we have that $\delta^\rho_a(M(\Phi_S\downarrow)\downarrow) = \delta^\rho_b(M(\Phi_S\downarrow)\downarrow) = M\delta^\rho(\Phi_S\downarrow)\downarrow = M\Phi_D\downarrow$. But $M(\Phi_S\downarrow)\downarrow = M\Phi_S\downarrow = u\downarrow$ and $M(\Phi_D\downarrow)\downarrow = u'\downarrow$. Thus, we have that $\delta^\rho(u\downarrow) = u'\downarrow$. Since $\mathcal{Q}'_1\downarrow = \delta^\rho(\mathcal{Q}^0_1)\delta^\rho(\alpha\downarrow)\downarrow$, we deduce that $\mathcal{Q}'_1\{x \mapsto u'\}\downarrow = \delta^\rho(q^0_1)\delta^\rho(\alpha\downarrow)\{x \mapsto \delta^\rho(u\downarrow)\}\downarrow$.

At last, let $\mathcal{P}'_0 = \mathcal{Q}^0_1 \uplus \mathcal{Q}^0$ and let $\alpha' = \alpha \cup \{x \mapsto u\}$. We have that $\delta^\rho(\alpha'\downarrow) = \delta^\rho(\alpha\downarrow) \cup \{x \mapsto \delta^\rho(u\downarrow)\}$. Thus, we conclude that $\mathcal{Q}'_1\{x \mapsto u'\}\downarrow = \delta^\rho(\mathcal{Q}^0_1)\delta^\rho(\alpha'\downarrow)\downarrow$ and since x does not appear in $\mathcal{Q}^0$, $\mathcal{Q}'\downarrow = \delta^\rho(\mathcal{Q}^0)\delta^\rho(\alpha\downarrow)\downarrow = \delta^\rho(\mathcal{Q}^0)\delta^\rho(\alpha'\downarrow)\downarrow$. Hence the result holds.

Case of the rule OUT-T*:* In this case, there exists u, p terms and $\mathcal{Q}$, $\mathcal{P}$ processes such that $\mathcal{P}_S = \{\mathtt{out}(p,u).\mathcal{Q}_1\} \uplus \mathcal{Q}$, $\mathcal{P}'_S = \mathcal{Q}_1 \uplus \mathcal{Q}$, $\mathcal{E}_S = \mathcal{E}'_S$ and $\Phi'_S = \Phi_S \cup \{w_n \rhd u\}$. Furthermore, we have that $\ell = \nu w_n.\mathtt{out}(p, w_n)$, $p \notin \mathcal{E}_S$ and w_n is a variable such that $n - 1 = |\Phi_S|$.

First of all, note that $\delta^\rho_a(u\downarrow) = \delta^\rho_b(u\downarrow)$. Indeed by hypothesis, we know that, $\mathtt{new}\ \mathcal{E}_S.\Phi'_S \not\vdash k$, $\mathcal{E}_S.\Phi'_S \not\vdash \mathsf{pk}(k)$ and $\mathcal{E}_S.\Phi'_S \not\vdash \mathsf{vk}(k)$, for all $k \in \mathrm{img}(\rho)$. Hence, by Lemma 13, $\delta^\rho_a(u\downarrow) = \delta^\rho_b(u\downarrow)$.

Since $\mathcal{P}_S = \mathcal{P}_0\alpha$, then there exists $\mathcal{Q}_1^0$ and $\mathcal{Q}^0$ processes and v term such that $\mathcal{Q}_1^0\alpha = \mathcal{Q}_1$, $\mathcal{Q}^0\alpha = \mathcal{Q}$ and $u = [v]_i\alpha$ where $i = \mathsf{col}(\mathtt{out}(p,u).\mathcal{Q}_1)$. Furthermore, since $(\mathcal{P}_0, \alpha)$ is an original well tagged process, we have that $\alpha \vDash \mathsf{test}_i([v]_i)$.

But we also have that $\mathcal{P}_D\downarrow = \delta^\rho(\mathcal{P}_0)\delta^\rho(\alpha\downarrow)\downarrow$, which means that there exists p', u' terms and $\mathcal{Q}_1'$, $\mathcal{Q}'$ processes such that $\mathcal{P}'_D = \{\mathtt{out}(p',u').\mathcal{Q}_1'\} \uplus \mathcal{Q}'$, $\mathcal{E}_D = \mathcal{E}_S$ with $\mathcal{Q}_1'\downarrow = \delta^\rho(\mathcal{Q}_1^0)\delta^\rho(\alpha\downarrow)\downarrow$, $\mathcal{Q}'\downarrow = \delta^\rho(\mathcal{Q}^0)\delta^\rho(\alpha\downarrow)\downarrow$, $\Phi_D\downarrow = \delta^\rho(\Phi_S\downarrow)$, $p' = \delta_i^\rho(p)$ and $u'\downarrow = \delta_i^\rho([v]_i)\delta_i^\rho(\alpha\downarrow)\downarrow$.

p' and p are both channel type term and we assumed that $\mathrm{dom}(\rho)$ and $\mathrm{img}(\rho)$ only contains name of base type. Thus, we have that $p' = p$. Furthermore, $\mathcal{E}_D = \mathcal{E}_S$ which means that $p' \notin \mathcal{E}_D$. We also have that $\Phi_D\downarrow = \delta(\Phi_S\downarrow)$ which means that $|\Phi_D| = |\Phi_S| = n-1$. Thus, we can deduce that $(\mathcal{E}_D; \mathcal{P}_D; \Phi_D) \xrightarrow{\ell} (\mathcal{E}_D; \mathcal{Q}_1' \uplus \mathcal{Q}'; \Phi_D \cup \{w_n \rhd u'\})$.

Thanks to Lemma 7 and $\alpha \vDash \mathsf{test}_i([v]_i)$, $\delta_i^\rho([v]_i)\delta_i^\rho(\alpha\downarrow)\downarrow = \delta_i^\rho([v]_i(\alpha\downarrow))\downarrow = \delta_i^\rho([v]_i\alpha\downarrow) = \delta_i^\rho(u\downarrow)$. But $u'\downarrow = \delta_i^\rho([v]_i)\delta_i^\rho(\alpha\downarrow)\downarrow$, which means that $u'\downarrow = \delta_i^\rho(u\downarrow)$. Since we already proved that $\delta_a^\rho(u\downarrow) = \delta_b^\rho(u\downarrow)$, we can conclude that $\Phi'_D\downarrow = \delta^\rho(\Phi'_S\downarrow)$. Hence the result holds.

Case of the rule OUT-CH*:* Obvious since $\mathrm{dom}(\rho)$ and $\mathrm{img}(\rho)$ only contains name of base type

Case of the rule OPEN-CH*:* Obvious since $\mathrm{dom}(\rho)$ and $\mathrm{img}(\rho)$ only contains name of base type

Case of the rule PAR*:* In this case, there exists $\mathcal{Q}_1$, $\mathcal{Q}_2$, $\mathcal{Q}$ processes such that $\mathcal{P}_S = \{\mathcal{Q}_1 \mid \mathcal{Q}_2\} \uplus \mathcal{Q}$, $\mathcal{P}'_S = \mathcal{Q}_1 \uplus \mathcal{Q}_2 \uplus \mathcal{Q}$, $\mathcal{E}_S = \mathcal{E}'_S$ and $\Phi_S = \Phi'_S$. Assume that $\mathsf{col}(\mathcal{Q}_1) = \mathsf{col}(\mathcal{Q}_2) = i$.

Since $\mathcal{P}_S = \mathcal{P}_0\alpha$, then there exists $\mathcal{Q}_1^0$, $\mathcal{Q}_2^0$ and $\mathcal{Q}^0$ processes such that $\mathcal{Q}_1^0\alpha = \mathcal{Q}_1$, $\mathcal{Q}_2^0\alpha = \mathcal{Q}_2$, and $\mathcal{Q}^0\alpha = \mathcal{Q}$.

But we also have that $\mathcal{P}_D\downarrow = \delta^\rho(\mathcal{P}_0)\delta^\rho(\alpha\downarrow)\downarrow$, which means that there exists $\mathcal{Q}_1'$, $\mathcal{Q}_2'$, $\mathcal{Q}'$ processes such that $\mathcal{P}'_D = \{\mathcal{Q}_1; \mathcal{Q}_2'\} \uplus \mathcal{Q}'$, $\mathcal{E}_D = \mathcal{E}_S$ with $\mathcal{Q}_1'\downarrow = \delta^\rho(\mathcal{Q}_1^0)\delta^\rho(\alpha\downarrow)\downarrow$, $\mathcal{Q}_2'\downarrow = \delta^\rho(\mathcal{Q}_2^0)\delta^\rho(\alpha\downarrow)\downarrow$, $\mathcal{Q}'\downarrow = \delta^\rho(\mathcal{Q}^0)\delta^\rho(\alpha\downarrow)\downarrow$, and $\Phi_D\downarrow = \delta^\rho(\Phi_S\downarrow)$. Therefore, we have that $(\mathcal{E}_D; \mathcal{P}_D; \Phi_D) \xrightarrow{\ell} (\mathcal{E}_D; \mathcal{Q}_1' \uplus \mathcal{Q}_2' \uplus \mathcal{P}'; \Phi_D)$.

Let $\alpha' = \alpha$ and $\mathcal{P}_0' = \mathcal{Q}_1^0 \uplus \mathcal{Q}_2^0 \uplus \mathcal{Q}^0$. We obviously conclude that $\mathcal{P}_0'\alpha' = \mathcal{P}'_S$ and $\delta^\rho(\mathcal{P}_0')\delta^\rho(\alpha'\downarrow)\downarrow = \mathcal{P}'_D\downarrow$. Hence the result holds. ∎

Lemma 15: Let $S = (\mathcal{E}_S; \mathcal{P}_S; \Phi_S)$, $D = (\mathcal{E}_D; \mathcal{P}_D; \Phi_D)$ and $D' = (\mathcal{E}'_D; \mathcal{P}'_D; \Phi'_D)$ and be three extended processes. Assume that $D \xrightarrow{\ell} D'$ and there exists an original well-tagged process $(\mathcal{P}_0, \alpha)$ and a renaming ρ, such that

- $\mathrm{dom}(\rho) \cup \mathrm{img}(\rho) \subseteq \mathcal{E}_S$, $\mathrm{dom}(\rho) \cap fn(\mathcal{P}_S, \Phi_S) = \emptyset$;
- $\mathcal{E}_S = \mathcal{E}_D$, $\Phi_D\downarrow = \delta^\rho(\Phi_S\downarrow)$;
- $\mathcal{P}_S = \mathcal{P}_0\alpha$ and $\mathcal{P}_D\downarrow = \delta^\rho(\mathcal{P}_0)\delta^\rho(\alpha\downarrow)\downarrow$.
- for all trace (tr, Φ) of D, for all $k \in \mathrm{img}(\rho)$, $\Phi \nvdash k$, $\Phi \nvdash \mathsf{pk}(k)$ and $\Phi \nvdash \mathsf{vk}(k)$

We have that there exists an intermediate process $S' = (\mathcal{E}'_S; \mathcal{P}'_S; \Phi'_S)$, an original well tagged process $(\mathcal{P}_0', \alpha')$ such that $S \xrightarrow{\ell} S'$, $\mathcal{E}'_S = \mathcal{E}_D = \mathcal{E}'_D$, $\Phi'_D\downarrow = \delta^\rho(\Phi'_S\downarrow)$, $\mathcal{P}'_S\downarrow = \mathcal{P}_0'\alpha'\downarrow$ and $\mathcal{P}'_D\downarrow = \delta^\rho(\mathcal{P}_0')\delta^\rho(\alpha')\downarrow$.

Proof: The proof of this Lemma is almost identical to the proof of Lemma 14. Indeed, in the proof of Lemma 14, we used Lemma 7 to show that $\alpha \vDash \mathsf{test}_i([u]_i)$ implies that $\delta_i^\rho(\alpha) \vDash \mathsf{test}_i(\delta_i^\rho([u]_i))$. But Lemma 7 shows that those two properties are equivalent. The same goes for Corollary 2. At last, the conditions of Lemma 13 are fulfilled in both lemmas thus, it can be also used in this proof. ∎

G. Proof of Theorem 2

In this subsection, we will focus on the proof of Theorem 2. We will assume, as in Subsection F, that processes and frames are colored by a or b. Let ρ_0 be a renaming on names. We will consider the two following frames $\Phi_a = \{w_1^a \rhd u_1, \ldots, w_n^a \rhd u_n\}$ and $\Phi_b = \{w_1^b \rhd u_1', \ldots, w_n^b \rhd u_n'\}$ such that for all $j \in \{1, \ldots, n\}$, $u_j = u_j' = \mathsf{f}(k)$, for some $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}\}$ and $k \in \mathrm{img}(\rho_0)$. Furthermore assume that Φ_a (resp. Φ_b) is colored by a (resp. b).

Lemma 16: Let Φ and Φ' two frame in normal form such that $\mathrm{dom}(\Phi) = \mathrm{dom}(\Phi')$. Assume that Φ and Φ' have the same colors, i.e. for all $(w \rhd u) \in \Phi$, for all $(w' \rhd u') \in \Phi'$, $w = w'$ implies $\mathsf{col}(w \rhd u) = \mathsf{col}(w' \rhd u')$. Let $\mathcal{E}$ be a set of names and let ρ a renaming such that $\mathrm{dom}(\rho) \cup \mathrm{img}(\rho) \subseteq \mathcal{E}$, $\mathrm{dom}(\rho) \cap fn(\Phi, \Phi') = \emptyset$ and $\rho_{|\,\mathrm{dom}(\rho_0)} = \rho_0$. Let's denote $\Phi_+ = \Phi_a \uplus \Phi_b \uplus \Phi$ and $\Phi'_+ = \Phi_a \uplus \Phi_b \uplus \Phi'$.

If $\mathtt{new}\ \mathcal{E}.\Phi_+$, $\mathtt{new}\ \mathcal{E}.\Phi'_+$ are well-tagged, $\mathtt{new}\mathcal{E}.\delta^\rho(\Phi_+) \sim \mathtt{new}\mathcal{E}.\delta^\rho(\Phi'_+)$ and for all $u \in \{k, \mathsf{pk}(k), \mathsf{vk}(k) \mid k \in \mathrm{img}(\rho) \cup \mathrm{dom}(\rho)\}$, $\mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi_+) \vdash u$ or $\mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi'_+) \vdash u$ implies that $u \in \mathrm{img}(\delta^\rho(\Phi_a \uplus \Phi_b))$, then we have that : For all M such that $fn(M) \cap \mathcal{E} = \emptyset$ and $fv(M) \subseteq \mathrm{dom}(\Phi_+)$, there exists M_a and M_b such that $fn(M_a, M_b) \cap \mathcal{E} = \emptyset$, $fv(M_a, M_b) \subseteq \mathrm{dom}(\Phi_+)$ and :

1) $\delta_a^\rho(M\Phi_+\downarrow) = M_a\delta^\rho(\Phi_+)\downarrow$ and $\delta_a^\rho(M\Phi'_+\downarrow) = M_a\delta^\rho(\Phi'_+)\downarrow$
2) $\delta_b^\rho(M\Phi_+\downarrow) = M_b\delta^\rho(\Phi_+)\downarrow$ and $\delta_b^\rho(M\Phi'_+\downarrow) = M_b\delta^\rho(\Phi'_+)\downarrow$

Proof: We prove this lemma by induction on $\mathcal{M}(M)$.

Base case $\mathcal{M}(M) = (0,0)$*:* There exists no term M such that $|M| = 0$, thus the result holds.

Inductive step $\mathcal{M}(M) > (0,0)$*:* The first step of the proof will be to show that there exists $c \in \{a, b\}$, a terms M_c such that $fn(M_c) \cap \mathcal{E} = \emptyset$, $fv(M_c) \subseteq \mathrm{dom}(\Phi_+)$, $\delta_c^\rho(M\Phi_+\downarrow) = M_c\delta^\rho(\Phi_+)\downarrow$ and $\delta_c^\rho(M\Phi'_+\downarrow) = M_c\delta^\rho(\Phi'_+)\downarrow$. Then the second step will consist in showing that there exists another term M_d, with $d \in \{a, b\}$ and $c \neq d$ that verifies the wanted properties.

First step: Assume first that $|M| = 1$. In such a case, we have that either $M \in \mathcal{N}$ or there exists $w \in \mathrm{dom}(\Phi_+)$ such that $M = w$. If $M \in \mathcal{N}$, then by hypothesis, we

know that $M \notin \mathcal{E}$. Since $\mathrm{dom}(\rho) \cup \mathrm{img}(\rho) \subseteq \mathcal{E}$, we can deduce that $\delta_i^\rho(M) = M$, for all $i \in \{a, b\}$. Furthermore, $M \in \mathcal{N}$ also implies that $M\Phi_+\downarrow = M$, $M\Phi'_+\downarrow = M$, $M\delta^\rho(\Phi_+)\downarrow = M$ and $M\delta^\rho(\Phi'_+)\downarrow = M$. Thus, the result holds. Else if $w \in \mathrm{dom}(\Phi_+)$, since $\mathrm{dom}(\Phi_+) = \mathrm{dom}(\Phi'_+)$, we know that there exists u, u' such that $(w \rhd u) \in \Phi_+$ and $(w \rhd u') \in \Phi'_+$. Furthermore, we assumed that Φ and Φ' have the same colors, then so do Φ_+ and Φ'_+. Hence we have that $\mathsf{col}(w \rhd u) = \mathsf{col}(w \rhd u')$. Let's denote $i = \mathsf{col}(w \rhd u)$. By definition of $\delta^\rho(\Phi_+)$ and $\delta^\rho(\Phi'_+)$, we have that $w\delta^\rho(\Phi_+) = \delta_i^\rho(w\Phi_+)$ and $w\delta^\rho(\Phi'_+) = \delta_i^\rho(w\Phi'_+)$. Note that we assumed that Φ and Φ' are both in normal form, thus so are Φ_+ and Φ'_+. Hence, we can conclude that $M\delta^\rho(\Phi_+)\downarrow = \delta_i^\rho(M\Phi_+\downarrow)$ and $M\delta^\rho(\Phi'_+)\downarrow = \delta_i^\rho(M\Phi'_+\downarrow)$.

Assume now that $|M| > 1$. It implies that there exists $M_1, \ldots, M_n$ term and a function symbol f such that $M = \mathsf{f}(M_1, \ldots, M_n)$. We do a case analysis on f.

Case $\mathsf{f} \in \Sigma_i \cup \Sigma_{\mathsf{tag}_i}$, $i \in \{a, b\}$*:* In such a case, let $t = \mathsf{f}(M_1\Phi_+\downarrow, \ldots, M_n\Phi_+\downarrow)$ and $t' = \mathsf{f}(M_1\Phi'_+\downarrow, \ldots, M_n\Phi'_+\downarrow)$. Since $\mathsf{f} \in \Sigma_i$, we have that $\delta_i^\rho(t) = \mathsf{f}(\delta_i^\rho(M_1\Phi_+\downarrow), \ldots, \delta_i^\rho(M_n\Phi_+\downarrow))$ and $\delta_i^\rho(t') = \mathsf{f}(\delta_i^\rho(M_1\Phi'_+\downarrow), \ldots, \delta_i^\rho(M_n\Phi'_+\downarrow))$. But we have that $\mathcal{M}(M_1) < \mathcal{M}(M)$, ..., $\mathcal{M}(M_n) < \mathcal{M}(M)$. Thus we can apply or inductive hypothesis on $M_1, \ldots, M_n$ and so there exists $M_1^i, \ldots, M_n^i$ such that $fn(M_k^i) \cap \mathcal{E} = \emptyset$, $fv(M_k^i) \subseteq \mathrm{dom}(\Phi_+)$, $\delta_i^\rho(M_k\Phi_+\downarrow) = M_k^i\delta^\rho(\Phi_+)\downarrow$ and $\delta_i^\rho(M_k\Phi'_+\downarrow) = M_k^i\delta^\rho(\Phi'_+)\downarrow$, for all $k \in \{1, \ldots, n\}$. Hence, we deduce that $\delta_i^\rho(t)\downarrow = \mathsf{f}(M_1^i, \ldots, M_n^i)\delta^\rho(\Phi_+)\downarrow$ and $\delta_i^\rho(t')\downarrow = \mathsf{f}(M_1^i, \ldots, M_n^i)\delta^\rho(\Phi'_+)\downarrow$.

On the other hand, $t = \mathsf{f}(M_1\Phi_+\downarrow, \ldots, M_n\Phi_+\downarrow)$ implies that there exists C context built on $\Sigma_i \cup \Sigma_{\mathsf{tag}_i}$ and $u_1, \ldots, u_m$ terms in normal form such that $t = C[u_1, \ldots, u_m]$ with $u_1, \ldots, u_m$ factors of t. Thanks to Lemma 1, there exists a context D (possibly a hole) built on $\Sigma_i \cup \Sigma_{\mathsf{tag}_i}$ such that $t\downarrow = D[u_{j_1}, \ldots, u_{j_k}]$ with $j_1, \ldots, j_k \in \{0, \ldots, m\}$ and $u_0 = n_{min}$. But using Lemma 2, 3 and 6, we can deduce that $C[\delta_i^\rho(u_1), \ldots, \delta_i^\rho(u_m)]\downarrow = D[\delta_i^\rho(u_{j_1}), \ldots, \delta_i^\rho(u_{j_k})]$. Since C and D are both built upon $\Sigma_i \cup \Sigma_{\mathsf{tag}_i}$, we can conclude that $\delta_i^\rho(C[u_1, \ldots, u_m])\downarrow = \delta_i^\rho(D[u_{j_1}, \ldots, u_{j_k}])$ and so $\delta_i^\rho(t)\downarrow = \delta_i^\rho(t\downarrow)$.

Similarly, we have that $t' = \mathsf{f}(M_1\Phi'_+\downarrow, \ldots, M_n\Phi'_+\downarrow)$ implies that $\delta_i^\rho(t')\downarrow = \delta_i^\rho(t'\downarrow)$. Since we proved that $\delta_i^\rho(t)\downarrow = \mathsf{f}(M_1^i, \ldots, M_n^i)\delta^\rho(\Phi_+)\downarrow$, $\delta_i^\rho(t')\downarrow = \mathsf{f}(M_1^i, \ldots, M_n^i)\delta^\rho(\Phi'_+)\downarrow$, and since $t\downarrow = M\Phi_+\downarrow$, $t'\downarrow = M\Phi'_+\downarrow$, we conclude that $\mathsf{f}(M_1^i, \ldots, M_n^i)\delta^\rho(\Phi_+)\downarrow = \delta_i^\rho(M\Phi_+\downarrow)$ and $\mathsf{f}(M_1^i, \ldots, M_n^i)\delta^\rho(\Phi'_+)\downarrow = \delta_i^\rho(M\Phi'_+\downarrow)$.

Case $\mathsf{f} \in \{\mathsf{senc}, \mathsf{aenc}, \mathsf{sign}, \langle\rangle\}$*:* By definition of Σ_0, we know that $M\Phi_+\downarrow = \mathsf{f}(M_1\Phi_+\downarrow, M_2\Phi_+\downarrow)$ and $M\Phi'_+\downarrow = \mathsf{f}(M_1\Phi'_+\downarrow, M_2\Phi'_+\downarrow)$. Thanks to Lemma 6, we can deduce that for all $i \in \{a, b\}$, $\mathsf{root}(\delta_i^\rho(M_1\Phi_+\downarrow)) = \mathsf{root}(M_1\Phi_+\downarrow)$ and $\mathsf{root}(\delta_i^\rho(M_1\Phi'_+\downarrow)) = \mathsf{root}(M_1\Phi'_+\downarrow)$. But $\mathcal{M}(M_1) < \mathcal{M}(M)$, thus by our inductive hypothesis, there exists M_1^i such that $\delta_i^\rho(M_1\Phi_+\downarrow) = M_1^i\delta^\rho(\Phi_+)\downarrow$ and $\delta_i^\rho(M_1\Phi'_+\downarrow) = M_1^i\delta^\rho(\Phi'_+)\downarrow$. Hence for all $j \in \{a, b\}$, $\mathsf{root}(M_1\Phi_+\downarrow) = \mathsf{tag}_j$ is equivalent to $\mathsf{root}(M_1^i\delta^\rho(\Phi_+)\downarrow) = \mathsf{tag}_j$, which is also equivalent to $\mathsf{tag}_j(\mathsf{untag}_j(M_1^i))\delta^\rho(\Phi_+)\downarrow = M_1^i\delta^\rho(\Phi_+)\downarrow$. But by hypothesis $\mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi_+) \sim \mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi'_+)$, thus $\mathsf{tag}_j(\mathsf{untag}_j(M_1^i))\delta^\rho(\Phi_+)\downarrow = M_1^i\delta^\rho(\Phi_+)\downarrow$ is equivalent to $\mathsf{tag}_j(\mathsf{untag}_j(M_1^i))\delta^\rho(\Phi'_+)\downarrow = M_1^i\delta^\rho(\Phi'_+)\downarrow$, which is equivalent to $\mathsf{root}(M_1^i\delta^\rho(\Phi'_+)\downarrow) = \mathsf{tag}_j$. Hence, we deduce that for all $j \in \{a, b\}$, $\mathsf{root}(M_1\Phi_+\downarrow) = \mathsf{tag}_j$ is equivalent to $\mathsf{root}(M_1\Phi'_+\downarrow) = \mathsf{tag}_j$.

This equivalence and the definition of δ_a^ρ and δ_b^ρ allow us to deduce that for all $i \in \{a, b\}$, there exists $j \in \{a, b\}$ such that $\delta_i^\rho(M\Phi_+\downarrow) = \mathsf{f}(\delta_j^\rho(M_1\Phi_+\downarrow), \delta_j^\rho(M_2\Phi_+\downarrow))$ and $\delta_i^\rho(M\Phi'_+\downarrow) = \mathsf{f}(\delta_j^\rho(M_1\Phi'_+\downarrow), \delta_j^\rho(M_2\Phi'_+\downarrow))$. By our inductive hypothesis on M_1 and M_2, we deduce that there exists M_1^j and M_2^j such that $\delta_j^\rho(M_k\Phi_+\downarrow) = M_k^j\delta^\rho(\Phi_+)\downarrow$ and $\delta_j^\rho(M_k\Phi'_+\downarrow) = M_k^j\delta^\rho(\Phi'_+)\downarrow$, for $k \in \{1, 2\}$. Hence, we have that $\delta_i^\rho(M\Phi_+\downarrow) = \mathsf{f}(M_1^j, M_2^j)\delta^\rho(\Phi_+)\downarrow$ and $\delta_i^\rho(M\Phi'_+\downarrow) = \mathsf{f}(M_1^j, M_2^j)\delta^\rho(\Phi'_+)\downarrow$. So the result holds.

Case $\mathsf{f} \in \{\mathsf{h}, \mathsf{pk}, \mathsf{vk}\}$*:* This case is analogous to the previous one and can be handled similarly.

Case $\mathsf{f} = \mathsf{sdec}$*:* We have that $M = \mathsf{f}(M_1, M_2)$. Thus, we can apply our inductive hypothesis on M_1 and M_2, which means that for all $i \in \{a, b\}$, for all $k \in \{1, 2\}$, there exists $M_k^i\delta^\rho(\Phi_+)\downarrow = \delta_i^\rho(M_k\Phi_+\downarrow)$ and $M_k^i\delta^\rho(\Phi'_+)\downarrow = \delta_i^\rho(M_k\Phi'_+\downarrow)$. Let's first focus on $M\Phi_+\downarrow$. We need to distinguish several cases:

(a) If the root occurence of sdec cannot be reduced, then $M\Phi_+\downarrow = \mathsf{sdec}(M_1\Phi_+\downarrow, M_2\Phi_+\downarrow)$. Thus for all $i \in \{a, b\}$, $\delta_i^\rho(M\Phi_+\downarrow) = \mathsf{sdec}(\delta_i^\rho(M_1\Phi_+\downarrow), \delta_i^\rho(M_2\Phi_+\downarrow)) = \mathsf{sdec}(M_1^i\delta^\rho(\Phi_+)\downarrow, M_2^i\delta^\rho(\Phi_+)\downarrow)$. Thanks to Lemma 6, we know that $\delta_i^\rho(M\Phi_+\downarrow)$ is in normal form which means that $\delta_i^\rho(M\Phi_+\downarrow) = \mathsf{f}(M_1^i, M_2^i)\delta^\rho(\Phi_+)\downarrow$

(b) If the root occurence of sdec can be reduced and $\mathsf{root}(M\Phi_+\downarrow) = \mathsf{tag}_j$, for some $j \in \{a, b\}$ then there exists u_1, u_2 such that $M_1\Phi_+\downarrow = \mathsf{senc}(\mathsf{tag}_j(u_1), u_2)$ and $M_2\Phi_+\downarrow = u_2$. Hence we have that $\mathsf{sdec}(\delta_j^\rho(M_1\Phi_+\downarrow), \delta_j^\rho(M_2\Phi_+\downarrow))\downarrow = \delta_j^\rho(\mathsf{tag}_j(u_1)) = \delta_j^\rho(M\Phi_+\downarrow)$. On the other hand, we also have $\mathsf{sdec}(\delta_j^\rho(M_1\Phi_+\downarrow), \delta_j^\rho(M_2\Phi_+\downarrow))\downarrow = \mathsf{sdec}(M_1^j\delta^\rho(\Phi_+)\downarrow, M_2^j\delta^\rho(\Phi_+)\downarrow)\downarrow$. Hence, we have that $\delta_j^\rho(M\Phi_+\downarrow) = \mathsf{sdec}(M_1^j, M_2^j)\delta^\rho(\Phi_+)\downarrow$.

(c) Else, the root occurence of sdec can be reduced and for all $j \in \{a, b\}$, $\mathsf{root}(M\Phi_+\downarrow) \neq \mathsf{tag}_j$. In such a case, there exist u_1, u_2 such that $M_1\Phi_+\downarrow = \mathsf{senc}(u_1, u_2)$, $M_2\Phi_+\downarrow = u_2$. Moreover, for all $i \in \{a, b\}$, $\delta_i^\rho(M_1\Phi_+\downarrow) = \mathsf{senc}(\delta_i^\rho(u_1), \delta_i^\rho(u_2))$ and $\delta_i^\rho(M_2\Phi_+\downarrow) = \delta_i^\rho(u_2)$. Hence, we have that for all $i \in \{a, b\}$, $\mathsf{sdec}(\delta_i^\rho(M_1\Phi_+\downarrow), \delta_i^\rho(M_2\Phi_+\downarrow))\downarrow = \delta_i^\rho(M\Phi_+\downarrow)$. At last, since $\mathsf{sdec}(\delta_i^\rho(M_1\Phi_+\downarrow), \delta_i^\rho(M_2\Phi_+\downarrow))\downarrow = \mathsf{sdec}(M_1^i, M_2^i)\delta^\rho(\Phi_+)\downarrow$, we conclude that $\delta_i^\rho(M\Phi_+\downarrow) = \mathsf{sdec}(M_1^i, M_2^i)\delta^\rho(\Phi_+)\downarrow$.

We could do the same case analysis for $M\Phi'_+\downarrow$ and we would obtain similar results. Since in two cases (a) and (c),

the result holds for all $i \in \{a, b\}$, it only remains to show that for all $j \in \{a, b\}$, the root occurence of sdec is reduced in $M\Phi_+\downarrow$ and $\mathsf{root}(M\Phi_+\downarrow) = \mathsf{tag}_j$, if and only if, the root occurence of sdec is reduced in $M\Phi'_+\downarrow$ and $\mathsf{root}(M\Phi'_+\downarrow) = \mathsf{tag}_j$.

We already showed that the root occurence of sdec is reduced in $M\Phi_+\downarrow$ and $\mathsf{root}(M\Phi_+\downarrow) = \mathsf{tag}_j$ imply that $\delta^\rho_j(M\Phi_+\downarrow) = \mathsf{sdec}(M^j_1, M^j_2)\delta^\rho(\Phi_+)\downarrow$. Thus we have $\mathsf{tag}_j(\mathsf{untag}_j(\mathsf{sdec}(M^j_1, M^j_2)))\delta^\rho(\Phi_+)\downarrow = \mathsf{sdec}(M^j_1, M^j_2)\delta^\rho(\Phi_+)\downarrow$. But by hypothesis, $\mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi_+) \sim \mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi'_+)$. Hence $\mathsf{tag}_j(\mathsf{untag}_j(\mathsf{sdec}(M^j_1, M^j_2)))\delta^\rho(\Phi'_+)\downarrow = \mathsf{sdec}(M^j_1, M^j_2)\delta^\rho(\Phi'_+)\downarrow$. Thus there exists v_1, v_2 such that $M^j_1\delta^\rho(\Phi'_+)\downarrow = \mathsf{senc}(\mathsf{tag}_j(v_1), v_2)$ and $M^j_2\delta^\rho(\Phi'_+)\downarrow = v_2$, which means that $\delta^\rho_j(M_1\Phi'_+\downarrow) = \mathsf{senc}(\mathsf{tag}_j(v_1), v_2)$ and $\delta^\rho_j(M_2\Phi'_+\downarrow) = v_2$. Thanks to Lemma 3 and 6, we deduce that there exists v'_1 and v'_2 such that $M_1\Phi'_+\downarrow = \mathsf{senc}(\mathsf{tag}_j(v'_1), v'_2)$, $M_2\Phi'_+\downarrow = v'_2$, $v_2 = \delta^\rho_j(v'_2)$ and $v_1 = \delta^\rho_j(v'_1)$. We can conclude that the root occurence of sdec is reduced in $M\Phi'_+\downarrow$ and $\mathsf{root}(M\Phi'_+\downarrow) = \mathsf{tag}_j$. The other implication is symmetrical to this one. Hence our result holds.

Case $\mathsf{f} \in \{\mathsf{adec}, \mathsf{check}\}$*:* This case is similar to the case $\mathsf{f} = \mathsf{sdec}$.

Case $\mathsf{f} \in \mathsf{proj}_k$, $k \in \{1, 2\}$*:* In such a case, we have $M = \mathsf{f}(M_1)$. Thus we can apply our inductive hypothesis on M_1 which means that for all $i \in \{a, b\}$, there exists $M^i_1\delta^\rho(\Phi_+)\downarrow = \delta^\rho_i(M_1\Phi_+\downarrow)$ and $M^i_1\delta^\rho(\Phi'_+)\downarrow = \delta^\rho_i(M_1\Phi'_+\downarrow)$. Let's first focus on $M\Phi_+\downarrow$. We need to distinguish two cases:

(a) If the root occurence of f cannot be reduced, then $M\Phi_+\downarrow = \mathsf{f}(M_1\Phi_+\downarrow)$. Thus for all $i \in \{a, b\}$, $\delta^\rho_i(M\Phi_+\downarrow) = \mathsf{f}(\delta^\rho_i(M_1\Phi_+\downarrow)) = \mathsf{f}(M^i_1\delta^\rho(\Phi_+)\downarrow)$. Thanks to Lemma 6, we know that $\delta^\rho_i(M\Phi_+\downarrow)$ is in normal form which means that $\delta^\rho_i(M\Phi_+\downarrow) = \mathsf{sdec}(M^i_1)\delta^\rho(\Phi_+)\downarrow$.

(b) Else, the root occurence of f can be reduced. In such a case there exists u_1, u_2 such that $M_1\Phi_+\downarrow = \langle u_1, u_2\rangle$. Hence for all $i \in \{a, b\}$, $\delta^\rho_i(M_1\Phi_+\downarrow) = \langle\delta^\rho_i(u_1), \delta^\rho_i(u_2)\rangle$ and so $\mathsf{f}(\delta^\rho_i(M_1\Phi_+\downarrow))\downarrow = \delta^\rho_i(u_k)$. Thus, we can conclude that $\mathsf{f}(M^i_1)\delta^\rho(\Phi_+)\downarrow = \delta^\rho_i(M\Phi_+\downarrow)$

We could do the same case analysis for $M\Phi'_+\downarrow$ and we would obtain similar results. Since the result holds in both cases, we can conclude.

Second step: Thanks to the first step, we showed that there exists $c \in \{a, b\}$, and a terms M_c such that $fn(M_c) \cap \mathcal{E} = \emptyset$, $fv(M_c) \subseteq \mathrm{dom}(\Phi_+)$, $\delta^\rho_c(M\Phi_+\downarrow) = M_c\delta^\rho(\Phi_+)\downarrow$ and $\delta^\rho_c(M\Phi'_+\downarrow) = M_c\delta^\rho(\Phi'_+)\downarrow$. Let $d \in \{a, b\}$ such that $d \neq c$.

By Lemma 12, we know that there exists two contexts C, C' (possibly holes) built on $\{\langle\ \rangle\}$, and $u_1, \ldots, u_m, v_1, \ldots, v_n$ such that $M\Phi_+\downarrow = C[u_1, \ldots, u_m]$ and $M\Phi'_+\downarrow = C'[v_1, \ldots, v_n]$. Hence, we have that $\delta^\rho_c(M\Phi_+\downarrow) = C[\delta^\rho_c(u_1), \ldots, \delta^\rho_c(u_m)]$ and $\delta^\rho_c(M\Phi'_+\downarrow) = C'[\delta^\rho_c(v_1), \ldots, \delta^\rho_c(v_n)]$. Our hypothesis implies that $M_c\delta^\rho(\Phi_+)\downarrow = C[\delta^\rho_c(u_1), \ldots, \delta^\rho_c(u_m)]$ and $M_c\delta^\rho(\Phi'_+)\downarrow = C'[\delta^\rho_c(v_1), \ldots, \delta^\rho_c(v_n)]$. But $\mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi_+) \sim \mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi'_+)$. Thus, since C and C' are both built upon $\{\langle\ \rangle\}$, we deduce that $C = C'$ and $n = m$. Note that it also implies that there exists $D_1, \ldots, D_n$ context built on $\{\mathsf{proj}_1, \mathsf{proj}_2\}$ such that for all $k \in \{1, \ldots, n\}$, $D_k(M_c)\delta^\rho(\Phi_+)\downarrow = \delta^\rho_c(u_k)$ and $D_k(M_c)\delta^\rho(\Phi'_+)\downarrow = \delta^\rho_c(v_k)$.

Lemma 12 also tells us that for all $k \in \{1, \ldots, n\}$,

- either $u_k \in \mathsf{Flawed}(M\Phi_+\downarrow)$;
- or $u_k \in Fct_{\Sigma_0}(M\Phi_+\downarrow)$ and $\delta^\rho_a(u_k) = \delta^\rho_b(u_k)$,
- or $u_k = \mathsf{f}(n)$ for some $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}\}$ and $n \in \mathcal{N}$,
- or $u_k \in \mathcal{N}$.

Thus, we build M_d such that $M_d = C[N_1, \ldots, N_n]$ where, for all $k \in \{1, \ldots, n\}$,

(a) if $\delta^\rho_a(u_k) = \delta^\rho_b(u_k)$ then $N_k = D_k(M_c)$

(b) else if $u_k \in \mathsf{Flawed}(M\Phi_+\downarrow)$, then by Lemma 11, there exists f and $M_1, \ldots, M_\ell$ such that for all $i \in \{1 \ldots, \ell\}$, $\mathcal{M}(M_i) < \mathcal{M}(M)$ and $u_k = \mathsf{f}(M_1\Phi_+\downarrow, \ldots, M_\ell\Phi_+\downarrow)$. But by our inductive hypothesis on $M_1, \ldots, M_\ell$, we have that there exists $M^d_1, \ldots, M^d_\ell$ such that for all $i \in \{1, \ldots, \ell\}$, $M^d_i\delta^\rho(\Phi_+)\downarrow = \delta^\rho_d(M_i\Phi_+\downarrow)$ and $M^d_i\delta^\rho(\Phi'_+)\downarrow = \delta^\rho_d(M_i\Phi'_+\downarrow)$. Hence we have $\mathsf{f}(M^d_1, \ldots, M^d_\ell)\delta^\rho(\Phi_+)\downarrow = \delta^\rho_d(\mathsf{f}(M_1\Phi_+\downarrow, \ldots, M_\ell\Phi_+\downarrow))$ and $\mathsf{f}(M^d_1, \ldots, M^d_\ell)\delta^\rho(\Phi'_+)\downarrow = \delta^\rho_d(\mathsf{f}(M_1\Phi'_+\downarrow, \ldots, M_\ell\Phi'_+\downarrow))$.

We define $N_k = \mathsf{f}(M^d_1, \ldots, M^d_\ell)$. We know that $N_k\delta^\rho(\Phi_+)\downarrow = \delta^\rho_d(u_k)$, thus it remains to show that $N_k\delta^\rho(\Phi'_+)\downarrow = \delta^\rho_d(v_k)$. Our inductive hypothesis on $M_1, \ldots, M_\ell$ allows us to show that there exists $M^c_1, \ldots, M^c_\ell$ such that $\mathsf{f}(M^c_1, \ldots, M^c_\ell)\delta^\rho(\Phi_+)\downarrow = \delta^\rho_c(u_k)$. But $\delta^\rho_c(u_k) = D_k(M_c)\delta^\rho(\Phi_+)\downarrow$. Since $\delta^\rho(\Phi_+) \sim \delta^\rho(\Phi'_+)$, we deduce that $D_k(M_c)\delta^\rho(\Phi'_+)\downarrow = \mathsf{f}(M^c_1, \ldots, M^c_\ell)\delta^\rho(\Phi'_+)\downarrow$. Hence we have that $\delta^\rho_c(v_k) = \delta^\rho_c(\mathsf{f}(M_1\Phi'_+\downarrow, \ldots, M_\ell\Phi'_+\downarrow))$. By applying Lemma 3, we obtain that $\mathsf{f}(M_1\Phi'_+\downarrow, \ldots, M_\ell\Phi'_+\downarrow) = v_k$ and so $N_k\delta^\rho(\Phi'_+)\downarrow = \delta^\rho_d(v_k)$

(c) else if $u_k \in \mathsf{f}(n)$ *for some* $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}\}$ *and* $n \in \mathcal{N}$, then $D_k(M_c)\delta^\rho(\Phi_+)\downarrow = \mathsf{f}(\delta^\rho_c(n))$, i.e. $\mathsf{f}(\delta^\rho_c(n))$ is deducible. If $n \notin \mathrm{img}(\rho)$ then $\delta^\rho_a(u_k) = \delta^\rho_b(u_k)$ and so it is the same as case (a). Else by hypothesis on $\delta^\rho(\Phi_+)$, we know that $n \in \mathrm{img}(\rho)$ implies that $\mathsf{f}(\delta^\rho_c(n)) \in \delta^\rho(\Phi_a \uplus \Phi_b)$. By construction of Φ_a and Φ_b, we know that there exist $(w_d \rhd \mathsf{f}(\delta^\rho_d(n)), w_c \rhd \mathsf{f}(\delta^\rho_c(n))) \in \delta^\rho(\Phi_a \uplus \Phi_b)$ with $\mathsf{col}(w_d) = d$ and $\mathsf{col}(w_c) = c$.

We define $N_k = w_d$. It remains to show that $w_d\delta^\rho(\Phi'_+)\downarrow = \delta^\rho_d(v_k)$. But we have $w_c\delta^\rho(\Phi_+)\downarrow = D_k(M_c)\delta^\rho(\Phi_+)\downarrow$. Thanks to our hypothesis $\delta^\rho(\Phi_+) \sim \delta^\rho(\Phi'_+)$, we deduce that $w_c\delta^\rho(\Phi'_+)\downarrow = D_k(M_c)\delta^\rho(\Phi'_+)\downarrow$. By definition of Φ'_+, we have that $w_c\delta^\rho(\Phi'_+)\downarrow = \mathsf{f}(\delta^\rho_c(n))$ which means that $\delta^\rho_c(v_k) = D_k(M_c)\delta^\rho(\Phi'_+)\downarrow = \delta^\rho_c(\mathsf{f}(n))$. By applying Lemma 3, we obtain that $v_k = \mathsf{f}(n)$. At last, by definition of Φ'_+, we have that $w_d\delta^\rho(\Phi'_+)\downarrow = \delta^\rho_d(\mathsf{f}(n)) = \delta^\rho_d(v_k)$.

(d) Else $u_k \in \mathcal{N}$. But by hypothesis on $\delta^\rho(\Phi_+)$, we know that $n \in \text{img}(\rho)$ implies $\delta_c^\rho(n)$ is not deducible. Thus $u_k \notin \text{img}(\rho)$ and so $\delta_a^\rho(u_k) = \delta_b^\rho(u_k)$, which leads to $N_k = D_k(M_c)$, i.e. same as case (a).

We showed that for such $M_d = C[N_1, \ldots, N_n]$, we have that $M_d\delta^\rho(\Phi_+)\downarrow = \delta_d^\rho(C[u_1, \ldots, u_n])$ and $M_d\delta^\rho(\Phi'_+)\downarrow = \delta_d^\rho(C[v_1, \ldots, v_n])$, which leads to $M_d\delta^\rho(\Phi_+)\downarrow = \delta_d^\rho(M\Phi_+\downarrow)$ and $M_d\delta^\rho(\Phi'_+)\downarrow = \delta_d^\rho(M\Phi'_+\downarrow)$. ■

Corollary 6: Let Φ and Φ' two frames in normal form such that $\text{dom}(\Phi) = \text{dom}(\Phi')$. Assume that Φ and Φ' have the same colors. Let $\mathcal{E}$ be a set of names and let ρ a renaming such that $\text{dom}(\rho) \cup \text{img}(\rho) \subseteq \mathcal{E}$, $\text{dom}(\rho) \cap fn(\Phi, \Phi') = \emptyset$ and $\rho_{|\text{dom}(\rho_0)} = \rho_0$. Let's denote $\Phi_+ = \Phi_a \uplus \Phi_b \uplus \Phi$ and $\Phi'_+ = \Phi_a \uplus \Phi_b \uplus \Phi'$.

If $\texttt{new}\ \mathcal{E}.\Phi_+$, $\texttt{new}\ \mathcal{E}.\Phi'_+$ are well-tagged, $\texttt{new}\ \mathcal{E}.\delta^\rho(\Phi_+) \sim \texttt{new}\ \mathcal{E}.\delta^\rho(\Phi'_+)$ and for all $u \in \{k, \mathsf{pk}(k), \mathsf{vk}(k) \mid k \in \text{img}(\rho) \cup \text{dom}(\rho)\}$, $\texttt{new}\ \mathcal{E}.\delta^\rho(\Phi_+) \vdash u$ or $\texttt{new}\ \mathcal{E}.\delta^\rho(\Phi'_+) \vdash u$ implies that $u \in fn(\delta^\rho(\Phi_a \uplus \Phi_b))$, then we have that $\texttt{new}\ \mathcal{E}.\Phi_+ \sim \texttt{new}\ \mathcal{E}.\Phi'_+$

Corollary 7: Let Φ be a colored frame in normal form. Let $\mathcal{E}$ be a set of names and let ρ a renaming such that $\text{dom}(\rho) \cup \text{img}(\rho) \subseteq \mathcal{E}$, $\text{dom}(\rho) \cap fn(\Phi) = \emptyset$ and $\rho_{|\text{dom}(\rho_0)} = \rho_0$. Let's denote $\Phi_+ = \Phi_a \uplus \Phi_b \uplus \Phi$.

If $\texttt{new}\ \mathcal{E}.\Phi_+$ is well-tagged and for all $u \in \{k, \mathsf{pk}(k), \mathsf{vk}(k) \mid k \in \text{img}(\rho) \cup \text{dom}(\rho)\}$, $\texttt{new}\ \mathcal{E}.\delta^\rho(\Phi_+) \vdash u$ implies that $u \in fn(\delta^\rho(\Phi_a \uplus \Phi_b))$, then we have that for all $u \in \{k, \mathsf{pk}(k), \mathsf{vk}(k) \mid k \in \text{img}(\rho)\}$, $\texttt{new}\ \mathcal{E}.\Phi_+ \vdash u$ implies that $u \in fn(\Phi_a)$,

For the next three lemmas, we consider the following: Let A and B be two closed processes respectively built on $\Sigma_a \cup \Sigma_0$ and $\Sigma_b \cup \Sigma_0$ such that all variables in A and B are of base type. Moreover, assume that $\mathsf{col}(A) = a$ and $\mathsf{col}(B) = b$. Let $\mathcal{E}$ be a set of names and a bijective renaming ρ on name of base type such that $\text{dom}(\rho) \cup \text{img}(\rho) \subseteq \mathcal{E}$, $\rho_{|\text{dom}(\rho_0)} = \rho_0$ and for all $k \in \text{dom}(\rho)$, k does not appear in A or B.

Let $\mathcal{C}h_a$ and $\mathcal{C}h_b$ be two sets of 'fresh' names of channel type such that $\mathcal{C}h_a \cap \mathcal{C}h_b = \emptyset$. Let $\rho_{\mathcal{C}h_a}$ (resp $\rho_{\mathcal{C}h_b}$) be a bijective renaming from $\mathcal{C}h(A)\backslash\mathcal{E}$ (resp. $\mathcal{C}h(B)\backslash\mathcal{E}$) to $\mathcal{C}h_a$ (resp. $\mathcal{C}h_b$). Furthermore, assume that $(\mathcal{C}h(A) \cap \mathcal{E}) \cap (\mathcal{C}h(B) \cap \mathcal{E}) = \emptyset$.

Let $S = (\mathcal{E}; \mathcal{P}_S; \Phi_S)$ and $D = (\mathcal{E}; \mathcal{P}_D; \Phi_D)$ such that $\mathcal{P}_S = \{[A]_a, [B]_b\}$, $\Phi_S = \Phi_a \uplus \Phi_b$, $\mathcal{P}_D = \{\delta_a^\rho([A]_a)\rho_{\mathcal{C}h_a}, \delta_b^\rho([B]_b)\rho_{\mathcal{C}h_b}\}$ and $\Phi_D = \delta^\rho(\Phi_S)$.

Lemma 17: Let D' be an extended process and tr a trace such that $D \xrightarrow{\mathsf{tr}} D'$. If $\mathsf{tr} = \mathsf{tr}_1.\mathsf{tr}_2.\mathsf{tr}_3$ with $\mathsf{tr}_2 \in \texttt{new}\ w.\texttt{out}(c_1, w).\tau^*.\texttt{in}(c_2, M)$, $c_1 \in \mathcal{C}h_i$, $c_2 \in \mathcal{C}h_j$ and $i \neq j$, then there exists two traces tr', tr'_2 such that $\mathsf{tr}' = \mathsf{tr}_1.\mathsf{tr}'_2.\mathsf{tr}_3$, $\mathsf{tr}'_2 \in \tau^*.\texttt{new}\ w.\texttt{out}(c_1, w).\texttt{in}(c_2, M).\tau^*$ and $D \xrightarrow{\mathsf{tr}'} D'$.

Proof: $D \xrightarrow{\mathsf{tr}} D'$ and $\mathsf{tr} = \mathsf{tr}_1.\mathsf{tr}_2.\mathsf{tr}_3$ with $\mathsf{tr}_2 \in \texttt{new}\ w.\texttt{out}(c_1, w).\tau^*.\texttt{in}(c_2, M)$ implies that there exists $D_1, \ldots, D_n$ such that $D \xrightarrow{\mathsf{tr}_1} D_1 \xrightarrow{\texttt{new}\ w.\texttt{out}(c_1,w)} D_2 \xrightarrow{\tau} \ldots \xrightarrow{\tau} D_{n-1} \xrightarrow{\texttt{in}(c_2,M)} D_n$. By hypothesis on D, we know that processes of different colors in D do not share public nor private channels. Thus, a τ action, i.e. ELSE, THEN and COMM, can only be applied on processes of same color. But, applying a τ action on a process of color i does not modify the processes whom color is $j \neq i$.

Let $S_1 = (\mathcal{E}; \mathcal{P}_a \uplus \mathcal{P}_b; \Phi)$ be an extended process such that for all $P \in \mathcal{P}_a$ (resp. $\mathcal{P}_b$), $\mathsf{col}(P) = a$ (resp. b). Assume that we apply a τ action, denoted τ_a, on S_1, i.e. there exists $\mathcal{P}'_a$ such that $S_1 \xrightarrow{\tau_a} (\mathcal{E}; \mathcal{P}'_a \uplus \mathcal{P}_b; \Phi) = S_2$. Assume now that we apply a τ action, denoted τ_b, on S_2, i.e. there exists $\mathcal{P}'_b$ such that $S_2 \xrightarrow{\tau_b} (\mathcal{E}; \mathcal{P}'_a \uplus \mathcal{P}'_b; \Phi) = S_3$. Hence we have that $S_1 \xrightarrow{\tau_a} S_2 \xrightarrow{\tau_b} S_3$. But we can see that $S_1 \xrightarrow{\tau_b} (\mathcal{E}; \mathcal{P}_a \uplus \mathcal{P}'_b; \Phi) \xrightarrow{\tau_a} S_3$. Hence it is possible to swap τ actions.

With a similar proof, we can show that if $S_1 \xrightarrow{\texttt{new}\ w.\texttt{out}(c_1,w)} S_2 \xrightarrow{\tau_b} S_3$, with τ_b a τ action initiated by a process of color b, then we also have that there exists S'_2 such that $S_1 \xrightarrow{\tau_b} S'_2 \xrightarrow{\texttt{new}\ w.\texttt{out}(c_1,w)} S_3$.

In The same way, we can show that if $S_1 \xrightarrow{\tau_a} S_2 \xrightarrow{\texttt{in}(c_2,M)} S_3$, with τ_a a τ action initiated by a process of color a, then we also have that there exists S'_2 such that $S_1 \xrightarrow{\texttt{in}(c_2,M)} S'_2 \xrightarrow{\tau_a} S_3$.

Hence, a simple induction on the number of τ actions in the derivation $D \xrightarrow{\mathsf{tr}_1.\mathsf{tr}_2} D'$ allows us to prove that there exists $D'_2, \ldots, D'_{n-1}$ and an index $k \in \{1, \ldots, n-2\}$ such that $D_1 \xrightarrow{\tau_b} D'_2 \xrightarrow{\tau_b} \ldots \xrightarrow{\tau_b} D'_k \xrightarrow{\texttt{new}\ w.\texttt{out}(c_1,w)} D'_{k+1} \xrightarrow{\texttt{in}(c_2,M)} D'_{k+1} \xrightarrow{\tau_a} \ldots \xrightarrow{\tau_a} D_n$. Hence the result holds. ■

Lemma 18: If for all $(\mathsf{tr}, \phi) \in \mathsf{trace}(D)$, for all $u \in \{k, \mathsf{vk}(k), \mathsf{pk}(k) \mid k \in \text{img}(\rho) \cup \text{dom}(\rho)\}$, $\phi \vdash u$ implies that $u \in fn(\Phi_D)$, then for all $S \xrightarrow{\mathsf{tr}} (\mathcal{E}; \mathcal{P}'_S; \Phi'_S)$ such that tr does not contain any internal communication between two processes of different colors, we have that Φ'_S is well-tagged and there exists $D \xrightarrow{\mathsf{tr}'} (\mathcal{E}; \mathcal{P}'_D; \Phi'_D)$ such that $\Phi'_D\downarrow = \delta^\rho(\Phi'_S\downarrow)$. Furthermore, if $\mathsf{tr} = \ell_1.\ldots.\ell_n$ then $\mathsf{tr}' = \ell'_1.\ldots.\ell'_n$ such that for all $k \in \{1, \ldots, n\}$,

- if $\ell_k = \texttt{new}\ w.\texttt{out}(c, w)$ is an output coming from a process colored by $i \in \{a, b\}$, then $\ell'_k = \texttt{new}\ w.\texttt{out}(c\rho_{\mathcal{C}h_i}, w)$
- if $\ell_k = \texttt{in}(c, M)$ is an input coming from a process colored by $i \in \{a, b\}$, then $\ell'_k = \texttt{in}(c\rho_{\mathcal{C}h_i}, M_i)$ with $M_i\Phi'_D\downarrow = \delta_i^\rho(M\Phi'_S\downarrow)$.
- if $\ell_k = \texttt{out}(c, d)$ is an output coming from a process colored by $i \in \{a, b\}$ with d a channel name, then $\ell'_k = \texttt{out}(c\rho_{\mathcal{C}h_i}, d\rho_{\mathcal{C}h_i})$
- if $\ell_k = \tau$, then $\ell'_k = \tau$.

Proof: We have that $S \xrightarrow{\mathsf{tr}} (\mathcal{E}; \mathcal{P}'_S; \Phi'_S)$. We will show by induction on $|\mathsf{tr}|$ the properties stated in the Lemma but also that there exists $(\mathcal{P}_a, \alpha_a)$ and $(\mathcal{P}_b, \alpha_b)$ two original well-tagged multi-sets of processes such that $\mathsf{col}(\mathcal{P}_a) = a$, $\mathsf{col}(\mathcal{P}_b) = b$, $\mathcal{P}'_S = \mathcal{P}_a\alpha_a \uplus \mathcal{P}_b\alpha_b$ and $\mathcal{P}'_D = \mathcal{P}^a_D \uplus \mathcal{P}^b_D$ with for all $i \in \{a, b\}$, $\mathcal{P}^i_D{\downarrow} = \delta^\rho_i(\mathcal{P}_i)\rho_{\mathcal{C}h_i}\delta^\rho_i(\alpha_i{\downarrow}){\downarrow}$.

Base case $|\mathsf{tr}| = 0$*:* In this case, we need to verify that S and D satisfy the wanted properties. By hypothesis on S and D, we have $S = (\mathcal{E}; \mathcal{P}_S; \Phi_S)$ and $D = (\mathcal{E}; \mathcal{P}_D; \Phi_D)$ such that $\mathcal{P}_S = \{[A]_a, [B]_b\}$, $\Phi_S = \Phi_a \uplus \Phi_b$, $\mathcal{P}_D = \{\delta^\rho_a([A]_a)\rho_{\mathcal{C}h_a}, \delta^\rho_b([B]_b)\rho_{\mathcal{C}h_b}\}$ and $\Phi_D = \delta^\rho(\Phi_S)$.

Thus by definition of an original well-tagged multi-set of processes, we define $\mathcal{P}_a = \{[A]_a\}$, $\alpha_a = id$, $\mathcal{P}_b = \{[B]_b\}$, $\alpha_b = id$. Hence, we have $\delta^\rho_i(\alpha_i{\downarrow}) = id$ and so $\mathcal{P}^i_D{\downarrow} = \delta^\rho_i(\mathcal{P}_i)\rho_{\mathcal{C}h_i}{\downarrow}$, for all $i \in \{a, b\}$.

Furthermore, we know by hypothesis that for all $(w \rhd u) \in \Phi_A \cup \Phi_B$, $u = \mathsf{f}(n)$ for some $\mathsf{f} \in \{\mathsf{pk}, \mathsf{vk}\}$ and $n \in \mathcal{N}$. Hence we have that $[u]_i = u$ and $\mathsf{test}_i(u) = \mathsf{true}$ for all $i \in \{a, b\}$, which implies that Φ_S is well-tagged and in normal form.

At last, $\delta^\rho_a(\Phi_A) = \Phi_A$ and $\delta^\rho_b(\Phi_B) = \Phi_B\rho_0^{-1}$, hence we have that $\delta^\rho(\Phi_S{\downarrow}) = \Phi_D{\downarrow}$.

Inductive case $|\mathsf{tr}| > 0$*:* In this case, we have that $\mathsf{tr} = \mathsf{tr}_1.\ell$ and so $S \xrightarrow{\mathsf{tr}_1} (\mathcal{E}; \mathcal{P}''_S; \Phi''_S) \xrightarrow{\ell} (\mathcal{E}; \mathcal{P}'_S; \Phi'_S)$ with no internal communication between two processes of different colors. Hence, by inductive hypothesis on tr_1, we have that there exists $D \xrightarrow{\mathsf{tr}'_1} (\mathcal{E}; \mathcal{P}''_D; \Phi''_D)$, $(\mathcal{P}'_a, \alpha'_a)$ and $(\mathcal{P}'_b, \alpha'_b)$ two original well-tagged multi sets of processes such that:

- $\Phi''_D{\downarrow} = \delta^\rho(\Phi''_S{\downarrow})$
- Φ''_S is well-tagged
- $\mathcal{P}''_S = \mathcal{P}'_a\alpha'_a \uplus \mathcal{P}'_b\alpha'_b$ and $\mathcal{P}''_D = \mathcal{P}^a_D{}' \uplus \mathcal{P}^b_D{}'$ with for all $i \in \{a, b\}$, $\mathcal{P}^i_D{}'{\downarrow} = \delta^\rho_i(\mathcal{P}'_i)\rho_{\mathcal{C}h_i}\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$.
- for all $k \in \{1, \ldots, |\mathsf{tr}| - 1\}$, ℓ_k satisfies the desired properties.

We proceed by case analysis on the rule applied for the transition $(\mathcal{E}; \mathcal{P}''_S; \Phi''_S) \xrightarrow{\ell} (\mathcal{E}; \mathcal{P}'_S; \Phi'_S)$. Since by hypothesis, we know that there is no internal communication between two processes of different colors, and $\mathcal{P}''_S = \mathcal{P}'_a\alpha'_a \uplus \mathcal{P}'_b\alpha'_b$ we can assume that a rule is applied on $\mathcal{P}'_i\alpha'_i$, with $i \in \{a, b\}$. Let $j \in \{a, b\}$ such that $i \neq j$

Case of the rule THEN*:* In this case, by definition of $(\mathcal{P}'_i, \alpha_i)$, there exists ϕ formula and Q_1, Q_2 processes and $\mathcal{Q}'_i \subseteq \mathcal{P}'_i$ such that $\mathcal{P}''_S = \{\mathtt{if}\ \phi\ \mathtt{then}\ [Q_1]_i\alpha'_i\ \mathtt{else}\ [Q_2]_i\alpha'_i\} \uplus \mathcal{Q}'_i\alpha'_i \uplus \mathcal{P}'_j\alpha'_j$, $\mathcal{P}'_S = \{[Q_1]_i\alpha'_i\} \uplus \mathcal{Q}'_i\alpha'_i \uplus \mathcal{P}'_j\alpha'_j$, $\Phi'_S = \Phi''_S$ and ϕ a conjunction of equations $(u = v)$. By definition, we have that $([Q_1]_i, \alpha'_i)$ and $([Q_2]_i, \alpha'_i)$ are both original well-tagged processes.

Furthermore, we have that for all equation $(u = v) \in \phi$, $u =_{\mathsf{E}} v$; and either (a) there exists u such that ϕ is the formula $\mathsf{test}_i([u]_i)\alpha'_i$, or (b) there exists u_1, u_2 such that ϕ is the formula $[u_1]_i\alpha'_i = [u_2]_i\alpha'_i$ and $\alpha'_i \vDash \mathsf{test}_i([u_1]_i) \wedge \mathsf{test}_i([u_2]_i)$.

But we also have $\mathcal{P}^i_D{}'{\downarrow} = \delta^\rho_i(\mathcal{P}'_i)\rho_{\mathcal{C}h_i}\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$, which means that there exists ϕ' formula and Q'_1, Q'_2 processes and $\mathcal{Q}^i_D \subseteq p^i_D{}'$ such that $\mathcal{P}''_D = \{\mathtt{if}\ \phi'\ \mathtt{then}\ Q'_1\ \mathtt{else}\ Q'_2)\} \uplus \mathcal{Q}^i_D \uplus \mathcal{P}^j_D{}'$ with $Q'_1{\downarrow} = \delta^\rho_i([Q_1]_i)\rho_{\mathcal{C}h_i}\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$, $Q'_2{\downarrow} = \delta^\rho_i([Q_2]_i)\rho_{\mathcal{C}h_i}\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$. Furthermore, in case (a) $\phi'{\downarrow}$ is the formula $\mathsf{test}_i(\delta^\rho_i([u]_i))\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$; and in case (b) ϕ' is the formula $(u' = v')$ where $u'{\downarrow} = \delta^\rho_i([u_1]_i)\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$ and $v'{\downarrow} = \delta^\rho_i([u_2]_i)\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$.

In case (a), For all equation $(u' = v') \in \phi$, $u' =_{\mathsf{E}} v'$ which is equivalent to $u'{\downarrow} = v'{\downarrow}$. Thus it is equivalent to $(\alpha'_i{\downarrow}) \vDash \mathsf{test}_i([u]_i)$. But by Lemma 8, we know that this is equivalent to $\delta^\rho_i(\alpha'_i{\downarrow}) \vDash \mathsf{test}_i(\delta^\rho_i([u]_i))$. Thus we have that for all equation $(u' = v')$ of $\mathsf{test}_i(\delta^\rho_i([u]_i))$, $u'\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow} = v'\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$. Since $\phi'{\downarrow}$ is the formula $\mathsf{test}_i(\delta^\rho_i([u]_i))\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$, we can conclude that for all equation $(u' = v')$ of ϕ', $u'{\downarrow} = v'{\downarrow}$ and so $u' =_{\mathsf{E}} v'$.

In case (b), we know that $\alpha{\downarrow} \vDash \mathsf{test}_i([u_1]_i) \wedge \mathsf{test}_i([u_2]_i)$ and $[u_1]_i\alpha'_i{\downarrow} = [u_2]_i\alpha{\downarrow}$. Thus by Corollary 2, we can deduce that $\delta^\rho_i([u_1]_i)\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow} = \delta^\rho_i([u_2]_i)\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$ which means that $u'{\downarrow} = v'{\downarrow}$ and so $u' =_{\mathsf{E}} v'$.

ϕ' being satisfied in both cases allows us to deduce that $(\mathcal{E}; \mathcal{P}''_D; \Phi''_D) \xrightarrow{\ell} (\mathcal{E}; \{Q'_1\} \uplus \mathcal{Q}^i_D \uplus \mathcal{P}^j_D{}'; \Phi''_D)$.

Thus, we have $\Phi'_D = \Phi''_D$, but $\Phi'_S = \Phi''_S$, hence we have that Φ'_S is well-tagged and $\Phi'_D{\downarrow} = \delta^\rho(\Phi'_S{\downarrow})$. Furthermore, let $\ell' = \ell$ and $\mathcal{P}_i = \{[Q_1]_i\} \uplus \mathcal{Q}'_i$, $\alpha_i = \alpha'_i$, $\mathcal{P}_j = \mathcal{P}'_j$, $\alpha_j = \alpha'_j$. By definition of $[Q_1]_i$, we have that $(\mathcal{P}_i, \alpha_i)$ and $(\mathcal{P}_j, \alpha_j)$ are both original well-tagged multisets of processes.

We already showed that $\mathcal{P}_S = \mathcal{P}_i\alpha_i \uplus \mathcal{P}_j\alpha_i$ and thanks to $Q'_1{\downarrow} = \delta^\rho_i([Q_1]_i)\rho_{\mathcal{C}h_i}\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$, we can deduce that $\mathcal{P}_D{\downarrow} = \delta^\rho_i(\mathcal{P}_i)\rho_{\mathcal{C}h_i}\delta^\rho_i(\alpha_i{\downarrow}){\downarrow} \uplus \delta^\rho_j(\mathcal{P}_j)\rho_{\mathcal{C}h_j}\delta^\rho_j(\alpha_j{\downarrow}){\downarrow}$.

Hence the result holds.

Case of the rule ELSE*:* This case similar to the rule THEN.

Case of the rule COMM*:* In this case, by definition of $(\mathcal{P}'_i, \alpha'_i)$, there exists u, x, p terms, Q_1, Q_2 processes and $\mathcal{Q}'_i \subseteq \mathcal{P}'_i$ such that $\mathcal{P}''_S = \{\mathtt{out}(p, [u]_i\alpha'_i).[Q_1]_i\alpha_i; \mathtt{in}(p, x).[Q_2]_i\alpha'_i\} \uplus \mathcal{Q}'_i\alpha'_i \uplus \mathcal{P}'_j\alpha'_j$, $\mathcal{P}'_S = \{[Q_1]_i\alpha'_i; [Q_2]_i\alpha'_i\{x \mapsto [u]_i\alpha_i\}\} \uplus \mathcal{Q}'_i\alpha'_i \uplus \mathcal{P}'_j\alpha'_j$, $\Phi'_S = \Phi''_S$ and $\alpha'_i \vDash \mathsf{test}_i([u]_i)$.

First of all, we trivially have that Φ'_S is well-tagged.

Furthermore, we have $\mathcal{P}^i_D{}'{\downarrow} = \delta^\rho_i(\mathcal{P}'_i)\rho_{\mathcal{C}h_i}\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$, which means that there exists p', u' terms and Q'_1, Q'_2, processes such that $\mathcal{P}''_D = \{\mathtt{out}(p', u').Q'_1; \mathtt{in}(p', x).Q'_2\} \uplus \mathcal{Q}^i_D \uplus \mathcal{P}^j_D{}'$ with $Q'_1{\downarrow} = \delta^\rho_i([Q_1]_i)\rho_{\mathcal{C}h_i}\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$, $Q'_2{\downarrow} = \delta^\rho_i([Q_2]_i)\rho_{\mathcal{C}h_i}\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$, $p' = p'{\downarrow} = \delta^\rho_i(p)\rho_{\mathcal{C}h_i} = p\rho_{\mathcal{C}h_i}$ and $u'{\downarrow} = \delta^\rho_i([u]_i)\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow}$.

Hence for $\ell' = \ell = \tau$, we have that $(\mathcal{E}; \mathcal{P}''_D; \Phi''_D) \xrightarrow{\ell'} (\mathcal{E}; Q'_1 \uplus Q'_2\{x \mapsto u'\} \uplus \mathcal{Q}^i_D \uplus \mathcal{P}^j_D{}'; \Phi''_D)$.

Let $\Phi'_D = \Phi''_D$. Since $\Phi'_S = \Phi''_S$, then by hypothesis, we have that $\Phi'_D{\downarrow} = \delta^\rho(\Phi'_S{\downarrow})$.

Let's denote $\alpha_i = \alpha'_i\{x \mapsto [u]_i\alpha_i\}$. We already know that $\alpha'_i \vDash \mathsf{test}_i([u]_i)$. Thus thanks to Lemma 7 and $\alpha'_i \vDash \mathsf{test}_i([u]_i)$, we deduce that $u'{\downarrow} = \delta^\rho_i([u]_i)\delta^\rho_i(\alpha'_i{\downarrow}){\downarrow} =$

$\delta_i^\rho([u]_i(\alpha_i'\downarrow))\downarrow = \delta_i^\rho([u]_i\alpha_i'\downarrow)$. This implies that $\delta_i^\rho(\alpha_i\downarrow) = \delta_i^\rho(\alpha_i')\{x \mapsto u'\downarrow\}$. But $Q_2'\{x \mapsto u'\}\downarrow = (Q_2'\downarrow)\{x \mapsto (u'\downarrow)\}\downarrow$, hence $Q_2'\{x \mapsto u'\}\downarrow = \delta_i^\rho([Q_2]_i)\rho_{\mathcal{C}h_i}\delta_i^\rho(\alpha_i\downarrow)\downarrow$.

Moreover, since $x \notin fv([Q_1]_i) \cup fv(\mathcal{Q}_i')$, we can deduce that $[Q_1]_i\alpha_i' = [Q_1]_i\alpha_i$, $Q_2'\downarrow = \delta_i^\rho([Q_2]_i)\rho_{\mathcal{C}h_i}\delta_i^\rho(\alpha_i\downarrow)\downarrow$ and $\mathcal{Q}_i'\alpha_i' = \mathcal{Q}_i'\alpha_i$. Thus, we have that $(\{[Q_1]_i, [Q_2]_i\} \uplus \mathcal{Q}_i', \alpha_i)$ is an original well-tagged multi-set of processes. Hence the result holds.

Case of the rule IN*:* In this case, by definition of $(\mathcal{P}_i', \alpha_i')$, there exists x, p, M terms, Q_1 process, $\mathcal{Q}_i' \subseteq \mathcal{P}_i'$, $M\Phi_S'' = u$, $fv(M) \subseteq \mathrm{dom}(\Phi_S'')$ and $fn(M) \cap \mathcal{E} = \emptyset$ such that $\mathcal{P}_S'' = \{\mathtt{in}(p, x).[Q_1]_i\alpha_i'\} \uplus \mathcal{Q}_i'\alpha_i' \uplus \mathcal{P}_j'\alpha_j'$, $\mathcal{P}_S' = \{[Q_1]_i\alpha_i'\{x \mapsto u\}\} \uplus \mathcal{Q}_i'\alpha_i' \uplus \mathcal{P}_j'\alpha_j'$, $\Phi_S' = \Phi_S''$ and $\alpha_i' \vDash \mathsf{test}_i([u]_i)$.

First of all, we trivially have that Φ_S' is well-tagged.

Furthermore, we have $\mathcal{P}_D^{i\,\prime}\downarrow = \delta_i^\rho(\mathcal{P}_i')\rho_{\mathcal{C}h_i}\delta_i^\rho(\alpha_i'\downarrow)\downarrow$, which means that there exists p' term and Q_1' process such that $\mathcal{P}_D'' = \{\mathtt{in}(p', x).Q_1'\} \uplus \mathcal{Q}_D^i \uplus \mathcal{P}_D^{j\,\prime}$ with $Q_1'\downarrow = \delta_i^\rho([Q_1]_i)\rho_{\mathcal{C}h_i}\delta_i^\rho(\alpha_i'\downarrow)\downarrow$ and $p' = p'\downarrow = \delta_i^\rho(p)\rho_{\mathcal{C}h_i} = p\rho_{\mathcal{C}h_i}$.

By hypothesis, we assumed that for all $v \in \{k, \mathsf{vk}(k), \mathsf{pk}(k) \mid k \in \mathrm{img}(\rho) \cup \mathrm{dom}(\rho)\}$, $\mathtt{new}\ \mathcal{E}.\Phi_D'' \not\vdash v$ implies that $v \in \Phi_D$, where $\Phi_D = \delta^\rho(\Phi_a \uplus \Phi_b)$.

Thus, thanks to Lemma 16, we have that there exists a term M_i such that $fv(M_i) \subseteq \mathrm{dom}(\Phi_D'')$, $fn(M_i) \cap \mathcal{E} = \emptyset$ and $M_i\Phi_D''\downarrow = \delta_i^\rho(M\Phi_S''\downarrow)$. Thus, with $\ell' = \mathtt{in}(p\rho_{\mathcal{C}h_i}, M_i)$, we can deduce that $(\mathcal{E}; \mathcal{P}_D''; \Phi_D'') \xrightarrow{\ell'} (\mathcal{E}; Q_1'\{x \mapsto u'\} \uplus \mathcal{Q}_D^i \uplus \mathcal{P}_D^{j\,\prime}; \Phi_D'')$ where $u' = M_i\Phi_D''$.

But $M\Phi_S''\downarrow = u\downarrow$ and $M_i\Phi_D''\downarrow = u'\downarrow$. Thus, we have that $\delta_i^\rho(u\downarrow) = u'\downarrow$. Since $Q_1'\downarrow = \delta_i^\rho([Q_1]_i)\rho_{\mathcal{C}h_i}\delta_i^\rho(\alpha_i'\downarrow)\downarrow$, we deduce that $Q_1'\{x \mapsto u'\}\downarrow = \delta_i^\rho([Q_1]_i)\rho_{\mathcal{C}h_i}\delta_i^\rho(\alpha_i'\downarrow)\{x \mapsto \delta_i^\rho(u\downarrow)\}\downarrow$.

Let $\alpha_i = \alpha_i'\{x \mapsto u\}$, we have that that $\delta_i^\rho(\alpha_i\downarrow) = \delta_i^\rho(\alpha_i'\downarrow)\{x \mapsto \delta_i^\rho(u\downarrow)\}$ and so $Q_1'\{x \mapsto u'\}\downarrow = \delta_i^\rho([Q_1]_i)\rho_{\mathcal{C}h_i}\delta_i^\rho(\alpha_i\downarrow)\downarrow$ and $[Q_1]_i\alpha_i'\{x \mapsto u\} = [Q_1]_i\alpha_i$. Moreover, since $x \notin fv(\mathcal{Q}_i')$, we can deduce that , $Q_1'\downarrow = \delta_i^\rho([Q_1]_i)\rho_{\mathcal{C}h_i}\delta_i^\rho(\alpha_i\downarrow)\downarrow$ and $\mathcal{Q}_i'\alpha_i' = \mathcal{Q}_i'\alpha_i$. Thus, we have that $(\{[Q_1]_i\} \uplus \mathcal{Q}_i', \alpha_i)$ is an original well-tagged multi-set of processes. Hence the result holds.

Case of the rule OUT-T*:* In such a case, by definition of $(\mathcal{P}_i', \alpha_i')$, there exists u, p terms and Q_1 process, $\mathcal{Q}_i' \subseteq \mathcal{P}_i'$ such that $\mathcal{P}_S'' = \{\mathtt{out}(p, [u]_i\alpha_i').[Q_1]_i\alpha_i'\} \uplus \mathcal{Q}_i'\alpha_i' \uplus \mathcal{P}_j'\alpha_j'$, $\mathcal{P}_S' = \{[Q_1]_i\alpha_i'\} \uplus \mathcal{Q}_i'\alpha_i' \uplus \mathcal{P}_j'\alpha_j'$, $\Phi_S' = \Phi_S'' \uplus \{w_n \rhd [u]_i\alpha_i'\}$ and $\alpha_i' \vDash \mathsf{test}_i([u]_i)$. Furthermore, we have that $\ell = \mathtt{new}\ w_n.\mathtt{out}(p, w_n)$.

Moreover, we have $\mathcal{P}_D^{i\,\prime}\downarrow = \delta_i^\rho(\mathcal{P}_i')\rho_{\mathcal{C}h_i}\delta_i^\rho(\alpha_i'\downarrow)\downarrow$, which means that there exists p', u' terms and Q_1' process such that $\mathcal{P}_D'' = \{\mathtt{out}(p', u').Q_1'\} \uplus \mathcal{Q}_D^i \uplus \mathcal{P}_D^{j\,\prime}$ with $Q_1'\downarrow = \delta_i^\rho([Q_1]_i)\rho_{\mathcal{C}h_i}\delta_i^\rho(\alpha_i'\downarrow)\downarrow$, $p' = p'\downarrow = \delta_i^\rho(p)\rho_{\mathcal{C}h_i} = p\rho_{\mathcal{C}h_i}$ and $u'\downarrow = \delta_i^\rho([u]_i)\delta_i^\rho(\alpha_i'\downarrow)\downarrow$.

Hence for $\ell' = \mathtt{new}\ w_n.\mathtt{out}(p\rho_{\mathcal{C}h_i}, w_n)$, we have that $(\mathcal{E}; \mathcal{P}_D''; \Phi_D'') \xrightarrow{\ell'} (\mathcal{E}; \{Q_1'\} \uplus \mathcal{Q}_D^i \uplus \mathcal{P}_D^{j\,\prime}; \Phi_D')$.

We first need to show that Φ_S' is well-tagged. Since Φ_S'' is well tag, we only need to focus on the new term $[u]_i\alpha_i'$. Let $x \in \mathrm{dom}(\alpha_i')$, we know that x was initially a variable from S. Thus, x was introduced by an input $\mathtt{in}(c, x)$ for some c and so there exists a transition ℓ_x in tr_1 such that this transition reduces $\mathtt{in}(c, x)$. If $\ell_x = \mathtt{in}(c, M)$ (i.e. the rule IN), then we trivially have that the result holds with M; else $\ell_x = \tau$ (i.e. the rule COMM). But in the case $\ell_x = \tau$, we know that the output comes from a process colored by i and so there exists v such that $x\alpha_i' = [v]_i\alpha_i'$. Thus with a simple induction on the size of tr_1, we can show that Φ_S' is well-tagged.

Since all variables in $[u]_i$ are colored by i, then thanks to Lemma 7 and $\alpha_i' \vDash \mathsf{test}_i([u]_i)$, $\delta_i^\rho([u]_i)\delta_i^\rho(\alpha_i'\downarrow)\downarrow = \delta_i^\rho([u]_i(\alpha_i'\downarrow))\downarrow = \delta_i^\rho([u]_i\alpha_i'\downarrow) = \delta_i^\rho(u\downarrow)$. But $u'\downarrow = \delta_i^\rho([u]_i)\delta_i^\rho(\alpha_i'\downarrow)\downarrow$, which means that $u'\downarrow = \delta_i^\rho(u\downarrow)$. Since $\mathsf{col}(w_n) = i$, we can conclude that $\Phi_D'\downarrow = \delta^\rho(\Phi_S'\downarrow)$. Hence the result holds.

At last, let $\alpha_i = \alpha_i'$, we have that $(\{[Q_1]_i\} \uplus \mathcal{Q}_i', \alpha_i)$ is an original well-tagged multi-set of processes. Hence our result holds.

Case of the rule OUT-CH*:* Obvious since $\mathrm{dom}(\rho)$ and $\mathrm{img}(\rho)$ only contain names of base type and so if $\ell = \mathtt{out}(c, d)$, since c and d are public then $\ell' = \mathtt{out}(c\rho_{\mathcal{C}h_i}, d\rho_{\mathcal{C}h_i})$

Case of the rule OPEN-CH*:* Obvious since $\mathrm{dom}(\rho)$ and $\mathrm{img}(\rho)$ only contain names of base type and so if $\ell = \mathtt{new}\ ch\mathtt{out}(c, ch)$, since c is public and $\mathrm{dom}(\rho_{\mathcal{C}h_i})$ is only composed of public channels, then $\ell' = \mathtt{new}\ ch\mathtt{out}(c\rho_{\mathcal{C}h_i}, ch)$

Case of the rule PAR*:* Obvious ■

Lemma 19: Let $\Phi_+ = \Phi_a \uplus \Phi_b \uplus \Phi$ be a ground well-tagged frame. Let $\mathsf{tr} = \ell_1.\ldots.\ell_n$ be a label and Φ'_D a ground frame. If $D \xrightarrow{\mathsf{tr}} (\mathcal{E}; \mathcal{P}'_D; \Phi'_D)$ such that:

- $\mathtt{new}\ \mathcal{E}.\Phi'_D \sim \mathtt{new}\ \mathcal{E}.\delta^\rho(\Phi_+)$
- Φ'_D and Φ_+ have the same colors
- for all $u \in \{k, \mathsf{vk}(k), \mathsf{pk}(k) \mid k \in \mathrm{img}(\rho) \cup \mathrm{dom}(\rho)\}$, $\Phi_+ \vdash u$ or $\Phi'_D \vdash u$ implies that $u \in \Phi_D$
- for all $k \in \{1, \ldots, n\}$, if $\ell_k = \mathtt{in}(c, M_k)$ with $c \in \mathcal{C}h_i$, $i \in \{a, b\}$ then there exists M_k^i such that $M_k\delta^\rho(\Phi_+\downarrow)\downarrow = \delta_i^\rho(M_k^i\Phi_+\downarrow)$.

then we have that there exists a label $\mathsf{tr}' = \ell'_1.\ldots.\ell'_n$ and a well-tagged frame Φ'_S such that $S \xrightarrow{\mathsf{tr}'} (\mathcal{E}; \emptyset'_S; \Phi'_S)$, $\Phi'_D\downarrow = \delta^\rho(\Phi'_S\downarrow)$, and for all $k \in \{1, \ldots, n\}$,

- if $\ell_k = \mathtt{new}\ w.\mathtt{out}(c, w)$ with $c \in \mathcal{C}h_i$, $i \in \{a, b\}$ then $\ell'_k = \mathtt{new}\ w.\mathtt{out}(c\rho_{\mathcal{C}h_i}^{-1}, w)$
- if $\ell_k = \mathtt{in}(c, M_k)$ with $c \in \mathcal{C}h_i$, $i \in \{a, b\}$, then $\ell'_k = \mathtt{in}(c\rho_{\mathcal{C}h_i}^{-1}, M_k^i)$.
- if $\ell_k = \mathtt{out}(c, d)$ with $c \in \mathcal{C}h_i$, $i \in \{a, b\}$ and d a channel name, then $\ell'_k = \mathtt{out}(c\rho_{\mathcal{C}h_i}^{-1}, d\rho_{\mathcal{C}h_i}^{-1})$
- if $\ell_k = \tau$ then $\ell'_k = \tau$

Proof: The proof of this Lemma is very similar to the proof of Lemma 18. Indeed, in the proof of Lemma 18, we used Lemma 7 to show that $\alpha \vDash \mathsf{test}_i([u]_i)$ implies that

$\delta_i^\rho(\alpha) \vDash \mathsf{test}_i(\delta_i^\rho([u]_i))$. But Lemma 7 shows that those two properties are equivalent. The same goes for Corollary 2

The only difference is that in the case of the rule IN, Lemma 16 cannot be called. Intuitively, Lemma 16 allows us to show that for all recipes applied on Φ'_S, we can create an equivalent recipe for Φ'_D; but not the other way around. On the other hand, the new hypothesis is added in this lemma (the last one) which is the corresponding result of Lemma 16.

Indeed we have $\texttt{new}\ \mathcal{E}.\Phi_+ \sim \texttt{new}\ \mathcal{E}.\Phi'_D$. But with our inductive step, we would have $\Phi''_D = \Phi'_D$, Φ''_S well tagged and $\Phi''_D{\downarrow} = \delta^\rho(\Phi''_S{\downarrow})$. Thus we have that $\texttt{new}\ \mathcal{E}.(\Phi_+{\downarrow}) \sim \texttt{new}\ \mathcal{E}.(\Phi''_S{\downarrow})$. Let M_k be the recipe from an input. By hypothesis, we have that $M_k\delta^\rho(\Phi_+{\downarrow}){\downarrow} = \delta_i^\rho(M_k^i\Phi_+{\downarrow})$. But by Lemma 16, there exists M such that $M\delta^\rho(\Phi_+{\downarrow}){\downarrow} = \delta_i^\rho(M_k^i\Phi_+{\downarrow})$ and $M\delta^\rho(\Phi''_S{\downarrow}){\downarrow} = \delta_i^\rho(M_k^i\Phi''_S{\downarrow})$. But $M_k\delta^\rho(\Phi_+{\downarrow}){\downarrow} = M\delta^\rho(\Phi_+{\downarrow}){\downarrow}$ implies $M_k\delta^\rho(\Phi''_S{\downarrow}){\downarrow} = M\delta^\rho(\Phi''_S{\downarrow}){\downarrow}$, which allows us to conclude that $M_k\delta^\rho(\Phi''_S{\downarrow}){\downarrow} = M_k\Phi''_D{\downarrow} = \delta_i^\rho(M_k^i\Phi''_S{\downarrow})$. ■

Lemma 20: Let C be a composition context. Let $\overline{P_A}$ (resp. $\overline{P_B}$) be a sequences of plain processes built on $\Sigma_a \cup \Sigma_0$ (resp. $\Sigma_b \cup \Sigma_0$).

Let $D = (\mathcal{K}_0; C[[\overline{P_A}]_a\rho_{\mathcal{C}h_a}] \mid C[[\overline{P_B}]_b\rho_0^{-1}\rho_{\mathcal{C}h_b}]; \Phi_a \uplus \Phi_b\rho_0^{-1})$ and $S = (\mathcal{K}_0; C[[\overline{P_A}]_a \mid [\overline{P_B}]_b]; \Phi_a \uplus \Phi_b)$, we have that for all $(\mathsf{tr}, \texttt{new}\ \mathcal{E}.\Phi) \in \mathsf{trace}(D)$ (resp. $(\mathsf{tr}, \texttt{new}\ \mathcal{E}.\Phi) \in \mathsf{trace}(S)$), there exists a renaming ρ and two extended processes $S' = (\mathcal{E}; \{P_a, P_b\}; \Phi_a \uplus \Phi_b)$ and $D' = (\mathcal{E}; \{P_a\rho_{\mathcal{C}h_a}, P_b\rho^{-1}\rho_{\mathcal{C}h_b}\}; \Phi_a \uplus \Phi_b\rho^{-1})$ without replication or new such that

- $\rho_{|\,\mathrm{dom}(\rho_0)} = \rho_0$, $\mathrm{dom}(\rho) \cup \mathrm{img}(\rho) \subseteq \mathcal{E}$ and $\mathrm{dom}(\rho)$ does not appear in $\{P_a, P_b\}$
- for all $i \in \{a, b\}$, P_i is colored by i and there exists P'_i built on $\Sigma_i \cup \Sigma_0$ such that such that $P_i = [P'_i]_i$
- $(\mathsf{tr}, \texttt{new}\ \mathcal{E}.\Phi) \in \mathsf{trace}(D')$ (resp. $\mathsf{trace}(S')$)
- $\mathsf{trace}(D') \subseteq \mathsf{trace}(D)$
- $\mathsf{trace}(S') \subseteq \mathsf{trace}(S)$

Note that $\delta_a^\rho(P_a) = P_a$ and $\delta_b^\rho(P_b) = P_b\rho^{-1}$

Proof: (of Theorem 2) Before we start the proof, we rename Φ_0 into Φ_a. We color Φ_a by a. If we assume that $\Phi_a = \{w_1^a \rhd u_1, \ldots, w_n^a \rhd u_n\}$, we build the frame Φ_b, colored by b such that $\Phi_b = \{w_1^b \rhd u_1, \ldots, w_n^b \rhd u_n\}$. At last, let ρ_0 be the bijective renaming such that $\mathrm{img}(\rho_0) = \mathcal{K}_0$ (so $fn(\Phi_a) \subseteq \mathrm{img}(\rho_0)$) and $\mathrm{dom}(\rho_0)$ are composed of fresh names.

Let $\mathcal{C}h_a$ and $\mathcal{C}h_b$ be two sets of fresh channel type name. Furthermore, let $\rho_{\mathcal{C}h_a}$ be a bijective renaming from the public channel of $(\mathcal{K}_0; C[[\overline{P_A}]_a]; \Phi_0)$ and $(\mathcal{K}_0; C'[[\overline{P'_A}]_a]; \Phi_0)$ to $\mathcal{C}h_a$. We define $\rho_{\mathcal{C}h_b}$ in the same way.

We know by hypothesis that $(\mathcal{K}_0; C[[\overline{P_A}]_a]; \Phi_0) \approx (\mathcal{K}_0; C'[[\overline{P'_A}]_a]; \Phi_0)$ and $(\mathcal{K}_0; C[[\overline{P_B}]_b]; \Phi_0) \approx (\mathcal{K}_0; C'[[\overline{P'_B}]_b]; \Phi_0)$. But the trace equivalence is stable under renaming. Thus, we have that $(\mathcal{K}_0; C[[\overline{P_A}]_a\rho_{\mathcal{C}h_a}]; \Phi_a) \approx (\mathcal{K}_0; C'[[\overline{P'_A}]_a\rho_{\mathcal{C}h_a}]; \Phi_a)$ and $(\mathcal{K}_0\rho_0^{-1}; C[[\overline{P_B}]_b\rho_{\mathcal{C}h_b}\rho_0^{-1}]; \Phi_b\rho_0^{-1}) \approx (\mathcal{K}_0\rho_0^{-1}; C'[[\overline{P'_B}]_b\rho_{\mathcal{C}h_b}\rho_0^{-1}]; \Phi_b\rho_0^{-1})$. Since the sets $\mathcal{K}_0\rho_0^{-1}$ and $\mathcal{K}_0$ are disjoint, and $\mathrm{dom}(\Phi_a) \cap \mathrm{dom}(\Phi_b) = \emptyset$, we can compose the two equivalences such that if we denote $D = (\mathcal{K}_0 \uplus \mathcal{K}_0\rho_0^{-1}; C[[\overline{P_A}]_a\rho_{\mathcal{C}h_a}] \mid C[[\overline{P_B}]_b\rho_{\mathcal{C}h_b}\rho_0^{-1}]; \Phi_a \uplus \Phi_b\rho_0^{-1})$ and $D' = (\mathcal{K}_0 \uplus \mathcal{K}_0\rho_0^{-1}; C'[[\overline{P'_A}]_a\rho_{\mathcal{C}h_a}] \mid C'[[\overline{P'_B}]_b\rho_{\mathcal{C}h_b}\rho_0^{-1}]; \Phi_a \uplus \Phi_b\rho_0^{-1})$, we have that $D \approx D'$.

Let's denote now $S = (\mathcal{K}_0 \uplus \mathcal{K}_0\rho_0^{-1}; C[[\overline{P_A}]_a \mid [\overline{P_B}]_b]; \Phi_a \uplus \Phi_b)$ and $S' = (\mathcal{K}_0 \uplus \mathcal{K}_0\rho_0^{-1}; C'[[\overline{P'_A}]_a \mid [\overline{P'_B}]_b]; \Phi_a \uplus \Phi_b)$.

We will show that $S \approx S'$. Indeed, since $\Phi_0 = \Phi_a$ and Φ_b have exactly the same terms and since the names in $\mathcal{K}_0\rho_0^{-1}$ do not appear in $\overline{P_A}, \overline{P'_A}, \overline{P_B}$ or $\overline{P'_B}$, then $S \approx S'$ is equivalent to $(\mathcal{K}_0; C[[\overline{P_A}]_a \mid [\overline{P_B}]_b]; \Phi_0) \approx (\mathcal{K}_0; C'[[\overline{P'_A}]_a \mid [\overline{P'_B}]_b]; \Phi_0)$.

Let $(\mathsf{tr}, \texttt{new}\ \mathcal{E}.\Phi) \in \mathsf{trace}(S)$, by Lemma 20, there exists a renaming ρ and two extended processes $S_1 = (\mathcal{E}; \{P_a, P_b\}; \Phi_a \uplus \Phi_b)$ and $D_1 = (\mathcal{E}; \{P_a\rho_{\mathcal{C}h_a}, P_b\rho^{-1}\rho_{\mathcal{C}h_b}\}; \Phi_a \uplus \Phi_b\rho^{-1})$ without replication or new such that

- $\rho_{|\,\mathrm{dom}(\rho_0)} = \rho_0$, $\mathrm{dom}(\rho) \cup \mathrm{img}(\rho) \subseteq \mathcal{E}$ and $\mathrm{dom}(\rho)$ does not appear in $\{P_a, P_b\}$
- for all $i \in \{a, b\}$, P_i is colored by i and there exists P'_i built on $\Sigma_i \cup \Sigma_0$ such that such that $P_i = [P'_i]_i$
- $(\mathsf{tr}, \texttt{new}\ \mathcal{E}.\Phi) \in \mathsf{trace}(S_1)$
- $\mathsf{trace}(D_1) \subseteq \mathsf{trace}(D)$

However, in order to apply Lemma 18, we need to get rid of the possible internal communications between two processes of different colors in $(\mathsf{tr}, \texttt{new}\ \mathcal{E}.\Phi)$. Since $(\mathsf{tr}, \texttt{new}\ \mathcal{E}.\Phi) \in \mathsf{trace}(S_1)$ and S_1 does not contain any new, we have that $S_1 \overset{\mathsf{tr}}{\Rightarrow} (\mathcal{E}; \mathcal{P}; \Phi)$, for some $\mathcal{P}$. Let's denote $\widetilde{\mathsf{tr}}$ the label from tr with τ actions apparent such that $S_1 \xrightarrow{\widetilde{\mathsf{tr}}} (\mathcal{E}; \mathcal{P}; \Phi)$. We show by induction on $\#\{\tau \in \widetilde{\mathsf{tr}}\}$ that there exists $\widetilde{\mathsf{tr}'}$ such that $S_1 \xrightarrow{\widetilde{\mathsf{tr}'}} (\mathcal{E}; \mathcal{P}; \Phi \cup \Phi_+)$, for some Φ_+ and $\widetilde{\mathsf{tr}'}$ does not contain any internal communications between two processes of different colors:

Base case $\#\{\tau \in \widetilde{\mathsf{tr}}\} = 0$*:* The result trivially holds since there is no τ action.

Inductive step $\#\{\tau \in \widetilde{\mathsf{tr}}\} > 0$*:* Assume that there is an internal communication between two processes of different colors (if not the result holds trivially). Thus there exists $\mathsf{tr}_1, \mathsf{tr}_2$ such that $\widetilde{\mathsf{tr}} = \mathsf{tr}_1.\tau.\mathsf{tr}_2$ and $S_1 \xrightarrow{\mathsf{tr}_1} (\mathcal{E}; \mathcal{P}_1; \Phi_1) \xrightarrow{\tau} (\mathcal{E}; \mathcal{P}_2; \Phi_1) \xrightarrow{\mathsf{tr}_2} (\mathcal{E}; \mathcal{P}; \Phi)$. Since τ is an internal communication, then $\mathcal{P}_1 = \{\mathtt{in}(c, x).P_1; \mathtt{out}(c, u).P_2\} \uplus \mathcal{Q}$ and $\mathcal{P}_2 = \{P_1\{x \mapsto u\}; P_2\} \uplus \mathcal{Q}$, for some $P_1, P_2, \mathcal{Q}, c, x, u$. But by hypothesis, we know that processes of different colors do not share private channels. Thus c is a public channel. Hence, we have that $(\mathcal{E}; \mathcal{P}_1; \Phi_1) \xrightarrow{\texttt{new}\ w.\mathtt{out}(c,w)} P'$ where $P' = (\mathcal{E}; \{\mathtt{in}(c, x).P_1; P_2\} \uplus \mathcal{Q}; \Phi_1 \uplus \{w \rhd u\})$. Once again, since c is a public channel, we now have that $P' \xrightarrow{\mathtt{in}(c,w)} (\mathcal{E}; \mathcal{P}_2; \Phi \uplus \{w \rhd u\})$. A simple induction on the rest of the trace allows us to show that $(\mathcal{E}; \mathcal{P}_2; \Phi \uplus \{w \rhd$

$u\}) \xrightarrow{\mathsf{tr}_2} (\mathcal{E};\mathcal{P};\Phi \uplus \{w \rhd u\})$. Since we removed the τ action without adding new ones, we can apply our inductive hypothesis on $\mathsf{tr}_1.\mathtt{new}\ w.\mathtt{out}(c,w).\mathtt{in}(c,w).\mathsf{tr}_2$ in order to conclude.

We showed that $S_1 \xrightarrow{\widetilde{\mathsf{tr}'}} (\mathcal{E};\mathcal{P};\Phi\cup\Phi_+)$, for some Φ_+ and $\widetilde{\mathsf{tr}'}$ does not contain any internal communication between two processes of different colors. Let's denote $\Phi_S = \Phi \cup \Phi_+$, we have that $(\widetilde{\mathsf{tr}'}, \mathtt{new}\ \mathcal{E}.\Phi_S) \in \mathsf{trace}(S_1)$. Thanks to Lemma 18, we can deduce that Φ_S is well-tagged and there exists $D_1 \xrightarrow{\widetilde{\mathsf{tr}''}} (\mathcal{E};\mathcal{P}_D;\Phi_D)$ such that $\Phi_D{\downarrow} = \delta^\rho(\Phi_S{\downarrow})$. Furthermore, if $\widetilde{\mathsf{tr}'} = \ell_1.\ldots.\ell_n$ then $\widetilde{\mathsf{tr}''} = \ell'_1.\ldots.\ell'_n$ such that for all $k \in \{1,\ldots,n\}$,

- if $\ell_k = \mathtt{new}\ w.\mathtt{out}(c,w)$ is an output coming from a process colored by $i \in \{a,b\}$, then $\ell'_k = \mathtt{new}\ w.\mathtt{out}(c\rho_{\mathcal{C}h_i}, w)$
- if $\ell_k = \mathtt{in}(c,M)$ is an input coming from a process colored by $i \in \{a,b\}$, then $\ell'_k = \mathtt{in}(c\rho_{\mathcal{C}h_i}, M_i)$ with $M_i\Phi_D{\downarrow} = \delta^\rho_i(M\Phi_S{\downarrow})$.
- if $\ell_k = \mathtt{out}(c,d)$ is an output coming from a process colored by $i \in \{a,b\}$ with d a channel name, then $\ell'_k = \mathtt{out}(c\rho_{\mathcal{C}h_i}, d\rho_{\mathcal{C}h_i})$
- if $\ell_k = \tau$, then $\ell'_k = \tau$.

Hence we have that $(\widetilde{\mathsf{tr}''}, \mathtt{new}\mathcal{E}.\Phi_D) \in \mathsf{trace}(D_1)$. But we showed earlier that $\mathsf{trace}(D_1) \subseteq \mathsf{trace}(D)$, hence we have that $(\widetilde{\mathsf{tr}''}, \mathtt{new}\mathcal{E}.\Phi_D) \in \mathsf{trace}(D)$. Furthermore, we showed that $D \approx D'$, hence we deduce that there exists ϕ'_D such that $(\widetilde{\mathsf{tr}''}, \phi'_D) \in \mathsf{trace}(D')$ and $\mathtt{new}\mathcal{E}.\Phi_D \sim \phi'_D$.

Once again by Lemma 20, there exists a renaming ρ' and two extended processes $S'_1 = (\mathcal{E}';\{P'_a,P'_b\};\Phi_a \uplus \Phi_b)$ and $D'_1 = (\mathcal{E}';\{P'_a\rho_{\mathcal{C}h_a}, P'_b\rho^{-1}\rho_{\mathcal{C}h_b}\};\Phi_a \uplus \Phi_b\rho^{-1})$ without replication or new such that

- $\rho'_{|\,\mathrm{dom}(\rho_0)} = \rho_0$, $\mathrm{dom}(\rho') \cup \mathrm{img}(\rho') \subseteq \mathcal{E}'$ and $\mathrm{dom}(\rho')$ does not appear in $\{P'_a, P'_b\}$
- for all $i \in \{a,b\}$, P'_i is colored by i and there exists P''_i built on $\Sigma_i \cup \Sigma_0$ msuch that such that $P'_i = [P''_i]_i$
- $(\widetilde{\mathsf{tr}''}, \phi'_D) \in \mathsf{trace}(D'_1)$
- $\mathsf{trace}(S'_1) \subseteq \mathsf{trace}(S')$

Since our processes do not contain new, we can assume that $\mathcal{E} = \mathcal{E}'$ and $\rho = \rho'$ (if not we can apply some renaming on private name in $\mathcal{E}$ or $\mathcal{E}'$ in order to make them equal). Thus, there exists Φ'_D thus that $\mathtt{new}\mathcal{E}.\Phi'_D = \phi'_D$ and $(\widetilde{\mathsf{tr}''}, \mathtt{new}\ \mathcal{E}.\Phi'_D) \in \mathsf{trace}(D'_1)$.

Let's summarize what we have proved so far. We had $(\mathsf{tr},\Phi) \in \mathsf{trace}(S)$. We modified this trace into $(\widetilde{\mathsf{tr}'}, \Phi\uplus\Phi_+)$ where external actions $\mathtt{new}\ w.\mathtt{out}(c,w).\mathtt{in}(c,w)$ replace some τ actions. Then Lemma 20 allowed us to build the trace $(\widetilde{\mathsf{tr}''}, \mathtt{new}\mathcal{E}.\Phi_D) \in \mathsf{trace}(D_1)$ which only modify the terms in the trace but not the actions themselves. At last, we know $(\widetilde{\mathsf{tr}''}, \mathtt{new}\mathcal{E}.\Phi'_D) \in \mathsf{trace}(D'_1)$. Hence, $D'_1 \overset{\widetilde{\mathsf{tr}''}}{\Rightarrow} (\mathcal{E};\mathcal{P}';\Phi'_D)$ which means that there exists $\underline{\mathsf{tr}''}$ such that $D'_1 \xrightarrow{\underline{\mathsf{tr}''}} (\mathcal{E};\mathcal{P}';\Phi'_D)$. But, it is possible that there exists $\ell \in \tau^*$ such that $\mathtt{new}\ w.\mathtt{out}(c_a,w).\ell.\mathtt{in}(c_b,w) \in \underline{\mathsf{tr}''}$. But, thanks to Lemma 17, we can assume that this case does not occur and so there is no τ action between $\mathtt{new}\ w.\mathtt{out}(c_a,w)$ and $\mathtt{in}(c_b,w)$, for any $w \in \mathrm{dom}(\Phi'_D)$.

Since $\mathtt{new}\ \mathcal{E}.\Phi_D \sim \mathtt{new}\ \mathcal{E}.\Phi'_D$, Φ_S is a well-tagged frame and $\Phi_D{\downarrow} = \delta^\rho(\Phi_S{\downarrow})$, we can apply Lemma 19 on $(\underline{\mathsf{tr}''}, \mathtt{new}\ \mathcal{E}.\Phi'_D)$ which allow us to deduce that there exists a well-tagged frame Φ'_S such that $(\underline{\mathsf{tr}'''}, \mathtt{new}\ \mathcal{E}.\Phi'_S) \in \mathsf{trace}(S'_1)$, $\Phi'_D{\downarrow} = \delta^\rho(\Phi'_S{\downarrow})$, and if $\underline{\mathsf{tr}''} = \ell_1.\ldots.\ell_n$ then $\underline{\mathsf{tr}'''} = \ell'_1.\ldots.\ell'_n$ and for all $k \in \{1,\ldots,n\}$,

- if $\ell_k = \mathtt{new}\ w.\mathtt{out}(c,w)$ with $c \in \mathcal{C}h_i$, $i \in \{a,b\}$ then $\ell'_k = \mathtt{new}\ w.\mathtt{out}(c\rho^{-1}_{\mathcal{C}h_i}, w)$
- if $\ell_k = \mathtt{in}(c,M_k)$ with $c \in \mathcal{C}h_i$, $i \in \{a,b\}$, then $\ell'_k = \mathtt{in}(c\rho^{-1}_{\mathcal{C}h_i}, M^i_k)$.
- if $\ell_k = \mathtt{out}(c,d)$ with $c \in \mathcal{C}h_i$, $i \in \{a,b\}$ and d a channel name, then $\ell'_k = \mathtt{out}(c\rho^{-1}_{\mathcal{C}h_i}, d\rho^{-1}_{\mathcal{C}h_i})$
- if $\ell_k = \tau$ then $\ell'_k = \tau$

where $\underline{\mathsf{tr}''}$ and $\underline{\mathsf{tr}'''}$ are the respective label of tr'' and tr''' with τ actions. Furthermore, the couples (M_k, M^i_k) were generated by the application of Lemma 18 earlier on the trace $\widetilde{\mathsf{tr}'}$. Note that it is possible because the channels (public and private) of processes with different colors are disjoints in D_1 and D'_1. For example, if we have $\mathtt{in}(c,M) \in \widetilde{\mathsf{tr}''}$ and $c \in \mathcal{C}h_a$, then we know for sure that the input was done by a process colored by a in D'_1 and in D_1.

Hence, by construction of $\widetilde{\mathsf{tr}'}$ and $\widetilde{\mathsf{tr}'''}$, we have in fact that $\widetilde{\mathsf{tr}'} = \widetilde{\mathsf{tr}'''}$ and so $(\widetilde{\mathsf{tr}'}, \mathtt{new}\ \mathcal{E}.\Phi'_S) \in \mathsf{trace}(S'_1)$. Thanks to Corollary 6, we can also deduce that $\mathtt{new}\mathcal{E}.\Phi_S \sim \mathtt{new}\mathcal{E}.\Phi'_S$.

But $\Phi_S = \Phi \uplus \Phi_+$ and since $\mathrm{dom}(\Phi_S) = \mathrm{dom}(\Phi'_S)$, there exists Φ' and Φ'_+ such that $\mathrm{dom}(\Phi') = \mathrm{dom}(\Phi)$, $\mathrm{dom}(\Phi_+) = \mathrm{dom}(\Phi'_+)$ and $\Phi'_S = \Phi' \uplus \Phi'_+$. Since the transformation between $\widetilde{\mathsf{tr}'}$ and $\widetilde{\mathsf{tr}''}$ only modifies the terms of the trace and not the actions themselves, and since we assume that there is no τ action between $\mathtt{new}\ w.\mathtt{out}(c_a,w)$ and $\mathtt{in}(c_b,w)$ in $\underline{\mathsf{tr}''}$, for any $w \in \mathrm{dom}(\Phi'_+)$, a simple induction on $|\mathrm{dom}(\Phi'_+)|$ allows us to show that $(\mathsf{tr}, \mathtt{new}\mathcal{E}.\Phi') \in \mathsf{trace}(S'_1) \subseteq \mathsf{trace}(S')$. Lastly, since we have that $\mathtt{new}\mathcal{E}.(\Phi\uplus\Phi_+) \sim \mathtt{new}\mathcal{E}.(\Phi'\uplus\Phi'_+)$, we can deduce that $\mathtt{new}\mathcal{E}.\Phi \sim \mathtt{new}\mathcal{E}.\Phi'$. Hence the result holds. ■