Composition of Password-based Protocols

Céline Chevalier, Stéphanie Delaune, Steve Kremer, Mark D. Ryan

Noname manuscript No.
(will be inserted by the editor)

Composition of Password-based Protocols

Céline Chevalier · Stéphanie Delaune · Steve Kremer · Mark D. Ryan

Received: date / Accepted: date

Abstract Formal and symbolic techniques are extremely useful for modelling and analysing security protocols. They have helped to improve our understanding of such protocols, allowed us to discover flaws, and they also provide support for protocol design. However, such analyses usually consider that the protocol is executed in isolation or assume a bounded number of protocol sessions. Hence, no security guarantee is provided when the protocol is executed in a more complex environment.

In this paper, we study whether password protocols can be safely composed, even when a same password is reused. More precisely, we present a transformation which maps a password protocol that is secure for a single protocol session (a decidable problem) to a protocol that is secure for an unbounded number of sessions. Our result provides an effective strategy to design secure password protocols: (i) design a protocol intended to be secure for one protocol session; (ii) apply our transformation and obtain a protocol which is secure for an unbounded number

This work has been partially supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n^o 258865, project ProSecure and the ANR project JCJC VIP n^o 11-JS02-006, EPSRC Leadership Fellowship *Analysing Security and Privacy Properties* and project *Trust Domains - A Framework for Modelling and Designing E-Service Infrastructures for Controlled Sharing of Information*.

Céline Chevalier
LSV, CNRS & INRIA project Secsi & ENS de Cachan, France
E-mail: celine.chevalier@lsv.ens-cachan.fr

Séphanie Delaune
LSV, CNRS & INRIA project Secsi & ENS de Cachan, France
E-mail: delaune@lsv.ens-cachan.fr

Steve Kremer
LORIA, CNRS & INRIA project Cassis, Nancy, France
E-mail: kremer@inria.fr

Mark D. Ryan
School of Computer Science, University of Birmingham, UK
E-mail: M.D.Ryan@cs.bham.ac.uk

of sessions. Our technique also applies to compose different password protocols allowing us to obtain both inter-protocol and inter-session composition.

1 Introduction

Security protocols are small programs that aim at securing communications over a public network like the Internet. Considering their increasing ubiquity, a high level of assurance is needed in the correctness of such protocols. Developments in formal methods have produced considerable success in analysing security protocols. Automated tools such as Avispa [9] and ProVerif [15] are now capable of analysing large protocols involving several or even an unbounded number of sessions. However, these analyses usually consider that the protocol is executed in isolation, ignoring other protocols that may be executed in parallel.

The assumption that another parallel protocol cannot interfere with the protocol under investigation is valid if the two protocols do not share any secret data (such as cryptographic keys or passwords). This comes from the fact that in models like the applied pi calculus, security properties, even though they are shown in isolation, in fact hold in the presence of an arbitrary (public) environment. This is similar to *universal composition* (UC) [20] in computational models. These arbitrary environments are public, in the sense that they do not have access to the secrets of the protocol under analysis. This is of course necessary as otherwise a completely arbitrary environment could simply output all secret cryptographic key material and trivially break the protocol's security.

While the absence of shared keys between different protocols is obviously desirable, it is not always possible or realistic. For example, *password-based protocols* are those in which a user picks a password which forms one of the secrets used in the protocol. It is unrealistic to assume that users never share the same passwords between different applications. In this paper, we consider the situation in which secret data may be shared between protocols, and we particularly focus on password-based protocols. We investigate under what conditions we can guarantee that such protocols will not interfere with each other. Under certain conditions, we may have that

$$\text{if } P_1 \text{ and } P_2 \text{ are secure then } P_1 \mid P_2 \text{ is secure.}$$

For example, in the context of cryptographic pi calculi (e.g. spi calculus [3], applied pi calculus [2]), "is secure" is often formalised as observational equivalence to some specification. We have that $P_1 \approx S_1$ and $P_2 \approx S_2$ imply $P_1 \mid P_2 \approx S_1 \mid S_2$, where S_1 and S_2 are specifications, and therefore the security of the composition follows from the security of each protocol. Here, the composition of security relies on two facts. First, as mentioned, security means observational equivalence to a specification; the attacker is an *arbitrary context*, and $P_i \approx S_i$ means P_1 and S_1 are equivalent in any environment. Second, by forming the composition $P_1 \mid P_2$ we have made the assumption that P_1 and P_2 do not share any secret.

Now suppose that P_1 and P_2 do share a secret w. To prove that their security composes, one would like to show that

$$\text{if } \nu w.P_1 \text{ and } \nu w.P_2 \text{ are secure then } \nu w.(P_1 \mid P_2) \text{ is secure.}$$

Note in particular that $\nu w.(P_1 \mid P_2)$ is different from $(\nu w.P_1) \mid (\nu w.P_2)$ because the latter refers to two different secrets, since the two ν have different scopes. In contrast with the previously mentioned composition result, this one does not hold in general.

Password-based cryptographic protocols are a prominent means to achieve authentication or to establish authenticated, shared session keys, *e.g.* EKE [14], SPEKE [32], the KOY protocol [33], or J-PAKE [30]. The advantage of such schemes is that they do not rely on a key infrastructure but only on a shared password, which is often human chosen or at least human memorable. The J-PAKE protocol has for instance been used to secure Firefox Sync, a browser synchronization tool which allows to synchronize data such as preferences and bookmarks among different computers [31]. The Trusted Platform Module (TPM) [38] also relies on passwords, called authdata, for authentication. However, such passwords are generally *weak* and may be subject to dictionary attacks (also called guessing attacks). In an *online* dictionary attack an adversary tries to execute the protocol for each possible password. While online attacks are difficult to avoid they can be made impracticable by limiting the number of password trials or adding a time-out of few seconds after a wrong password. In an *offline* guessing attack an adversary interacts with one or more sessions in a first phase. In a second, offline phase the attacker uses the collected data to verify each potential password. In this paper we concentrate on offline guessing attacks.

Several attempts have been made, based on the initial work of Lowe [34], to characterize guessing attacks [22, 24, 28]. In [23], Corin *et al.* proposed an elegant definition of resistance to passive guessing attacks, based on static equivalence in the applied pi calculus. A similar definition has also been used by Baudet [12] who uses constraint solving techniques to decide resistance against guessing attacks for an active attacker and a bounded number of sessions. Recent versions of the ProVerif tool also aim at proving resistance against guessing attacks for an active attacker and an unbounded number of sessions (at the price of being incomplete and not guaranteeing termination) [16]. Moreover, Abadi *et al.* further increase the confidence in this definition by showing its computational soundness for a given equational theory in the case of a passive attacker [1].

Our contributions. In this paper, we study whether resistance against guessing attacks composes when the same password is used for different protocols. Protocols are modelled in a cryptographic process calculus inspired by the applied pi calculus. We use the definition introduced by Corin *et al.* (see [23]). This allows us to provide results for protocols involving a variety of cryptographic primitives represented by means of an arbitrary equational theory. First we show that in the case of a passive attacker, resistance against guessing attacks composes (Section 5).

In the case of an active attacker we prove that as expected, resistance against guessing attacks does compose when no secrets are shared. However, resistance against active guessing attacks does *not* compose in general when the same password is shared between different protocols. In this paper we propose a simple protocol transformation which ensures that a same password can safely be shared between different protocols. More precisely, our results can be summarized as follows. We use a safe transformation which replaces a weak password w by $\mathsf{h}(t, w)$ where t is some *tag* and h a hash function. Then, we show how to use this tagging

technique to compose different protocols. Consider n password protocols such that each protocol resists separately against guessing attacks on w. When we instantiate the tag t to a unique protocol identifier *pid*, one for each of the n protocols, we show that the parallel composition of these tagged protocols resists against guessing attacks on w, where w is the password shared by each of these protocols. Next we show how to dynamically establish a session identifier *sid*. Instantiating the tag t by this session identifier allows us to compose different sessions of a same protocol. Hence it is sufficient to prove resistance against guessing attacks on a single session of a protocol to conclude that the transformed protocol resists against guessing attacks for an unbounded number of sessions. These techniques can also be combined into a tag which consists of both the protocol and session identifier obtaining both inter-protocol and inter-session composition.

One may note that resistance against guessing attack is generally not the main goal of a protocol, which may be authentication or key exchange. Therefore we additionally show that secrecy and authentication properties are also preserved when composing transformed protocols.

Related work. The problem of secure composition has been approached by several authors. Datta *et al.* provide a general strategy [27] whereas our composition result identifies a specific class of protocols that can be composed. In [29, 25], some criteria are given to ensure that parallel composition is safe. Andova *et al.* provide conditions to allow a broader class of composition operations [6].

However, none of these works deal with composing resistance against guessing attacks. They consider secrecy in term of deducibility or authentication properties. To the best of our knowledge only Malladi *et al.* [35] have studied composition w.r.t. guessing attacks. They point out vulnerabilities that arise when the same password is used for different applications and develop a method to derive conditions that a protocol has to follow in order to be resistant against guessing attacks. However, applying their methods to particular protocols is not always straightforward. Moreover, their work relies on the definition of guessing attack due to Lowe [34] which relies on a particular set of cryptographic primitives. Our results are general and independent of the underlying equational theory.

In computational models, Boyko *et al.* [19] presented a security model for password-based key-exchange based on simulation proofs, ensuring security in case of composition. A more generic solution was proposed by Canetti *et al.* [21] who propose a protocol based on KOY, which is secure in the UC model [20]. This work has been extended to active adversaries [5], group key exchange [4] and to define distributed public-key cryptography from passwords in *e.g.* [18]. A main difference between works in the UC model and our work (besides the obvious differences between symbolic and computational models) is that in the UC model designers generally apply an "ad-hoc recipe" (often using "magical" session identifiers given by the framework) and show that one session of a protocol fulfills the given requirements. The UC theorem then ensures composition, *i.e.*, composition follows from the strong security definition which has to be proven. In our work we make explicit the construction of session identifiers in our transformation and prove that a generic protocol transformation can be used to achieve composition. Note, however, that despite this difference, both approaches share many essential ideas.

Finally, we may note that *tagging* is a well known technique. We have already mentioned its use to achieve some forms of composition [7, 26]. Other forms of tagging were used to ensure termination of a verification procedure [17], safely bound the length of messages [8] or obtain decidability for the verification of some classes of protocols [37].

— PART I: Models —

In this part, we introduce the model that will be used throughout the paper. After some preliminaries (see Section 2), we describe the model of protocols in Section 3. In Section 4, we define the security properties for which we will study composition in Part II.

2 Preliminaries

2.1 Messages

A protocol consists of some agents communicating on a network. The messages sent by the agents are formed from data that the agents hold, as well as cryptographic keys and messages that the agent has previously received. We assume an infinite set of *names* $\mathcal{N}$, for representing keys, data values, nonces, and names of agents, and we assume a *signature* Σ, i.e. a finite set of *function symbols* such as senc and sdec, each with an arity. Messages are abstracted by *terms*, and cryptographic operations are represented by function symbols. Given a signature Σ and an infinite set of variables $\mathcal{X}$, we denote by $\mathcal{T}(\Sigma)$ (resp. $\mathcal{T}(\Sigma, \mathcal{X})$) the set of *terms* over $\Sigma \cup \mathcal{N}$ (resp. $\Sigma \cup \mathcal{N} \cup \mathcal{X}$). The former is called the set of *ground terms* over Σ, while the latter is simply called the set of terms over Σ. We write $fn(M)$ (resp. $fv(M)$) for the set of names (resp. variables) that occur in the term M. A *substitution* σ is a mapping from a finite subset of $\mathcal{X}$ called its *domain* and written $\mathrm{dom}(\sigma)$ to $\mathcal{T}(\Sigma, \mathcal{X})$. Substitutions are extended to endomorphisms of $\mathcal{T}(\Sigma, \mathcal{X})$ as usual. We use a postfix notation for their application. Similarly, we allow replacement of names: the term $M\{^N/_n\}$ is the term obtained from M after replacing any occurrence of the name n by the term N.

As in the applied pi calculus [2], we use *equational theories* for modelling the algebraic properties of the cryptographic primitives. An equational theory is defined by a finite set E of equations $U = V$ with $U, V \in \mathcal{T}(\Sigma, \mathcal{X})$ and U, V without names. We define $=_{\mathsf{E}}$ to be the smallest equivalence relation on terms, that contains E and that is closed under application of contexts and substitutions of terms for variables. Since the equations in E do not contain any names, we have that $=_{\mathsf{E}}$ is also closed by substitutions of terms for names.

Example 1 Consider the signature $\Sigma_{\mathsf{enc}} = \{\mathsf{sdec}, \mathsf{senc}, \mathsf{adec}, \mathsf{aenc}, \mathsf{pk}, \langle\,\rangle, \mathsf{proj}_1, \mathsf{proj}_2, \mathsf{f}\}$. The symbols sdec, senc, adec, aenc, and $\langle\,\rangle$ are functional symbols of arity 2 that represent respectively the symmetric and asymmetric decryption and encryption as well as pairing functions whereas pk, proj_1, proj_2, and f are functional symbols of arity 1 that represent public key and projection functions on respectively the first and the second component of a pair. The function symbol f represents any operation, e.g. a hash function. A typical example of an equational theory useful for cryptographic protocols is $\mathsf{E}_{\mathsf{enc}}$, defined by the following equations:

$$\mathsf{sdec}(\mathsf{senc}(x,y),y) = x \qquad \mathsf{adec}(\mathsf{aenc}(x,\mathsf{pk}(y)),y) = x$$
$$\mathsf{senc}(\mathsf{sdec}(x,y),y) = x \qquad \mathsf{proj}_i(\langle x_1, x_2\rangle) = x_i \qquad (i \in \{1,2\})$$

Let $M = \mathsf{senc}(\mathsf{f}(\mathsf{sdec}(\mathsf{senc}(n,w),w)),w)$ and $M' = \mathsf{senc}(\mathsf{f}(n),w)$. In this theory, we have that the terms M and M' are equal modulo $\mathsf{E}_{\mathsf{enc}}$, written $M =_{\mathsf{E}_{\mathsf{enc}}} M'$, while obviously the syntactic equality $M = M'$ does not hold.

2.2 Assembling Terms into Frames

At some moment, while engaging in one or more sessions of one or more protocols, an attacker may have observed a sequence of messages $M_1, \ldots, M_\ell$. We want to represent this knowledge of the attacker. It is not enough for us to say that the attacker knows the *set* of terms $\{M_1, \ldots, M_\ell\}$, since he also knows the order that he observed them in. Furthermore, we should distinguish those names that the attacker knows from those that were freshly generated by others and which remain secret from the attacker; both kinds of names may appear in the terms. Therefore, we use the concept of *frame* from the applied pi calculus [2] to represent the knowledge of the attacker.

Definition 1 (frame) A *frame* $\phi = \nu\tilde{n}.\sigma$ consists of a finite set $\tilde{n} \subseteq \mathcal{N}$ of *restricted* names (those that the attacker does not know), and a substitution σ of the form $\{{}^{M_1}/_{x_1}, \ldots, {}^{M_\ell}/_{x_\ell}\}$ where: $x_1, \ldots, x_\ell$ are distinct variables, and $M_1, \ldots, M_\ell$ are ground terms.

The names $\tilde{n}$ are bound and can be renamed. We denote by $=_\alpha$ the α-renaming relation on frames. The *domain* of the frame ϕ, written $\mathrm{dom}(\phi)$, is defined as $\{x_1, \ldots, x_\ell\}$.

Example 2 We use the equational theory $\mathsf{E}_{\mathsf{enc}}$ presented in Example 1. The frame below represents the sequence of messages that are sent on the network during an honest execution of the EKE protocol [14].

$$\phi_{\mathsf{EKE}} = \nu w, k, r, na, nb.$$
$$\{{}^{\mathsf{senc}(\mathsf{pk}(k),w)}/_{x_1}, {}^{\mathsf{senc}(\mathsf{aenc}(r,\mathsf{pk}(k)),w)}/_{x_2}, {}^{\mathsf{senc}(na,r)}/_{x_3}, {}^{\mathsf{senc}(\langle na,nb\rangle,r)}/_{x_4}, {}^{\mathsf{senc}(nb,r)}/_{x_5}\}$$

A description of the protocol will be given in Example 8.

2.3 Deduction

Given a frame ϕ that represents the information available to an attacker, we may ask whether a given ground term M may be deduced from ϕ. Given an equational theory E on Σ, this relation is written $\phi \vdash_{\mathsf{E}} M$ and is formally defined below.

Definition 2 (deduction) Let M be a ground term and $\nu\tilde{n}.\sigma$ be a frame. We have that $\nu\tilde{n}.\sigma \vdash_{\mathsf{E}} M$ if and only if there exists a term $N \in \mathcal{T}(\Sigma, \mathcal{X})$ such that $fn(N) \cap \tilde{n} = \emptyset$ and $N\sigma =_{\mathsf{E}} M$. Such a term N is a *recipe* of the term M.

Intuitively, the deducible messages are the messages of ϕ and the names that are not protected in ϕ, closed by equality in E and closed by application of function symbols. When $\nu\tilde{n}.\sigma \vdash_{\mathsf{E}} M$, any occurrence of names from $\tilde{n}$ in M is bound by $\nu\tilde{n}$. So $\nu\tilde{n}.\sigma \vdash_{\mathsf{E}} M$ could be formally written $\nu\tilde{n}.(\sigma \vdash_{\mathsf{E}} M)$.

Example 3 Consider the theory $\mathsf{E}_{\mathsf{enc}}$ and the frame $\phi = \nu k, s_1.\{{}^{\mathsf{senc}(\langle s_1,s_2\rangle,k)}/_{x_1}, {}^{k}/_{x_2}\}$. We have that $\phi \vdash_{\mathsf{E}_{\mathsf{enc}}} k$, $\phi \vdash_{\mathsf{E}_{\mathsf{enc}}} s_1$ and $\phi \vdash_{\mathsf{E}_{\mathsf{enc}}} s_2$. Indeed x_2, $\mathsf{proj}_1(\mathsf{sdec}(x_1, x_2))$ and s_2 are recipes of the terms k, s_1 and s_2 respectively.

2.4 Static Equivalence

The frames we have introduced are a bit too fine-grained as representations of the attacker's knowledge. For example, $\nu k.\{^{\mathsf{senc}(s_0,k)}/_x\}$ and $\nu k.\{^{\mathsf{senc}(s_1,k)}/_x\}$ represent a situation in which the encryption of the public name s_0 (resp. s_1) by a randomly-chosen key has been observed. Since the attacker cannot detect the difference between these situations, the frames should be considered equivalent. To formalise this, we note that if two recipes M, N on the frame ϕ produce the same term, we say they are equal in the frame, and write $(M =_{\mathsf{E}} N)\phi$. If two frames have identical distinguishing power, then we say that they are *statically equivalent*. Formally:

Definition 3 (static equivalence [2]) We say that two terms M and N are *equal in the frame* ϕ, and write $(M =_{\mathsf{E}} N)\phi$, if there exists $\tilde{n}$ and a substitution σ such that $\phi =_\alpha \nu\tilde{n}.\sigma$, $M\sigma =_{\mathsf{E}} N\sigma$, and $\tilde{n} \cap (fn(M) \cup fn(N)) = \emptyset$. We say that two frames ϕ_1 and ϕ_2 are *statically equivalent*, $\phi_1 \approx_{\mathsf{E}} \phi_2$, when:

- $\mathrm{dom}(\phi_1) = \mathrm{dom}(\phi_2)$, and
- for all terms M, N we have that $(M =_{\mathsf{E}} N)\phi_1$ if and only if $(M =_{\mathsf{E}} N)\phi_2$.

Note that by definition of $\approx$, we have that $\phi_1 \approx \phi_2$ when $\phi_1 =_\alpha \phi_2$ and we have also that $\nu n.\phi \approx \phi$ when n does not occur in ϕ.

Example 4 Consider again the equational theory $\mathsf{E}_{\mathsf{enc}}$ provided in Example 1 and the frames:

$$\phi = \nu k.\{^{\mathsf{senc}(s_0,k)}/_{x_1}, {}^{k}/_{x_2}\}, \text{ and } \phi' = \nu k.\{^{\mathsf{senc}(s_1,k)}/_{x_1}, {}^{k}/_{x_2}\}.$$

Intuitively, s_0 and s_1 could be the two possible (public) values of a vote. We have $(\mathsf{sdec}(x_1, x_2) =_{\mathsf{E}_{\mathsf{enc}}} s_0)\phi$ whereas $(\mathsf{sdec}(x_1, x_2) \neq_{\mathsf{E}_{\mathsf{enc}}} s_0)\phi'$. Therefore we have that $\phi \not\approx \phi'$. However, we have that

$$\nu k.\{^{\mathsf{senc}(s_0,k)}/_{x_1}\} \approx \nu k.\{^{\mathsf{senc}(s_1,k)}/_{x_1}\}.$$

The following lemma is a consequence of some lemmas stated in [2] and will be useful later on to establish our composition result.

Lemma 1 *Let $\phi_1 = \nu\tilde{n}_1.\sigma_1$ and $\phi_2 = \nu\tilde{n}_2.\sigma_2$ be two frames such that $\phi_1 \approx \phi_2$.*

1. *$\nu n.\phi_1 \approx \nu n.\phi_2$ when $n \notin \tilde{n}_1 \cup \tilde{n}_2$,*
2. *$\phi_1\{^s/_n\} \approx \phi_2\{^s/_n\}$ when $n \notin \tilde{n}_1 \cup \tilde{n}_2$ and s is a fresh name.*

3 Protocols

We now define our cryptographic process calculus for describing protocols. This calculus is inspired by the applied pi calculus [2] but we prefer a simplified version which is sufficient for the purpose of this paper. In particular we only consider one channel, which is public (i.e. under the control of the attacker). Moreover, we only consider *closed* processes: all variables appearing in terms are under the scope of an input. Finally, we only consider finite processes, i.e. processes without replication. As we will argue in Section 8 this is not a restriction and our composition results carry over to an unbounded number of sessions.

3.1 Protocol Language

The grammar for *processes* is given below. One has *plain processes* P, Q, R and *extended processes* A, B, C. Plain processes built up in a similar way to processes in the pi calculus except that messages can contain terms rather than just names. Furthermore, we enrich plain processes with events. Events are function symbols ev, ev_1, ... with a given arity which do not appear in Σ. They are used to *annotate* processes and are useful when formalizing some security properties such as authentication. Plain processes are formed from the grammar

$P, Q, R :=$	plain processes	
	0	null process
	$P \mid Q$	parallel composition
	$\text{in}(x).P$	message input
	$\text{out}(M).P$	message output
	if $M = N$ then P else Q	conditional
	$\text{ev}(\tilde{M}).P$	event

such that a variable x appears in a term only if the term is in the scope of an input $\text{in}(x)$. The null process 0 does nothing; $P \mid Q$ is the parallel composition of P and Q. The conditional *if* $M = N$ *then* P *else* Q is standard, but $M = N$ represents equality modulo the underlying equational theory E. We omit *else* Q when Q is 0. The process $\text{in}(x).P$ is ready to input on the public channel, then to run P with the actual message instead of x, while $\text{out}(M).P$ is ready to output M, then to run P. Again, we omit P when P is 0. Given a set of names $\tilde{n} = \{n_1, \ldots, n_p\}$, we also write $\nu\tilde{n}.A$ instead of $\nu n_1 \ldots \nu n_p.A$.

Further, we extend processes with active substitutions and restrictions:

$$A, B, C := P \;\mid\; A \mid B \;\mid\; \nu n.A \;\mid\; \{^M/_x\}$$

where M is a ground term. As usual, names and variables have scopes, which are delimited by restrictions and by inputs. We write $fv(A)$, $bv(A)$, $fn(A)$, $bn(A)$ for the sets of free and bound variables (resp. names). Moreover, we require processes to be *name and variable distinct*, meaning that $bn(A) \cap fn(A) = \emptyset$, $bv(A) \cap fv(A) = \emptyset$, and also that any name and variable is bound at most once in A. Note that the only free variables are introduced by active substitutions (the x in $\{^M/_x\}$). Lastly, in an extended process, we require that there is at most one substitution for each variable. An *instance* of an extended process is a process obtained by a bijective renaming of its bound names and variables. We observe that given processes A and B, there always exist instances A' and B' of A, respectively B, such that the process $A' \mid B'$ will respect the disjointness conditions on names and variables.

We also extend replacements of names $\{^M/_n\}$ from terms to processes when the names $fn(M) \cup \{n\}$ are not bound by the process. An *evaluation context* is an extended process with a hole instead of an extended process. Extended processes built up from the null process, active substitutions using parallel composition and restriction are called *extended frames* (extending the notion of frame introduced in

Section 2.2[1]). Given an extended process A we denote by $\phi(A)$ the extended frame obtained by replacing any embedded plain processes in it with 0.

Example 5 Consider the following process:

$$A = \nu s, k_1.(\mathrm{out}(a) \mid \{^{\mathsf{senc}(s,k_1)}/_x\} \mid \nu k_2.\mathrm{out}(\mathsf{senc}(s,k_2))).$$

We have that $\phi(A) = \nu s, k_1.(0 \mid \{^{\mathsf{senc}(s,k_1)}/_x\} \mid \nu k_2.0)$.

3.2 Semantics

The semantics of our calculus is defined by two relations: *structural equivalence*, denoted $\equiv$, and *reduction*, denoted $\xrightarrow{\ell}$.

Structural equivalence. We consider a basic structural equivalence, i.e. the smallest equivalence relation closed by application of evaluation contexts and such that

Par-0	$A \mid 0 \equiv A$
Par-C	$A \mid B \equiv B \mid A$
Par-A	$(A \mid B) \mid C \equiv A \mid (B \mid C)$
New-Par	$A \mid \nu n.B \equiv \nu n.(A \mid B) \quad n \notin fn(A)$
New-C	$\nu n_1.\nu n_2.A \equiv \nu n_2.\nu n_1.A$

Using structural equivalence, every extended process A can be rewritten to consist of a substitution and a plain process with some restricted names, i.e.

$$A \equiv \nu \tilde{n}.(\{^{M_1}/_{x_1}\} \mid \ldots \mid \{^{M_k}/_{x_k}\} \mid P).$$

In particular, using structural equivalence, any frame can be rewritten as $\nu\tilde{n}.\sigma$ matching the notion of frame introduced in Section 2.2. Since we require our processes to be name and variable distinct, and extended processes have at most one active substitution for each variable, α-renaming is not needed to rewrite extended frame into frame.

Note that static equivalence on frames coincides with [2] (even though our process calculus is different). We note that unlike in the original applied pi calculus, active substitutions cannot "interact" with the extended processes. As we will see in the following active substitutions record the outputs of a process to the environment. The notion of frames will be particularly useful to define resistance against guessing attacks.

Example 6 Note that in Example 5, we have that $\phi(A) \equiv \nu s, k_1, k_2.\{^{\mathsf{senc}(s,k_1)}/_x\}$.

We have the following useful lemma which comes from [2].

Lemma 2 *Let $\phi_1 = \nu\tilde{n}_1.\sigma_1$ and $\phi_2 = \nu\tilde{n}_2.\sigma_2$ be two frames. Let $s \notin \tilde{n}_1 \cup \tilde{n}_2$.*

1. *$\nu s.\nu\tilde{n}_1.(\sigma_1 \mid \{^s/_x\}) \approx \nu s.\nu\tilde{n}_2.(\sigma_2 \mid \{^s/_x\})$ if and only if $\phi_1 \approx \phi_2$;*
2. *Let ϕ be another frame such that $\phi_1 \mid \phi$ and $\phi_2 \mid \phi$ are frames. If $\phi_1 \approx \phi_2$, then $\phi_1 \mid \phi \approx \phi_2 \mid \phi$.*

[1] More precisely, the notion of frame introduced in Definition 1 requires the restricted names to be written at the beginning of the frame, whereas this is not the case in an extended frame. But we show in Section 3.2 that using structural equivalence, any frame can be rewritten with all the restricted names at the beginning.

Operational semantics. We now define the semantics of our calculus. The labelled semantics defines a relation $A \xrightarrow{\ell} A'$ where ℓ is a label of one of the following forms:

- a label $\text{in}(M)$, where M is a ground term such that $\phi(A) \vdash_{\mathsf{E}} M$. This corresponds to an input of M;
- a label $\text{out}(M)$, where M is a ground term, which corresponds to an output of M;
- a label $\text{ev}(\tilde{M})$, where $\tilde{M}$ is a sequence of ground terms. This label is used to memorize that a specific event has been executed;
- a label τ corresponding to a silent action.

Labelled operational semantics ($\xrightarrow{\ell}$) is the smallest relation between extended processes which is closed under structural equivalence ($\equiv$) and such that

In	$\text{in}(x).P \xrightarrow{in(M)} P\{^{M}/_{x}\}$	
Out	$\text{out}(M).P \xrightarrow{out(M)} P \mid \{^{M}/_{x}\}$	where x is a fresh variable
Then	if $M = N$ then P else $Q \xrightarrow{\tau} P$ where $M =_{\mathsf{E}} N$	
Else	if $M = N$ then P else $Q \xrightarrow{\tau} Q$ where $M \neq_{\mathsf{E}} N$	
Event	$\text{ev}(\tilde{M}).P \xrightarrow{ev(\tilde{M})} P$	
Context	$\dfrac{A \xrightarrow{\ell} B}{C[A] \xrightarrow{\ell} C[B]}$	where C is an evaluation context, and if $\ell = in(M)$ then $\phi(C[A]) \vdash_{\mathsf{E}} M$

These rules use standard ideas known from pi calculus derivatives. Note that the $in(M)$ label has as parameter the closed term being input, unlike in the applied pi calculus where the input term may contain variables. The side condition on Context ensures that the environment can deduce the input message M even though the context may restrict some names in M. The output of a message M adds an active substitution. Note that an output M may contain restricted names without revealing these names. Further we note that events do not influence the execution of processes and are merely annotations used to record that a process reached a given point. As explained previously, some of the design choices of the semantics differ slightly from the applied pi calculus. Our choices allow us to consider a very simple structural equivalence and avoid unnecessary complications in the proofs of our main results. We denote by $\rightarrow$ the relation $\left\{\xrightarrow{\ell} \mid \ell \in \{in(M), out(M), ev(\tilde{M}), \tau\},\ M, \tilde{M} \text{ ground terms}\right\}$ and by $\rightarrow^*$ its reflexive and transitive closure.

Example 7 We illustrate our syntax and semantics with the well-known handshake protocol.

$$\begin{aligned} \text{A} &\rightarrow \text{B} : \mathsf{senc}(n, w) \\ \text{B} &\rightarrow \text{A} : \mathsf{senc}(\mathsf{f}(n), w) \end{aligned}$$

The goal of this protocol is to authenticate B from A's point of view, provided that they share an initial secret w. This is done by a simple challenge-response

transaction: A sends a random number (a *nonce*) encrypted with the shared secret key w. Then, B decrypts this message, applies a given function (for instance $\mathsf{f}(n) = n + 1$) to it, and sends the result back, also encrypted with w. Finally, the agent A checks the validity of the result by decrypting the message and checking the decryption against $\mathsf{f}(n)$. In our calculus, we model the protocol as $\nu w.(A \mid B)$ where

- $A = \nu n.\mathrm{out}(\mathsf{senc}(n, w)).\ \mathrm{in}(x).\ \mathrm{if}\ \mathsf{sdec}(x, w) = \mathsf{f}(n)\ \mathrm{then}\ P$
- $B = \mathrm{in}(y).\ \mathrm{out}(\mathsf{senc}(\mathsf{f}(\mathsf{sdec}(y, w)), w))$.

where P models an application that is executed when B has been successfully authenticated. The derivation described below represents a normal execution of the protocol. For simplicity of this example we suppose that $x \notin fv(P)$.

$$
\begin{array}{rl}
 & \nu w.(A \mid B) \\
\xrightarrow{out(\mathsf{senc}(n,w))} & \nu w.\nu n.(B \mid \{^{\mathsf{senc}(n,w)}/_{x_1}\} \mid \mathrm{in}(x).\ \mathrm{if}\ \mathsf{sdec}(x, w) = \mathsf{f}(n)\ \mathrm{then}\ P) \\
\xrightarrow{in(\mathsf{senc}(n,w))} & \nu w.\nu n.(\mathrm{out}(M) \mid \{^{\mathsf{senc}(n,w)}/_{x_1}\} \mid \mathrm{in}(x).\ \mathrm{if}\ \mathsf{sdec}(x, w) = \mathsf{f}(n)\ \mathrm{then}\ P) \\
\xrightarrow{out(M)} & \nu w.\nu n.(\{^{\mathsf{senc}(n,w)}/_{x_1}\} \mid \{^{M}/_{x_2}\} \mid \mathrm{in}(x).\ \mathrm{if}\ \mathsf{sdec}(x, w) = \mathsf{f}(n)\ \mathrm{then}\ P) \\
\xrightarrow{in(\mathsf{senc}(\mathsf{f}(n),w))} & \nu w.\nu n.(\{^{\mathsf{senc}(n,w)}/_{x_1}\} \mid \{^{M}/_{x_2}\} \mid \ \ \mathrm{if}\ \mathsf{sdec}(\mathsf{senc}(\mathsf{f}(n), w), w) = \mathsf{f}(n)\ \mathrm{then}\ P) \\
\xrightarrow{\tau} & \nu w.\nu n.(\{^{\mathsf{senc}(n,w)}/_{x_1}\} \mid \{^{M}/_{x_2}\} \mid P)
\end{array}
$$

where $M = \mathsf{senc}(\mathsf{f}(\mathsf{sdec}(\mathsf{senc}(n, w), w)), w) =_{\mathsf{E_{enc}}} \mathsf{senc}(\mathsf{f}(n), w)$.

3.3 Password-Based Protocols

In the remaining, we will focus our attention on password-based protocols.

Definition 4 (ℓ-party password protocol specification) An *ℓ-party password protocol specification* $\mathcal{P}$ is a process such that:

$$\mathcal{P} = \nu w.(\nu \tilde{m}_1.P_1 \mid \ldots \mid \nu \tilde{m}_\ell.P_\ell)$$

where each P_i is a closed plain processes. The processes $\nu \tilde{m}_i.P_i$ are called the roles of $\mathcal{P}$.

Example 8 The EKE protocol [14] is a 2-party password protocol that can be informally described by the following 5 steps.

$$
\begin{array}{lll}
\mathrm{A} \to \mathrm{B} : & \mathsf{senc}(\mathsf{pk}(k), w) & \text{(EKE.1)} \\
\mathrm{B} \to \mathrm{A} & \mathsf{senc}(\mathsf{aenc}(r, \mathsf{pk}(k)), w) & \text{(EKE.2)} \\
\mathrm{A} \to \mathrm{B} & \mathsf{senc}(na, r) & \text{(EKE.3)} \\
\mathrm{B} \to \mathrm{A} & \mathsf{senc}(\langle na, nb \rangle, r) & \text{(EKE.4)} \\
\mathrm{A} \to \mathrm{B} & \mathsf{senc}(nb, r) & \text{(EKE.5)}
\end{array}
$$

In the first step (EKE.1) A generates a new private key k and sends the corresponding public key $\mathsf{pk}(k)$ to B, encrypted (using symmetric encryption) with the shared password w. Then, B generates a fresh session key r, which he encrypts (using asymmetric encryption) with the previously received public key $\mathsf{pk}(k)$. Finally, he encrypts the resulting ciphertext with the password w and sends the result to A

(EKE.2). The last three steps (EKE.3-5) perform a handshake to avoid replay attacks. One may note that this is a password-only protocol. A new private and public key are used for each session and the only shared secret between different sessions is the password w.

A formal description of this protocol in our calculus is given below. We use the equational theory $\mathsf{E}_{\mathsf{enc}}$ presented in Example 1 to model this protocol.

$$
\begin{array}{ll}
A = \nu k, na.\ \mathrm{ev}_{begin}(w, na, \mathsf{pk}(k)). & B = \nu r, nb. \\
\quad \mathrm{out}(\mathsf{senc}(\mathsf{pk}(k), w)). & \quad \mathrm{in}(y_1). \\
\quad \mathrm{in}(x_1). & \quad \mathrm{out}(\mathsf{senc}(\mathsf{aenc}(r, \mathsf{sdec}(y_1, w)), w)). \\
\quad \text{let } ra = \mathsf{adec}(\mathsf{sdec}(x_1, w), k). & \quad \mathrm{in}(y_2). \\
\quad \mathrm{out}(\mathsf{senc}(na, ra)) & \quad \mathrm{out}(\mathsf{senc}(\langle \mathsf{sdec}(y_2, r), nb \rangle, r)). \\
\quad \mathrm{in}(x_2). & \quad \mathrm{in}(y_3) \\
\quad \text{if } \mathsf{proj}_1(\mathsf{sdec}(x_2, ra)) = na \text{ then} & \quad \text{if } \mathsf{sdec}(y_3, r) = nb \text{ then} \\
\quad \mathrm{out}(\mathsf{sdec}(\mathsf{proj}_2(\mathsf{sdec}(x_2, ra)), ra)).0 & \quad \mathrm{ev}_{end}(w, \mathsf{sdec}(y_2, r), \mathsf{sdec}(y_1, w)).0
\end{array}
$$

We use the construction let $x = M$ to enhance readability. The semantics of this construction is to simply replace x by M in the remaining of the process. The process also includes two events which we will explain below when discussing correspondence properties. An honest execution of this protocol in the presence of a passive attacker yields the frame $\nu w.\phi_{\mathsf{EKE}}$ where:

$$
\begin{array}{l}
\phi_{\mathsf{EKE}} = \nu k, r, na, nb. \\
\{^{\mathsf{senc}(\mathsf{pk}(k),w)}/_{x_1}, ^{\mathsf{senc}(\mathsf{aenc}(r,\mathsf{pk}(k)),w)}/_{x_2}, ^{\mathsf{senc}(na,r)}/_{x_3}, ^{\mathsf{senc}(\langle na,nb\rangle,r)}/_{x_4}, ^{\mathsf{senc}(nb,r)}/_{x_5}\}.
\end{array}
$$

4 Security Properties

In this section, we define the security properties for which we will study composition in Part II. In particular, we consider secrecy (as a reachability property) and correspondence properties that allow one to express different forms of authentication among them aliveness, injective and non-injective agreements. In Section 4.3, we review resistance against guessing attacks, a security property that is particularly relevant when studying password based protocols.

4.1 Secrecy

In formal models secrecy of a term t is usually modelled as the attacker's inability to deduce the term t.

Definition 5 (secrecy) An extended process A preserves secrecy of a closed term t, written $A \not\vdash_{\mathsf{E}} t$, if for every extended process B such that $A \rightarrow^* B$ we have that $\phi(B) \not\vdash t$.

4.2 Correspondence Properties

Properties such as authentication are classically modeled as *correspondance properties.*

$$\begin{array}{lll} \Phi, \Phi_1 \ldots := & \text{correspondance properties} & \\ & \mathrm{ev}(x_1, \ldots, x_k) \Rightarrow \mathrm{ev}'(x_1, \ldots, x_k) & \text{simple correspondance} \\ & \mathrm{ev}(x_1, \ldots, x_k) \Rightarrow_{\mathsf{inj}} \mathrm{ev}'(x_1, \ldots, x_k) & \text{injective correspondance} \end{array}$$

Intuitively, a correspondence property $\mathrm{ev}(x_1, \ldots, x_k) \Rightarrow \mathrm{ev}'(x_1, \ldots, x_k)$ holds if whenever a process executes an event $\mathrm{ev}(u_1, \ldots, u_k)$ then it must have executed the event $\mathrm{ev}'(u_1, \ldots, u_k)$ before. With this simple grammar, we cover various form of authentication among them aliveness, weak agreement, injective and non-injective agreements.

Definition 6 (simple correspondence property) A *simple correspondence property* $ev(x_1, \ldots, x_k) \Rightarrow ev'(x_1, \ldots, x_k)$ *holds on an extended process* A if for every derivation

$$A \xrightarrow{\ell_1} A_1 \xrightarrow{\ell_2} \ldots \xrightarrow{\ell_n} A_n$$

we have that for every $1 \leq i \leq n$ and for every substitution σ if $\ell_i =_{\mathsf{E}} \mathrm{ev}(x_1, \ldots, x_k)\sigma$ then there exists $j < i$ such that $\ell_j =_{\mathsf{E}} \mathrm{ev}'(x_1, \ldots, x_k)\sigma$.

We also consider injective correspondence properties needed to model stronger flavors of authentication, such as injective agreement.

Definition 7 (injective correspondence property) An *injective correspondence property* $ev(x_1, \ldots, x_k) \Rightarrow_{\mathsf{inj}} ev'(x_1, \ldots, x_k)$ *holds on an extended process* A if for every derivation

$$A \xrightarrow{\ell_1} A_1 \xrightarrow{\ell_2} \ldots \xrightarrow{\ell_n} A_n$$

we have that for every $1 \leq i \leq n$ and for every substitution σ,

$$\#\{j \mid \mathrm{ev}(x_1\sigma, \ldots, x_k\sigma) = \ell_j \text{ with } j \leq i\} \leq \#\{j \mid \mathrm{ev}'(x_1\sigma, \ldots, x_k\sigma) = \ell_j \text{ with } j \leq i\}.$$

Example 9 Considering again the EKE protocol described in Example 8 we can model injective agreement using the following correspondance property:

$$\mathrm{ev}_{end}(x, y, z) \Rightarrow \mathrm{ev}_{begin}(x, y, z).$$

This roughly means that the EKE protocol guarantees agreement on the password w, the nonce na, and the public key $\mathsf{pk}(k)$ if whenever an agent executing the role B completes a run of the protocol, apparently with another honest agent executing the role A, then the role A has previously been executed, and the two agents that are involved in this session share the same password w and agree on the data na, and $\mathsf{pk}(k)$ that have been established during the execution of this session.

The idea behind the definition is the following. Suppose the frame ϕ represents the information gained by the attacker by eavesdropping one or more sessions and let w be the password. Then, we can represent resistance against guessing attacks by checking whether the attacker can distinguish a situation in which he guesses the correct password w and a situation in which he guesses an incorrect one, say w'. We model these two situations by adding $\{^{w}/_{x}\}$ (resp. $\{^{w'}/_{x}\}$) to the frame. We use static equivalence to capture the notion of indistinguishability. This definition is due to Baudet [12], inspired from the one by Corin *et al.* [23]. In our definition, we allow multiple shared secrets, and write $\tilde{w}$ for a sequence of such secrets.

Definition 8 (frame resistant to guessing attacks) Let $\phi \equiv \nu\tilde{w}.\phi'$ be a frame. We say that the frame ϕ is *resistant to guessing attacks* against $\tilde{w}$ if

$$\nu\tilde{w}.(\phi' \mid \{^{\tilde{w}}/_{\tilde{x}}\}) \approx \nu\tilde{w}'.\nu\tilde{w}.(\phi' \mid \{^{\tilde{w}'}/_{\tilde{x}}\})$$

where $\tilde{w}'$ is a sequence of fresh names, $\tilde{x}$ a sequence of variables with $\tilde{x} \cap \mathrm{dom}(\phi) = \emptyset$.

Note that this definition is general w.r.t. to the equational theory and the number of guessable data items. Now, we can define what it means for a protocol to be resistant against guessing attacks.

Definition 9 (process resistant to guessing attacks) Let A be a process and $\tilde{w} \subseteq bn(A)$. We say that A is *resistant to guessing attacks* against $\tilde{w}$ if, for every process B such that $A \rightarrow^* B$, we have that the frame $\phi(B)$ is resistant to guessing attacks against $\tilde{w}$.

Example 10 Consider the handshake protocol described in Example 7. An interesting problem arises if the shared key w is a weak secret, i.e. vulnerable to brute-force off-line testing. In such a case, the protocol has a guessing attack against w. Indeed, we have that

$$\nu w.(A \mid B) \rightarrow^* D \text{ with } \phi(D) = \nu w.\nu n.(\{^{\mathsf{senc}(n,w)}/_{x_1}\} \mid \{^{M}/_{x_2}\})$$

where $M = \mathsf{senc}(\mathsf{f}(\mathsf{sdec}(\mathsf{senc}(n,w),w)),w) =_{\mathsf{E}_{\mathsf{enc}}} \mathsf{senc}(\mathsf{f}(n),w)$. The frame $\phi(D)$ is not resistant to guessing attacks against w. The test $\mathsf{f}(\mathsf{sdec}(x_1,x)) \stackrel{?}{=} \mathsf{sdec}(x_2,x)$ allows us to distinguish the two associated frames:

- $\nu w, n.(\{^{\mathsf{senc}(n,w)}/_{x_1}\} \mid \{^{M}/_{x_2}\} \mid \{^{w}/_{x}\})$, and
- $\nu w', w, n.(\{^{\mathsf{senc}(n,w)}/_{x_1}\} \mid \{^{M}/_{x_2}\} \mid \{^{w'}/_{x}\})$.

Example 11 Consider the EKE password protocol described in Example 8. An execution of this protocol in the presence of a passive attacker yields the frame $\nu w.\phi_{\mathsf{EKE}}$ where

$$\phi_{\mathsf{EKE}} = \nu k, r, na, nb.$$
$$\{^{\mathsf{senc}(\mathsf{pk}(k),w)}/_{x_1}, {}^{\mathsf{senc}(\mathsf{aenc}(r,\mathsf{pk}(k)),w)}/_{x_2}, {}^{\mathsf{senc}(na,r)}/_{x_3}, {}^{\mathsf{senc}(\langle na,nb\rangle,r)}/_{x_4}, {}^{\mathsf{senc}(nb,r)}/_{x_5}\}$$

We have that $\nu w.(\phi_{\mathsf{EKE}} \mid \{^{w}/_{x}\}) \approx \nu w, w'.(\phi_{\mathsf{EKE}} \mid \{^{w'}/_{x}\})$. We have verified this static equivalence using the YAPA tool [11]. So, this allows us to conlude that

the EKE protocol is resistant to guessing attacks against w for one session of the protocol and in presence of a passive attacker. This result has also been shown by Corin *et al.* [23] with a slight difference in the modelling of the protocol. Actually, using an automatic tool such as ProVerif [15], it is possible to show that this protocol is resistant to guessing attacks against w. This is in contrast with some results presented in [36] where it is shown that many variants of EKE are vulnerable to active guessing attacks. However, those attacks rely on number-theoretic properties of the asymmetric encryption scheme that is used (e.g. RSA), and they are out of scope of the modelling we propose here.

— PART II: Composition Results —

It is well-known that composition works when processes do not share any secret, the so-called disjoint case. This is formally stated in the proposition below whose proof is given in Appendix A.

Proposition 1 *Let $A_1, \ldots, A_k$ be k extended processes such that $A \stackrel{\mathsf{def}}{=} A_1 \mid \ldots \mid A_k$ is also an extended process, and $w_i \in bn(A_i)$ for each $i \in \{1, \ldots, k\}$.*

1. *Let t be a ground term that occurs as a subterm in A_i for some $i \in \{1, \ldots, k\}$. If A_i preserves secrecy of t, then A preserves secrecy of t.*
2. *Let $\Phi = ev(\tilde{x}) \Rightarrow_{(\mathsf{inj})} ev'(\tilde{x})$ be a correspondence property (injective or not). If Φ holds on each A_i, then Φ holds on A.*
3. *If each A_i is resistant to guessing attack against w_i, then A is resistant to guessing attack against $w_1, \ldots, w_k$.*

A first idea to establish a composition result is to see under which conditions we can go back to the disjoint case. In the following sections (Sections 6 and 7), we will see that this is indeed possible provided that processes are tagged and only share some passwords. In Section 5, we will establish a result that will allow us to compose frames, and to derive an interesting result in presence of a passive attacker.

5 Composing Frames

In this section, we will review the definition of resistance against guessing attacks for a frame. We first show the equivalence of three definitions of resistance against guessing attacks: the first definition is due to Baudet [12] and the second one is due to Corin *et al.* [23]. The last definition is given in a composable way and establishes our composition result (see Corollary 1).

Proposition 2 *Let ϕ be a frame such that $\phi \equiv \nu\tilde{w}.\phi'$. The three following statements are equivalent:*

1. *ϕ is resistant to guessing attacks against $\tilde{w}$ (according to Definition 8),*
2. *$\phi' \approx \nu\tilde{w}.\phi'$,*
3. *$\phi' \approx \phi'\{^{\tilde{w}'}/_{\tilde{w}}\}$ where $\tilde{w}'$ is a sequence of fresh names.*

Proof Let ϕ be a frame such that $\phi \equiv \nu\tilde{w}.\phi'$. We first establish that the two first statements are equivalent. Indeed, we have that:

$$
\begin{array}{lll}
 & \phi' \approx \nu\tilde{w}.\phi' & \\
\Leftrightarrow & \phi' \approx \nu\tilde{w}'.\phi'\{^{\tilde{w}'}/_{\tilde{w}}\} & \text{by } \alpha\text{-renaming} \\
\Leftrightarrow & \nu\tilde{w}.(\phi' \mid \{^{\tilde{w}}/_{\tilde{x}}\}) \approx \nu\tilde{w}.\nu\tilde{w}'.(\phi'\{^{\tilde{w}'}/_{\tilde{w}}\} \mid \{^{\tilde{w}}/_{\tilde{x}}\}) & \text{by Lemma 2 (item } 1.) \\
\Leftrightarrow & \nu\tilde{w}.(\phi' \mid \{^{\tilde{w}}/_{\tilde{x}}\}) \approx \nu\tilde{w}'.\nu\tilde{w}.(\phi' \mid \{^{\tilde{w}'}/_{\tilde{x}}\}) & \text{by } \alpha\text{-renaming}
\end{array}
$$

Now, we show that $3 \Rightarrow 2$. We have the following implications.

$$\begin{array}{ll} & \phi' \approx \phi'\{^{\tilde{w}'}/_{\tilde{w}}\} \\ \Rightarrow & \nu\tilde{w}.\phi' \approx \nu\tilde{w}.\phi'\{^{\tilde{w}'}/_{\tilde{w}}\} \quad \text{by Lemma 1 (item } \textit{1.}\text{)} \\ \Rightarrow & \nu\tilde{w}.\phi' \approx \phi'\{^{\tilde{w}'}/_{\tilde{w}}\} \quad \text{since } \tilde{w} \text{ does not occur in } \phi'\{^{\tilde{w}'}/_{\tilde{w}}\} \\ \Rightarrow & \nu\tilde{w}.\phi' \approx \phi' \quad \text{since } \phi' \approx \phi'\{^{\tilde{w}'}/_{\tilde{w}}\} \text{ by hypothesis} \end{array}$$

Finally, we prove that $2 \Rightarrow 3$.

$$\begin{array}{ll} & \phi' \approx \nu\tilde{w}.\phi' \\ \Rightarrow & \phi' \approx \nu\tilde{w}'.\phi'\{^{\tilde{w}'}/_{\tilde{w}}\} \quad \text{by } \alpha\text{-renaming} \\ \Rightarrow & \phi'\{^{\tilde{w}'}/_{\tilde{w}}\} \approx \nu\tilde{w}'.\phi'\{^{\tilde{w}'}/_{\tilde{w}}\} \quad \text{by Lemma 1 (item } \textit{2.}\text{)} \\ \Rightarrow & \phi'\{^{\tilde{w}'}/_{\tilde{w}}\} \approx \nu\tilde{w}.\phi' \quad \text{by } \alpha\text{-renaming} \\ \Rightarrow & \phi'\{^{\tilde{w}'}/_{\tilde{w}}\} \approx \phi' \quad \text{since } \phi' \approx \nu\tilde{w}.\phi' \text{ by hypothesis} \end{array}$$

This concludes the proof. □

Now, by relying on Proposition 2 (item *3.*), it is easy to show that resistance to guessing attack against $\tilde{w}$ for two frames that share only the names $\tilde{w}$ is a composable notion. This is formally stated in the corollary below:

Corollary 1 *Let $\phi_1 \equiv \nu\tilde{w}.\phi_1'$ and $\phi_2 \equiv \nu\tilde{w}.\phi_2'$ be two frames such that $\nu\tilde{w}.(\phi_1' \mid \phi_2')$ is also a frame (this can be achieved by using α-renaming).*

If ϕ_1 and ϕ_2 are resistant to guessing attacks against $\tilde{w}$ then $\nu\tilde{w}.(\phi_1' \mid \phi_2')$ is also resistant to guessing attacks against $\tilde{w}$.

Proof By relying on Proposition 2 (item *3.*), we have that $\phi_1' \approx \phi_1'\{^{\tilde{w}'}/_{\tilde{w}}\}$ and also that $\phi_2' \approx \phi_2'\{^{\tilde{w}'}/_{\tilde{w}}\}$. Now, thanks to Lemma 2 (item *2.*), we have that

- $\phi_1' \mid \phi_2' \approx \phi_1'\{^{\tilde{w}'}/_{\tilde{w}}\} \mid \phi_2'$, and
- $\phi_1'\{^{\tilde{w}'}/_{\tilde{w}}\} \mid \phi_2' \approx \phi_1'\{^{\tilde{w}'}/_{\tilde{w}}\} \mid \phi_2'\{^{\tilde{w}'}/_{\tilde{w}}\}$.

This allows us to conclude that $\phi_1' \mid \phi_2' \approx (\phi_1' \mid \phi_2')\{^{\tilde{w}'}/_{\tilde{w}}\}$ which means that the frame $\nu\tilde{w}.(\phi_1' \mid \phi_2')$ is resistant to guessing attacks against $\tilde{w}$. □

Note that a similar result does not hold for deducibility (see Definition 2): even if w is neither deducible from ϕ_1 nor from ϕ_2, it can be deducible from $\phi_1 \mid \phi_2$. Such an example is given below.

Example 12 Consider again the equational theory $\mathsf{E}_{\mathsf{enc}}$. Consider the two following frames: $\phi_1 = \{^{\mathsf{senc}(w,\mathsf{senc}(w,w))}/_{x_1}\}$ and $\phi_2 = \{^{\mathsf{senc}(w,w)}/_{x_2}\}$. We have that $\nu w.\phi_i \not\vdash_{\mathsf{E}} w$ for $i = 1, 2$ whereas $\nu w.(\{^{\mathsf{senc}(w,\mathsf{senc}(w,w))}/_{x_1}\} \mid \{^{\mathsf{senc}(w,w)}/_{x_2}\}) \vdash_{\mathsf{E}} w$. Indeed, the term $\mathsf{sdec}(x_1, x_2)$ is a recipe of the term w.

In the case of *password-only* protocols, i.e., protocols that only share a password between different sessions and do not have any other long-term shared secrets we have the following direct consequence. Considering a passive attacker who does not interact with the protocol during its execution, we can prove resistance against guessing attacks for an unbounded number of parallel sessions by proving only resistance against guessing attacks for a single session.

An example of a password-only protocol is the well-known EKE protocol [14]. It directly follows from our previous result that the protocol is secure for any number of sessions as the only secret shared between different sessions is the password w. An analysis of one session of this protocol has also been done in [23] (with a slight difference in the modeling).

6 Composing Different Protocols

In the active case, contrary to the passive case, resistance against guessing attacks does not compose: even if two protocols separately resist against guessing attacks on w, their parallel composition under the shared password w may be insecure.

Example 13 Consider the processes defined in Example 8 where the occurrence of 0 in B has been replaced by $\mathrm{out}(w)$. Let A' and B' be the two resulting processes. The process $\nu w.(A' \mid B')$ models a variant of the EKE protocol where B' outputs the password w if the authentication of A' succeeds. We have that $\nu w.A'$ and $\nu w.B'$ resist against guessing attacks on w. We have verified these statements by using the ProVerif tool [16]. However, the process $\nu w.(A' \mid B')$ trivially leaks w. More generally any secure password only authentication protocol can be modified in this way to illustrate that resistance against guessing attacks does not compose in the active case.

The previous example may not be entirely convincing, since there is no environment in which either of the separate processes $\nu w.A'$ and $\nu w.B'$ is *executable*. We do not give a formal definition of what it means for a process to be executable. Therefore we present a second example in which each of the constituent processes admits a complete execution by interacting with the environment. However, the example requires a somewhat contrived equational theory.

Example 14 Consider the following processes A_1 and A_2:

$$A_1 = \nu n_1.\mathrm{out}(\mathsf{f}_1(w, n_1)).\mathrm{in}(x).\mathrm{out}(\mathsf{f}_3(x, n_1))$$
$$A_2 = \nu n_2.\mathrm{in}(y).\mathrm{out}(\mathsf{f}_2(y, w, n_2))$$

and the equational theory induced by the equation $\mathsf{f}_3(\mathsf{f}_2(\mathsf{f}_1(x, y), x, z), y) = x$. We indeed have that w resists against guessing attacks in $\nu w.A_1$ and in $\nu w.A_2$; we have verified this using the ProVerif tool [15]. However, the name w is subject to a guessing attack in $\nu w.(A_1 \mid A_2)$. Indeed, we have that

$$\nu w.(A_1 \mid A_2) \to^* \nu w, n_1, n_2.(\{^{\mathsf{f}_1(w,n_1)}/_{x_1}\} \mid \{^{\mathsf{f}_2(\mathsf{f}_1(w,n_1),w,n_2)}/_{x_2}\} \mid \{^{M}/_{x_3}\})$$

where $M =_{\mathsf{E}} w$. The obtained frame $\nu w.\phi'$ is not resistant to guessing attack against w. We indeed have that $\nu w.\phi' \vdash w$, and hence $\phi' \not\approx \phi'\{^{w'}/_{w}\}$. The see this, consider for instance the test $x_3 \stackrel{?}{=} w$.

This example shows that there is no hope to obtain a general composition result that holds for an arbitrary equational theory. Thus, to reach our goal, we need to consider a restricted class of protocols: the class of *well-tagged* protocols.

6.1 Well-Tagged Protocols

Intuitively, a protocol is well-tagged w.r.t. a secret w if all the occurrences of w are of the form $\mathsf{h}(c, w)$. We require that h is a hash function (i.e., has no equations in the equational theory), and c is a name, which we call the *tag*. The idea is that if each protocol is tagged with a different name (e.g. the name of the protocol) then the protocols compose safely. Note that a protocol can be very easily transformed into a well-tagged protocol (see Section 6.2). In the remainder, we will consider an arbitrary equational theory E, provided there is no equation for h.

Definition 10 (well-tagged) Let M be a term and w be a name. We say that M is *c-tagged* w.r.t. w if there exists M' such that $M'\{^{\mathsf{h}(c,w)}/_{w}\} =_{\mathsf{E}} M$.

A term is said *well-tagged* w.r.t. w if it is c-tagged w.r.t. w for some name c. An extended process A is *c-tagged* if any term occurring in it is c-tagged. An extended process is *well-tagged* if it is c-tagged for some name c.

Other ways of tagging a protocol exist in the literature. For example, in [25] encryptions are tagged to ensure that they cannot be used to attack other instances of the protocol. That particular method would not work here; on the contrary, that kind of tagging is likely to add guessing attacks.

Example 15 Let $A = \nu w, s.\mathrm{out}(\mathsf{senc}(s, w))$. We have that A is resistant to guessing attacks against w. However, the protocol, which is well-tagged according to the definition given in [25], is not. Indeed,

$$A' = \nu w, s.\mathrm{out}(\mathsf{senc}(\langle c, s\rangle, w))$$

is not resistant to guessing attack against w. The tag c which is publicly known can be used to mount such an attack.

Another tagging method we considered is to replace w by $\langle c, w\rangle$ (instead of $\mathsf{h}(c, w)$), which has the advantage of being computationally cheaper. This transformation does not work, although the only counterexamples we have are rather contrived. For example, this transformation does not preserve resistance against guessing attacks as soon as the equational theory allows one to test whether a given message is a pair. In particular this is possible in the theory $\mathsf{E}_{\mathsf{enc}}$ by testing whether $\langle \mathsf{proj}_1(x), \mathsf{proj}_2(x)\rangle =_{\mathsf{E}_{\mathsf{enc}}} x$.

Example 16 Consider the equational theory $\mathsf{E}_{\mathsf{enc}}$ and the following process:

$$A = \nu w, k.\mathrm{out}(\mathsf{senc}(w, k)).\mathrm{in}(x).\ \mathrm{if}\ \mathsf{proj}_1(\mathsf{dec}(x, k)) = c\ \mathrm{then}\ \mathrm{out}(w).$$

The process A is resistant to guessing attacks against w since the last instruction can never been executed. However, the protocol obtained by replacing w by $\langle c, w\rangle$ is clearly not.

Note that we can built a similar example without using c in the specification of A. We can simply compare the first component of two ciphertexts issued from the protocols. This should lead to an equality (i.e. a test) which does not necessarily exist in the original protocol.

6.2 Transformation to Obtain Well-Tagged Protocols

In the previous section, we introduced the notion of well-tagged protocols, a necessary condition to ensure composition. Unfortunately, most of the existing protocols are not well-tagged. In this section, we give a simple, syntactic transformation which allows us to transform any protocol into a well-tagged one. Let $\nu w.A$ be a process resistant to guessing attacks against w, the transformed process is defined as $\nu w.(A\{^{\mathsf{h}(c,w)}/_{w}\})$: any occurrence of the password w in A is replaced by $\mathsf{h}(c, w)$. In this section, we show that this transformation preserves the security properties introduced in Section 4. More precisely, we have that:

Theorem 1 *Let c be a name and $A \equiv \nu w.A'$ be a process such that $c \notin bn(A)$.*

1. *Let t be a ground term that occurs as a subterm in A. If A preserves secrecy of t, then $\nu w.(A'\{^{\mathsf{h}(c,w)}/_w\})$ preserves secrecy of $t\{^{\mathsf{h}(c,w)}/_w\}$.*
2. *Let $\Phi = ev(\tilde{x}) \Rightarrow_{(\mathsf{inj})} ev'(\tilde{x})$ be a correspondence property (injective or not). If Φ holds on A, then Φ holds on $\nu w.(A'\{^{\mathsf{h}(c,w)}/_w\})$.*
3. *If A is resistant to guessing attacks against w, then $\nu w.(A'\{^{\mathsf{h}(c,w)}/_w\})$ is also resistant to guessing attacks against w.*

Theorem 1 is proved by contradiction in two main steps. Omitted proofs are detailed in Appendix B. The first step relies on Proposition 3, in which we show how to map an execution of a well-tagged protocol to an execution of the original (not well-tagged) protocol. We maintain a strong connection between the two executions.

Proposition 3 *Let A be a process with $c, w \notin bn(A)$ and $A'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} A$ for some A'. If $\nu w.A \xrightarrow{\ell} \overline{B}$, then $\overline{B} \equiv \nu w.B$ and there exists a process B' and a label ℓ' such that $B'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} B$, $\ell'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} \ell$, and $\nu w.A' \xrightarrow{\ell'} \nu w.B'$.*

Finally, in a second step, we have to show that the same type of attacks can be mounted on the resulting trace. This relies either on Lemma 3, Lemma 4 or Lemma 5 depending on the security property under study. Actually, Lemma 3 and Lemma 4 are also useful to deal with the cases of an input and a conditional in the proof of Proposition 3.

Regarding secrecy preservation, we prove the following lemma:

Lemma 3 *Let ϕ be a frame such that $c, w \notin bn(\phi)$ and $\phi'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} \phi$ for some ϕ'. If $\nu w.\phi \vdash_{\mathsf{E}} M$ then there exists M' such that $M'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} M$ and $\nu w.\phi' \vdash_{\mathsf{E}} M'$.*

Regarding correspondence properties, we establish the following result:

Lemma 4 *Let M, N, M' and N' be four terms such that $M =_{\mathsf{E}} M'\{^{\mathsf{h}(c,w)}/_w\}$ and $N =_{\mathsf{E}} N'\{^{\mathsf{h}(c,w)}/_w\}$. Then, we have that*

$$M =_{\mathsf{E}} N \text{ if, and only if, } M' =_{\mathsf{E}} N'$$

Regarding resistance against guessing attacks, we show that static equivalence is preserved by the transformation $\{^{\mathsf{h}(c,w)}/_w\}$. This is crucial to ensure that the transformation does not introduce guessing attack.

Lemma 5 *Let ϕ_1 and ϕ_2 be two frames such that $\phi_1 \approx \phi_2$. Let w, c be such that $w, c \notin bn(\phi_1) \cup bn(\phi_2)$. We have that*

$$\phi_1\{^{\mathsf{h}(c,w)}/_w\} \approx \phi_2\{^{\mathsf{h}(c,w)}/_w\}.$$

Now, we are able to prove Theorem 1.

Proof Assume that $\nu w.(A'\{^{\mathsf{h}(c,w)}/_w\})$ admits an attack. This means that there exists a process $\overline{B}$, and some labels $\ell_1, \ldots, \ell_n$ such that

$$\nu w.(A'\{^{\mathsf{h}(c,w)}/_w\}) \xrightarrow{\ell_1} \ldots \xrightarrow{\ell_n} \overline{B}$$

and depending on the security property under study, we have that:

1. *(secrecy)* $\phi(\overline{B}) \vdash_{\mathsf{E}} t\{^{\mathsf{h}(c,w)}/_{w}\}$; or
2. *(non-injective correspondence property)* there exist j_0 and a substitution σ such that $\ell_{j_0} =_{\mathsf{E}} ev(\tilde{x}\sigma)$ and $\ell_j \neq_{\mathsf{E}} ev'(\tilde{x}\sigma)$ for any $j \leq j_0$ (the case of an injective correspondence property can be done in a similar way); or
3. *(guessing attack)* the frame $\phi(\overline{B})$ is not resistant to guessing attacks against w.

By applying Proposition 3, we easily obtained that $\overline{B} \equiv \nu w.B$ for some process B and there exists B' and some labels $\ell'_1, \ldots, \ell'_n$ such that $B'\{^{\mathsf{h}(c,w)}/_{w}\} =_{\mathsf{E}} B$, $\ell'_1\{^{\mathsf{h}(c,w)}/_{w}\} =_{\mathsf{E}} \ell_1, \ldots, \ell'_n\{^{\mathsf{h}(c,w)}/_{w}\} =_{\mathsf{E}} \ell_n$ and $\nu w.A' \xrightarrow{\ell'_1} \ldots \xrightarrow{\ell'_n} \nu w.B'$. To conclude, it remains to show that this trace admits an attack.

1. *(secrecy)* We have to show that $\nu w.\phi(B') \vdash t$. We have that $c, w \notin bn(\phi(B))$, $\phi(B')\{^{\mathsf{h}(c,w)}/_{w}\} =_{\mathsf{E}} \phi(B)$, and $\phi(\overline{B}) = \nu w.\phi(B) \vdash_{\mathsf{E}} t\{^{\mathsf{h}(c,w)}/_{w}\}$. Thanks to Lemma 3, we deduce that there exist t' such that $t'\{^{\mathsf{h}(c,w)}/_{w}\} =_{\mathsf{E}} t\{^{\mathsf{h}(c,w)}/_{w}\}$ and $\nu w.\phi(B') \vdash_{\mathsf{E}} t'$. Now, using Lemma 4, we easily conclude that $t =_{\mathsf{E}} t'$, and thus $\nu w.\phi(B') \vdash_{\mathsf{E}} t$. This means that A does not preserve secrecy of t.
2. *(non-injective correspondence property)* We have to show that there exist j'_0 and a substitution σ' with domain $\tilde{x}$ such that $\ell'_{j'_0} =_{\mathsf{E}} ev(\tilde{x}\sigma')$ and $\ell'_j \neq_{\mathsf{E}} ev'(\tilde{x}\sigma')$ for any $j \leq j'_0$. Let $j'_0 = j_0$ and σ' be a substitution such that $\tilde{x}\sigma = \tilde{x}\sigma'\{^{\mathsf{h}(c,w)}/_{w}\}$. Such a substitution exists since $ev(\tilde{x}\sigma) =_{\mathsf{E}} \ell_{j_0} =_{\mathsf{E}} \ell'_{j_0}\{^{\mathsf{h}(c,w)}/_{w}\}$. We have that $\ell_{j_0} =_{\mathsf{E}} ev(\tilde{x}\sigma)$, and thus $\ell'_{j_0}\{^{\mathsf{h}(c,w)}/_{w}\} =_{\mathsf{E}} ev(\tilde{x}\sigma')\{^{\mathsf{h}(c,w)}/_{w}\}$. Thanks to Lemma 4, we easily deduce that $\ell'_{j_0} = \ell'_{j'_0} = ev(\tilde{x}\sigma')$. In the same way, we can show that $\ell'_j \neq_{\mathsf{E}} ev'(\tilde{x}\sigma')$ for any $j \leq j'_0$. This allows us to conclude. For injective correspondence properties, the proof can be done in a similar way.
3. *(guessing attack)* We have to show that $\phi(B')$ is not resistant to guessing attack against w, i.e. $\phi(B') \not\approx \phi(B')\{^{w'}/_{w}\}$. Assume that $\phi(B') \approx \phi(B')\{^{w'}/_{w}\}$, thanks to Lemma 5, we easily obtain that
   - $\phi(B) =_{\mathsf{E}} \phi(B')\{^{\mathsf{h}(c,w)}/_{w}\} \approx (\phi(B')\{^{w'}/_{w}\})\{^{\mathsf{h}(c,w)}/_{w}\} = \phi(B')\{^{w'}/_{w}\}$, and
   - $\phi(B') = \phi(B')\{^{\mathsf{h}(c,w')}/_{w'}\} \approx (\phi(B')\{^{w'}/_{w}\})\{^{\mathsf{h}(c,w')}/_{w'}\} =_{\mathsf{E}} \phi(B)\{^{w'}/_{w}\}$.

   Since $\phi(B') \approx \phi(B')\{^{w'}/_{w}\}$, we obtain $\phi(B) \approx \phi(B)\{^{w'}/_{w}\}$ which contradicts the fact that $\phi(\overline{B})$ is not resistant to guessing attacks against w. □

6.3 Composition Theorem

We show that protocols that are separately secure can be safely composed provided that they use different tags. The following theorem formalizes the intuition that replacing the shared password w with a hash of the password and a tag, i.e. $\mathsf{h}(t_i, w)$, is similar to using different passwords which implies composition. Note that the theorem below is stated considering ground terms as tags (and not only names). This is not really needed to compose different protocols but this generalization will be useful in Section 7 to compose sessions coming from the same protocols.

Theorem 2 *Let $t_1, \ldots, t_k$ be distinct ground terms modulo E, and $\nu w.A_1, \ldots, \nu w.A_k$ be k extended processes such that $A \stackrel{\mathsf{def}}{=} \nu w.(A_1\{^{\mathsf{h}(t_1,w)}/_{w}\} \mid \ldots \mid A_k\{^{\mathsf{h}(t_k,w)}/_{w}\})$ is also an extended process (this can be achieved using α-renaming). Moreover, for each $1 \leq i \leq k$, we assume that $\phi(A) \vdash_{\mathsf{E}} t_i$.*

1. *Let t be a ground term that occurs as a subterm in A_i for some $i \in \{1, \ldots, k\}$. If $\nu w.A_i$ preserves secrecy of t, then A preserves secrecy of $t\{^{\mathsf{h}(t_i,w)}/_w\}$.*
2. *Let $\Phi = ev(\tilde{x}) \Rightarrow_{(\mathsf{inj})} ev(\tilde{x})$ be a correspondence property (injective or not). If Φ holds on each $\nu w.A_i$, then Φ holds on A.*
3. *If each $\nu w.A_i$ is resistant to guessing attacks against w, then the process A is also resistant to guessing attacks against w.*

To prove this theorem, we proceed in two steps as for Theorem 1. We first show in Proposition 4 how to map an execution of

$$A \stackrel{\mathsf{def}}{=} \nu w.(A_1\{^{\mathsf{h}(t_1,w)}/_w\} \mid \ldots \mid A_k\{^{\mathsf{h}(t_k,w)}/_w\}) \text{ (same password)}$$

to an execution of

$$\nu w_1.(A_1\{^{\mathsf{h}(c_1,w_1)}/_w\}) \mid \ldots \mid \nu w_k.(A_k\{^{\mathsf{h}(c_k,w_k)}/_w\}) \quad \text{(different passwords)}$$

by maintaining a strong connection between these two derivations. Intuitively, as each A_i is t_i-tagged and t_i are distinct ground terms modulo E, we can simply replace $\mathsf{h}(t_i, w)$ by $\mathsf{h}(c_i, w_i)$ in any execution. We denote by $\delta_{w_i,w}$ the replacement $\{^w/_{w_1}\} \ldots \{^w/_{w_k}\}$, by $\delta_{w_i,\mathsf{h}(c_i,w_i)}$ the replacement $\{^{\mathsf{h}(c_1,w_1)}/_{w_1}\} \ldots \{^{\mathsf{h}(c_k,w_k)}/_{w_k}\}$ and by δ_{c_i,t_i} the replacement $\{^{t_1}/_{c_1}\} \ldots \{^{t_k}/_{c_k}\}$.

Proposition 4 *Let $t_1, \ldots, t_k$ be distinct ground terms modulo E and $c_1, \ldots, c_k, w_1, \ldots, w_k$ be distinct fresh names. Let $\nu\tilde{n}.A$ be an extended process such that $bn(A) = \emptyset$, $w \notin fn(A)$, and $A =_{\mathsf{E}} A'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ for some A' such that $c_1, \ldots, c_k \notin fn(A')$. Moreover, we assume that $w, w_1, \ldots, w_k, c_1, \ldots, c_k \notin \tilde{n}$.*

Let $\overline{B}$ be such that $\nu w.\nu\tilde{n}.(A\delta_{c_i,t_i}\delta_{w_i,w}) \xrightarrow{\ell} \overline{B}$. Moreover, when $\ell = in(\tilde{M})$ we assume that $w_1, \ldots, w_k, c_1, \ldots, c_k \notin fn(\tilde{M})$. Then there exist extended processes B, B', and labels ℓ_0, ℓ' such that:

- *$\overline{B} \equiv \nu w.\nu\tilde{n}.(B\delta_{c_i,t_i}\delta_{w_i,w})$ with $bn(B) = \emptyset$ and $w \notin fn(B)$, $\ell = \ell_0\delta_{c_i,t_i}\delta_{w_i,w}$, and*
- *$B =_{\mathsf{E}} B'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ with $c_1, \ldots, c_k \notin fn(B')$, $\ell_0 =_{\mathsf{E}} \ell'\delta_{w_i,\mathsf{h}(c_i,w_i)}$, and*
- *$\nu w_1 \ldots \nu w_k.\nu\tilde{n}.A \xrightarrow{\ell_0} \nu w_1 \ldots \nu w_k.\nu\tilde{n}.B$.*

Then, we show in a second step that the same type of attacks can be mounted on the resulting trace. As for Theorem 1, this relies on three lemmas, Lemma 6, 7 and 8. As before, Lemmas 6 and 7 are also useful to deal with the cases of an input and a conditional in the proof of Proposition 4. This is a bit technical because mapping $w_1, \ldots, w_k$ on the same password can introduce additional equalities between terms. Intuitively, the results hold because the frames are well-tagged, and different passwords are tagged with distinct terms.

Regarding secrecy preservation, we show that if a frame, obtained by executing k protocols sharing a same password, allows one to deduce a term $\overline{M}$, then the frame obtained by the corresponding execution of the protocols with different passwords also allows one to deduce a similar term M. This lemma is also useful to deal with the case of an input in Proposition 4.

Lemma 6 *Let $t_1, \ldots, t_k$ be distinct ground terms modulo E and $c_1, \ldots, c_k, w_1, \ldots, w_k$ be distinct fresh names. Let $\phi = \nu\tilde{n}.\sigma$, $\tilde{\phi} = \nu\tilde{n}.\tilde{\sigma}$ and $\phi' = \nu\tilde{n}.\sigma'$ be three frames such that $w \notin fn(\sigma)$, and $w, w_1, \ldots, w_k, c_1, \ldots, c_k \notin \tilde{n}$. Moreover, we assume that $\sigma\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{\sigma}$, $\sigma =_{\mathsf{E}} \sigma'\delta_{w_i,\mathsf{h}(c_i,w_i)}$, and $c_1, \ldots, c_k \notin fn(\sigma')$. If $\nu w.\tilde{\phi} \vdash_{\mathsf{E}} \tilde{M}$ and $\{w_1, \ldots, w_k, c_1, \ldots, c_k\} \cap fn(\tilde{M}) = \emptyset$ for some ground term $\tilde{M}$ then there exist ground terms M, M' such that $c_1, \ldots, c_k \notin fn(M')$, $w \notin fn(M)$, $M\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{M}$, $M =_{\mathsf{E}} M'\delta_{w_i,\mathsf{h}(c_i,w_i)}$, and $\nu w_1 \ldots \nu w_k.\phi \vdash_{\mathsf{E}} M$.*

Regarding correspondence properties, we have to show that if a trace obtained by executing k protocols sharing a same password, allows one to falsify the correspondence property Φ, then the trace obtained by the corresponding execution of the protocols with different passwords also allows one to falsify this correspondence property in a similar way. For this, we establish the following result which is also useful to deal with the case of conditionals in Proposition 4.

Lemma 7 *Let $t_1, \ldots, t_k$ be distinct ground terms modulo E and $c_1, \ldots, c_k, w_1, \ldots, w_k$ be distinct fresh names. Let M, N, $\tilde{M}$ and $\tilde{N}$ be four terms such that*

- *$\tilde{M} = M\delta_{c_i,t_i}\delta_{w_i,w}$ and $\tilde{N} = N\delta_{c_i,t_i}\delta_{w_i,w}$ with $w \notin fn(M) \cup fn(N)$;*
- *$M =_{\mathsf{E}} M'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ and $N =_{\mathsf{E}} N'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ for some terms M' and N' such that $c_1, \ldots, c_k \notin fn(M') \cup fn(N')$.*

Then, we have that $M =_{\mathsf{E}} N$ if and only if $\tilde{M} =_{\mathsf{E}} \tilde{N}$.

Regarding resistance against guessing attacks, we show that if a frame, obtained by executing k protocols sharing a same password, is vulnerable to guessing attacks then the frame obtained by the corresponding execution of the protocols with different passwords is also vulnerable to guessing attacks.

Lemma 8 *Let $t_1, \ldots, t_k$ be distinct ground terms modulo E. Let $c_1, \ldots, c_k, w_1, \ldots, w_k$ be distinct fresh names, and $\phi = \nu\tilde{n}.\sigma$ be a frame such that $c_1, \ldots, c_k, w_1, \ldots, w_k \notin \tilde{n}$, and $\sigma =_{\mathsf{E}} \sigma_0\delta_{w_i,\mathsf{h}(c_i,w_i)}$ for some substitution σ_0. Let w be a fresh name, and $\psi = \nu\tilde{n}.(\sigma\delta_{c_i,t_i}\delta_{w_i,w})$. For each $1 \leq i \leq k$, we also assume that $\nu w.\psi \vdash_{\mathsf{E}} t_i$.*

If $\nu\tilde{w}.\phi$ is resistant to guessing attacks against $\tilde{w} = \{w_1, \ldots, w_k\}$, then $\nu w.\psi$ is resistant to guessing attacks against w.

Now, we can prove Theorem 2.

Proof We prove our composition result by contradiction. Assume that the process $A \stackrel{\mathsf{def}}{=} \nu w.(A_1\{^{\mathsf{h}(t_1,w)}/_w\} \mid \ldots \mid A_k\{^{\mathsf{h}(t_k,w)}/_w\})$ admits an attack. First, we show that the process

$$A_0 \stackrel{\mathsf{def}}{=} \nu w_1.(A_1\{^{\mathsf{h}(c_1,w_1)}/_w\}) \mid \ \ldots \ \mid \nu w_k.(A_k\{^{\mathsf{h}(c_k,w_k)}/_w\}) \qquad \text{(different passwords)}$$

also admits an attack. Then, Proposition 1 will allow us to derive the existence of an attack on $\nu w_{i_0}.A_{i_0}\{^{\mathsf{h}(c_{i_0},w_{i_0})}/_{w_{i_0}}\}$ for some $i_0 \in \{1, \ldots, n\}$. Lastly, we will conclude to the existence of an attack on $\nu w_{i_0}.A_{i_0}$ relying on Theorem 1. Below, we instantiate the sketch described above depending on the security property under study.

By definition of an attack, we have that there exists a trace

$$A = \overline{A}_0 \stackrel{\mathsf{def}}{=} \nu w.(A_1\{^{\mathsf{h}(t_1,w)}/_w\} \mid \ldots \mid A_k\{^{\mathsf{h}(t_k,w)}/_w\}) \xrightarrow{\overline{\ell_1}} \ldots \xrightarrow{\overline{\ell}_n} \overline{A}_n$$

such that:

1. *(secrecy)* $\phi(\overline{A}_n) \vdash_{\mathsf{E}} \bar{t}$ (where $\bar{t}$ is a subterm of $A_{i_0}\{^{\mathsf{h}(t_{i_0},w)}/_w\}$ for some $i_0 \in \{1,\ldots,k\}$); or
2. *(non-injective correspondence property)* there exists j_0 and a substitution σ such that $\overline{\ell_{j_0}} =_{\mathsf{E}} ev(\tilde{x}\sigma)$ and $\overline{\ell_j} \neq_{\mathsf{E}} ev(\tilde{x}\sigma)$ for any $j \leq j_0$ (the case of an injective correspondence property can be done in a similar way); or
3. *(guessing attack)* the frame $\phi(\overline{A}_n)$ is not resistant to guessing attacks against w.

We assume w.l.o.g. that the names $w_1,\ldots,w_k$ and $c_1,\ldots,c_k$, which do not occur in $\overline{A}_0$, are not used along the derivation. By definition of a process, we have that $A_i \equiv \nu\tilde{n}_i.A_i^0$ for some sequence $\tilde{n}_i$ and some process A_i^0 with $bn(A_i^0) = \emptyset$. We denote by $\tilde{n}$ the sequence $\tilde{n}_1,\ldots,\tilde{n}_k$.

By iterating Proposition 4 on $A \equiv \nu w.\nu\tilde{n}.(A_1^0\{^{\mathsf{h}(t_1,w)}/_w\} \mid \ldots \mid A_k^0\{^{\mathsf{h}(t_k,w)}/_w\})$, we have that there exist two extended processes A_n, A'_n and two sequences of labels $\ell_1^0,\ldots,\ell_n^0$ and $\ell'_1,\ldots,\ell'_n$ such that:

- $\overline{A}_n \equiv \nu w.\nu\tilde{n}.(A_n\delta_{c_i,t_i}\delta_{w_i,w})$ with $bn(A_n) = \emptyset$, $w \notin fn(A_n)$, and $\overline{\ell_j} = \ell_j^0\delta_{c_i,t_i}\delta_{w_i,w}$ for any $j \in \{1,\ldots,n\}$;
- $A_n =_{\mathsf{E}} A'_n\delta_{w_i,\mathsf{h}(c_i,w_i)}$ with $c_1,\ldots,c_k \notin fn(A'_n)$, and $\ell_j^0 =_{\mathsf{E}} \ell'_j\delta_{w_i,\mathsf{h}(c_i,w_i)}$ for any $j \in \{1,\ldots,n\}$, and
- $A_0 \xrightarrow{\ell_1^0} \ldots \xrightarrow{\ell_n^0} \nu w_1.\ldots.\nu w_k.\nu\tilde{n}.A_n$.

Now, we show that the trace $A_0 \xrightarrow{\ell_1^0} \ldots \xrightarrow{\ell_n^0} \nu w_1.\ldots.\nu w_k.\nu\tilde{n}.A_n$ also admits an attack and we conclude. We distinguish three cases depending on the security property under study.

1. *(secrecy)* We know that $\bar{t} = t\{^{\mathsf{h}(t_{i_0},w)}/_w\}$ for some t that occurs as a subterm of A_{i_0}. First, we show that $\nu w_1.\ldots.\nu w_k.\nu\tilde{n}.\phi(A_n) \vdash_{\mathsf{E}} t\{^{\mathsf{h}(c_{i_0},w_{i_0})}/_w\}$. We have that $\phi(\overline{A}_n) \vdash_{\mathsf{E}} \bar{t}$. Thanks to Lemma 6, we deduce that there exists t_0 and t' such that $\nu w_1 \ldots \nu w_k.\nu\tilde{n}.\phi(A_n) \vdash_{\mathsf{E}} t_0$, $t_0 =_{\mathsf{E}} t'\delta_{w_i,\mathsf{h}(c_i,w_i)}$, and $\bar{t} = t_0\delta_{c_i,t_i}\delta_{w_i,w}$. Hence we have that $t_0 =_{\mathsf{E}} t\{^{\mathsf{h}(c_{i_0},w_{i_0})}/_w\}$. Then, applying Proposition 1, we deduce that $\nu w_{i_0},\nu\tilde{n}_{i_0}.(A_{i_0}^0\{^{\mathsf{h}(c_{i_0},w_{i_0})}/_w\})$ does not preserve secrecy of t_0. Using α-renaming, we deduce that $\nu w.\nu\tilde{n}_{i_0}.A_{i_0}^0\{^{\mathsf{h}(c_{i_0},w)}/_w\}$ does not preserve secrecy of $t_0\{^{w}/_{w_{i_0}}\} = t\{^{\mathsf{h}(c_{i_0},w)}/_w\}$. Lastly, relying on Theorem 1, we conclude that $\nu w.\nu\tilde{n}_{i_0}.A_{i_0}^0 \equiv \nu w.A_{i_0}$ does not preserve secrecy of t. This allows us to conclude.
2. *(non-injective correspondence property)* First, we have to show that there exist j'_0 and a substitution σ' with domain $\tilde{x}$ such that $\ell_{j'_0}^0 =_{\mathsf{E}} ev(\tilde{x}\sigma')$ and $\ell_j \neq_{\mathsf{E}} ev'(\tilde{x}\sigma')$ for any $j \leq j'_0$. Let $j'_0 = j_0$ and σ' be a substitution such that $x\sigma = (x\sigma')\delta_{c_i,t_i}\delta_{w_i,w}$. Such a substitution exists since $ev(\tilde{x}\sigma) =_{\mathsf{E}} \overline{\ell_{j_0}} =_{\mathsf{E}} \ell_{j_0}^0\delta_{c_i,t_i}\delta_{w_i,w}$. We have that $\overline{\ell_{j_0}} =_{\mathsf{E}} ev(\tilde{x}\sigma)$, and thus $\ell_{j_0}^0\delta_{c_i,t_i}\delta_{w_i,w} =_{\mathsf{E}} ev(\tilde{x}\sigma')\delta_{c_i,t_i}\delta_{w_i,w}$. Thanks to Lemma 7, we easily deduce that $\ell_{j_0}^0 = \ell_{j'_0}^0 = ev(\tilde{x}\sigma')$. In the same way, we can show that $\ell_j^0 \neq_{\mathsf{E}} ev'(\tilde{x}'\sigma')$ for any $j \leq j'_0$. Then, applying Proposition 1, we deduce that the correspondence property Φ does not hold on $\nu w_{i_0},\nu\tilde{n}_{i_0}.(A_{i_0}^0\{^{\mathsf{h}(c_{i_0},w_{i_0})}/_w\})$. Using α-renaming, we deduce that Φ does not hold on $\nu w.\nu\tilde{n}_{i_0}.A_{i_0}^0\{^{\mathsf{h}(c_{i_0},w)}/_w\}$. Lastly, relying on Theorem 1, we conclude

that the correspondence property Φ does not hold on $\nu w.\nu\tilde{n}_{i_0}.A^0_{i_0} \equiv \nu w.A_{i_0}$. This allows us to conclude. For injective correspondence properties, the proof can be done in a similar way.

3. *(guessing attack)* First, we have to show that $\nu w_1, \ldots, w_k.\nu\tilde{n}.\phi(A_n)$ is not resistant to guessing attacks against $w_1, \ldots, w_k$. Actually, this is a direct consequence of Lemma 8. Then, applying Proposition 1, we deduce that there exists $i_0 \in \{1, \ldots, k\}$ such that $\nu w_{i_0}, \nu\tilde{n}_{i_0}.(A^0_{i_0}\{^{\mathsf{h}(c_{i_0}, w_{i_0})}/_w\})$ is not resistant to guessing attack against w_{i_0}. Relying on Theorem 1, we conclude that $\nu w.\nu\tilde{n}_{i_0}.A^0_{i_0} \equiv \nu w.A_{i_0}$ is not resistant to guessing attack against w. □

7 Composing Different Sessions

Again, there is no hope to be able to compose different sessions without introducing new attacks. However, such a composition result holds for a class of protocol. We now define a protocol transformation which establishes a dynamic tag that will guarantee composition (see Section 7.1). Then, we will establish our composition result (see Section 7.2).

7.1 Our Transformation

To establish such a tag that serves as a session identifier all participants generate a fresh nonce, that is sent to all other participants. This is similar to the establishment of session identifiers proposed by Barak *et al.* [10]. The sequence of these nonces is then used to tag the password. Note that an active attacker may interfere with this initialization phase and may intercept and replace some of the nonces. However, since each participant generates a fresh nonce, these tags are indeed distinct for each session. This transformation is formally defined as follows.

Definition 11 (transformation $\overline{\mathcal{P}}$) Let $\mathcal{P} = \nu w.(\nu\tilde{m}_1.P_1 \mid \ldots \mid \nu\tilde{m}_\ell.P_\ell)$ be a password protocol specification. Let $n_1, \ldots, n_\ell$ be fresh names and $\{x_i^j \mid 1 \leq i, j \leq \ell\}$ be a set of fresh variables. We define the protocol specification $\overline{\mathcal{P}}$ as follows:

$$\overline{\mathcal{P}} = \nu w.(\nu\tilde{m}_1, n_1.\overline{P_1} \mid \ldots \mid \nu\tilde{m}_\ell, n_\ell.\overline{P_\ell})$$

where:

- $\overline{P_i} = \mathrm{in}(x_i^1).\ldots.\mathrm{in}(x_i^{i-1}).\mathrm{out}(n_i).\mathrm{in}(x_i^{i+1})\ldots\mathrm{in}(x_i^\ell).P_i\{^{\mathsf{h}(tag_i, w)}/_w\}$; and
- $tag_i = \langle x_i^1, \langle \ldots \langle x_i^{i-1}, \langle n_i, \langle x_i^{i+1}, \langle \ldots \langle x_i^{\ell-1}, x_i^\ell \rangle \ldots \rangle \rangle \rangle \ldots \rangle \rangle$.

Example 17 We now illustrate our transformation on the EKE protocol that we introduced in Example 8. The informal description of the transformed EKE protocol is as follows.

$\mathrm{A} \to \mathrm{B}$:	n_1	(PRE.1)
$\mathrm{B} \to \mathrm{A}$:	n_2	(PRE.2)
$\mathrm{A} \to \mathrm{B}$:	$\mathsf{senc}(\mathsf{pk}(k), \mathsf{h}(\langle n_1, n_2 \rangle, w))$	(EKE'.1)
$\mathrm{B} \to \mathrm{A}$	$\mathsf{senc}(\mathsf{aenc}(r, \mathsf{pk}(k)), \mathsf{h}(\langle n_1, n_2 \rangle, w))$	(EKE'.2)
$\mathrm{A} \to \mathrm{B}$	$\mathsf{senc}(na, r)$	(EKE'.3)
$\mathrm{B} \to \mathrm{A}$	$\mathsf{senc}(\langle na, nb \rangle, r)$	(EKE'.4)
$\mathrm{A} \to \mathrm{B}$	$\mathsf{senc}(nb, r)$	(EKE'.5)

The first two messages (PRE.1) and (PRE.2) describe the preamble that establishes the tag $\langle n_1, n_2 \rangle$ used in the following messages. The formal description in our calculus of the transformed protocol is as follows.

$$
\begin{aligned}
A = \; & \nu k, na, n_1.\mathrm{out}(n_1).\ \mathrm{in}(x_1^2) \\
& \mathrm{ev}_{begin}(\mathsf{h}(\langle n_1, x_1^2 \rangle, w), na, \mathsf{pk}(k)). \\
& \mathrm{out}(\mathsf{senc}(\mathsf{pk}(k), \mathsf{h}(\langle n_1, x_1^2 \rangle, w))). \\
& \mathrm{in}(x_1). \\
& \mathrm{let}\ ra = \mathsf{adec}(\mathsf{sdec}(x_1, \mathsf{h}(\langle n_1, x_1^2 \rangle, w)), k). \\
& \mathrm{out}(\mathsf{senc}(na, ra)) \\
& \mathrm{in}(x_2). \\
& \mathrm{if}\ \mathsf{proj}_1(\mathsf{sdec}(x_2, ra)) = na\ \mathrm{then} \\
& \mathrm{out}(\mathsf{sdec}(\mathsf{proj}_2(\mathsf{sdec}(x_2, ra)), ra)).0
\end{aligned}
$$

$$
\begin{aligned}
B = \; & \nu r, nb, n_2.\ \mathrm{in}(x_2^1).\ \mathrm{out}(n_2) \\
& \mathrm{in}(y_1). \\
& \mathrm{out}(\mathsf{senc}(\mathsf{aenc}(r, \mathsf{sdec}(y_1, \mathsf{h}(\langle x_2^1, n_2 \rangle, w))), \mathsf{h}(\langle x_2^1, n_2 \rangle, w))). \\
& \mathrm{in}(y_2). \\
& \mathrm{out}(\mathsf{senc}(\langle \mathsf{sdec}(y_2, r), nb \rangle, r)). \\
& \mathrm{in}(y_3) \\
& \mathrm{if}\ \mathsf{sdec}(y_3, r) = nb\ \mathrm{then} \\
& \mathrm{ev}_{end}(\mathsf{h}(\langle x_2^1, n_2 \rangle, w), \mathsf{sdec}(y_2, r), \mathsf{sdec}(y_1, \mathsf{h}(\langle x_2^1, n_2 \rangle, w))).0
\end{aligned}
$$

7.2 Composition Results

We can now state our composition results for sessions of a same protocol.

Theorem 3 *Let $\mathcal{P} = \nu w.(\nu \tilde{m}_1.P_1 \mid \ldots \mid \nu \tilde{m}_\ell.P_\ell)$ be a password protocol specification and $\mathcal{P}'$ be such that $\overline{\mathcal{P}} = \nu w.\mathcal{P}'$, and $\mathcal{P}'_1, \ldots \mathcal{P}'_p$ be p instances of $\mathcal{P}'$.*

1. *Let t be a ground term that occurs as a subterm in $\mathcal{P}'_i$ for some $i \in \{1, \ldots, p\}$. If $\nu w.\mathcal{P}'_i$ preserves secrecy of t, then we have that $\nu w.(\mathcal{P}'_1 \mid \ldots \mid \mathcal{P}'_p)$ preserves secrecy of $t\{^{\mathsf{h}(t_i, w)}/_w\}$.*
2. *Let $\Phi = ev(\tilde{x}) \Rightarrow_{(\mathsf{inj})} ev(\tilde{x})$ be a correspondence property (injective or not). If Φ holds on $\mathcal{P}$, then Φ holds on $\nu w.(\mathcal{P}'_1 \mid \ldots \mid \mathcal{P}'_p)$.*
3. *If $\mathcal{P}$ is resistant to guessing attacks against w, then we have that $\nu w.(\mathcal{P}'_1 \mid \ldots \mid \mathcal{P}'_p)$ is resistant to guessing attacks against w.*

Proof (sketch) We here give an overview of the proof. A more detailed proof is given in Appendix C.

Assume, by contradiction, that $P = \nu w.(\mathcal{P}'_1 \mid \ldots \mid \mathcal{P}'_p)$ admits an attack on either of the three security properties. Hence there exists an attack derivation $P \rightarrow^* Q$ for some process Q such that the security property, either secrecy, a correspondence property or resistance to guessing attacks, fails. We are going to show that this attack also applies to $\mathcal{P}$ contradicting the hypothesis.

Step 1. We will first regroup the different roles of the protocol instances according to their tag. Thanks to our transformation, we know that each role involved in P has to execute its preamble, *i.e.*, the preliminary nonce exchange of our transformation, at the end of which it computes a tag. Let $t_1, \ldots, t_k$ be the distinct tags that are computed during this derivation. Then, we group together roles (*i.e.* closed plain processes) that computed the same tag in order to retrieve a situation that is similar to when we use static tags. We note that the tags are constructed such that each group contains at most one instance of each role of $\overline{\mathcal{P}}$. Our aim is to show that an attack already exists on one of these groups, and so the attack is not due to composition. However, one difficulty comes from the fact that once the preambles have been executed, the tags that have been computed by the different roles may share some names in addition to w.

Step 2. The aim of this step is to show that an attack on a transformed protocol also exists on a protocol that is tagged with constants (instead of the constructed tag) and different passwords (instead of the same password).

The fact that some names are shared between the processes we would like to separate in order to retrieve the disjoint case significantly complicates the situation. Indeed, if composition still works, it is due to the fact that names shared among differently tagged processes only occur at particular positions. To get rid of shared names, we show that we can mimic a derivation by another derivation where tags $t_1, \ldots, t_k$ are replaced by constants $c_1, \ldots, c_k$ and different password are used ($w_1, \ldots, w_k$ instead of w).

Using Proposition 4 we can map an execution of

$$P \equiv \nu n_1 \ldots \nu n_k \nu w.(A_1\delta_{c_i,t_i}\delta_{w_i,w} \mid \cdots \mid A_k\delta_{c_i,t_i}\delta_{w_i,w}) \text{ (same password)}$$

to an execution of

$$\nu n_1 \nu w_1.A_1 \mid \cdots \mid \nu n_k \nu w_k.A_k \text{ (different password)}$$

by maintaining a strong connection between these two derivations where the process $A_j\delta_{c_i,t_i}\delta_{w_i,w}$ contains the roles in P that computed the tag t_j in the attack derivation. Exactly as in the proof of Theorem 2, using Lemmas 6, 7 and 8 we show that the derivation with constant tags and different passwords also admits an attack.

Note that, except for w, a name that is shared between the processes $A_j\delta_{c_i,t_i}\delta_{w_i,w}$ and $A_{j'}\delta_{c_i,t_i}\delta_{w_i,w}$ ($j \neq j'$) necessarily occurs in a tag position in one of the processes. Now that tags have been replaced by some constants, and the password w has been replaced by different passwords according to the tag, the processes A_j and $A_{j'}$ do not share any name anymore.

Step 3. Applying Proposition 1, we conclude that there is a guessing attack on $\nu n_i.\nu w_i.A_i$ for some $i \in \{1, \ldots, k\}$. Then, it remains to show that the attack also works on the original protocol, *i.e.* the non-tagged version of the protocol. This is a direct application of Theorem 1. This leads us to a contradiction since we have assumed that $\mathcal{P}$ does not admit an attack.

8 Discussion

One may note that all our composition results hold for an unbounded number of sessions (even though our protocol language does not include replication). This is because our proofs proceed by contradiction and the fact that any attack only uses only a finite number of sessions. Indeed, for instance, suppose that two protocols are separately resistant against guessing attacks for an unbounded number of sessions and that their parallel composition allows a guessing attack. As any attack only requires a finite number of sessions, by Theorem 2, we have that one of the protocols admits an attack leading to a contradiction. The same reasoning can be done for the other security properties and also when we compose several sessions of the same protocol.

We also note that it is possible to combine the two ways of tagging that we proposed. Applying successively Theorems 2 and 3 we obtain that a tag of the form $\mathsf{h}(\langle n_1, \ldots, n_\ell \rangle, \mathsf{h}(c, w))$ allows to safely compose different sessions of a same protocol, and also sessions of other protocols. It would also be easy to adapt the proofs to directly show that a simpler tag of the form $\mathsf{h}(\langle c, \langle n_1, \ldots, n_\ell \rangle \rangle, w)$ could be used.

Finally, we note that our composition result yields a simple design methodology. It is sufficient to design a protocol which is secure for a single session. After applying the above protocol transformation we conclude that the transformed protocol is secure for an arbitrary number of sessions. As deciding resistance to guessing attacks is decidable for a bounded number of sessions (for a large class of equational theories) [12] our result can also be seen as a new decidability result for an unbounded number of sessions on a class of tagged protocols.

9 Conclusion

In this paper, we examined whether resistance to offline guessing attacks "composes" when the same password is used in two different protocols. More precisely, when each of two protocols resists offline guessing attacks by itself and the same password is used in each of them, we study whether the combination also resists. In the case of a passive attacker, the answer is yes. In the case of an active attacker, the answer is no in general. We propose a means to transform the protocol so that we can obtain the secure composition.

The transformation we propose works whether the composition is of two different protocols, or two sessions of the same protocol. Moreover, the transformation preserves other desirable trace-based properties that the protocols are intended to guarantee, such as authentication and secrecy.

An alternative direction of research would be to investigate whether there are conditions on the equational theory and a suitable condition of executability that would make the composition result hold without tagging for the active case. In particular we do not have an individually-executable counterexample for the common equational theory $\mathsf{E}_{\mathsf{enc}}$ given in Example 1. It would also be interesting to consider the case where additional long term keys are shared. Broader directions for future research include composition of other security properties, such as observational equivalence for processes that share secrets, and different composition operators, e.g. sequential composition.

Acknowledgments. Our paper benefited from comments and discussions with Véronique Cortier, Cédric Fournet and Bogdan Warinschi.

References

1. M. Abadi, M. Baudet, and B. Warinschi. Guessing attacks and the computational soundness of static equivalence. In L. Aceto and A. Ingólfsdóttir, editors, *Proc. 9th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'06)*, volume 3921 of *Lecture Notes in Computer Science*, pages 398–412. Springer, Mar. 2006.
2. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In H. R. Nielson, editor, *Proc. 28th Symposium on Principles of Programming Languages (POPL'01)*, pages 104–115. ACM Press, 2001.
3. M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *Proc. 4th Conference on Computer and Communications Security (CCS'97)*, pages 36–47. ACM Press, 1997.
4. M. Abdalla, C. Chevalier, L. Granboulan, and D. Pointcheval. UC-secure group key exchange with password-based authentication in the standard model. In *Proc. The Cryptographers' Track at the RSA Conference (CT-RSA'11)*, volume 6558 of *Lecture Notes in Computer Science*, pages 142–160. Springer, 2011.
5. M. Abdalla, C. Chevalier, and D. Pointcheval. Smooth projective hashing for conditionally extractable commitments. In *Advances in Cryptology – CRYPTO'09*, volume 5677 of *Lecture Notes in Computer Science*, pages 671–689. Springer, 2009.
6. S. Andova, C. J. F. Cremers, K. Gjøsteen, S. Mauw, S. F. Mjølsnes, and S. Radomirovic. A framework for compositional verification of security protocols. *Inf. Comput.*, 206(2-4):425–459, 2008.
7. M. Arapinis, S. Delaune, and S. Kremer. From one session to many: Dynamic tags for security protocols. In *Proc. 15th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'08)*, volume 5330 of *Lecture Notes in Artificial Intelligence*, pages 128–142. Springer, 2008.
8. M. Arapinis and M. Duflot. Bounding messages for free in security protocols. In *Proc. 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'07)*, volume 4855 of *Lecture Notes in Computer Science*, pages 376–387. Springer, 2007.
9. A. Armando, D. A. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron. The Avispa tool for the automated validation of internet security protocols and applications. In *Proc. 17th International Conference on Computer Aided Verification (CAV'05)*, volume 3576 of *Lecture Notes in Computer Science*, pages 281–285, 2005.
10. B. Barak, Y. Lindell, and T. Rabin. Protocol initialization for the framework of universal composability. Cryptology ePrint Archive, Report 2004/006, 2004. `http://eprint.iacr.org/`.
11. M. Baudet. YAPA. `http://www.lsv.ens-cachan.fr/~baudet/yapa/`.
12. M. Baudet. Deciding security of protocols against off-line guessing attacks. In *Proc. 12th ACM Conference on Computer and Communications Security (CCS'05)*, pages 16–25. ACM Press, Nov. 2005.
13. M. Baudet. *Sécurité des protocoles cryptographiques : aspects logiques et calculatoires*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, Jan. 2007.
14. S. M. Bellovin and M. Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *Proc. Symposium on Security and Privacy (SP'92)*, pages 72–84. IEEE Comp. Soc., 1992.
15. B. Blanchet. An Efficient Cryptographic Protocol Verifier based on Prolog Rules. In *Proc. 14th Computer Security Foundations Workshop (CSFW'01)*, pages 82–96. IEEE Comp. Soc., June 2001.
16. B. Blanchet. Automatic Proof of Strong Secrecy for Security Protocols. In *Proc. Symposium on Security and Privacy (SP'04)*, pages 86–100. IEEE Comp. Soc., May 2004.

17. B. Blanchet and A. Podelski. Verification of cryptographic protocols: Tagging enforces termination. In *Proc. Foundations of Software Science and Computation Structures (FoSSaCS'03)*, volume 2620 of *Lecture Notes in Computer Science*, pages 136–152. Springer, 2003.
18. X. Boyen, C. Chevalier, G. Fuchsbauer, and D. Pointcheval. Strong cryptography from weak secrets: Building efficient PKE and IBE from distributed passwords in bilinear groups. In *Progress in Cryptology – AFRICACRYPT'10*, volume 6055 of *Lecture Notes in Computer Science*, pages 297–315. Springer, 2010.
19. V. Boyko, P. D. MacKenzie, and S. Patel. Provably secure password-authenticated key exchange using Diffie-Hellman. In *Advances in Cryptology – EUROCRYPT'00*, volume 1807 of *Lecture Notes in Computer Science*, pages 156–171. Springer, 2000.
20. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proc. 42nd Annual Symposium on Foundations of Computer Science (FOCS'01)*, pages 136–145. IEEE Comp. Soc., 2001.
21. R. Canetti, S. Halevi, J. Katz, Y. Lindell, and P. D. MacKenzie. Universally composable password-based key exchange. In *Advances in Cryptology – EUROCRYPT'05*, volume 3494 of *Lecture Notes in Computer Science*, pages 404–421. Springer, 2005.
22. E. Cohen. Proving cryptographic protocols safe from guessing attacks. In *Proc. Foundations of Computer Security (FCS'02)*, 2002.
23. R. Corin, J. Doumen, and S. Etalle. Analysing password protocol security against off-line dictionary attacks. *ENTCS*, 121:47–63, 2005.
24. R. Corin, S. Malladi, J. Alves-Foss, and S. Etalle. Guess what? Here is a new tool that finds some new guessing attacks. In *Proc. of the Workshop on Issues in the Theory of Security (WITS'03)*, 2003.
25. V. Cortier, J. Delaitre, and S. Delaune. Safely composing security protocols. In V. Arvind and S. Prasad, editors, *Proc. 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07)*, Lecture Notes in Computer Science. Springer, Dec. 2007.
26. V. Cortier and S. Delaune. Safely composing security protocols. *Formal Methods in System Design*, 34(1):1–36, Feb. 2009.
27. A. Datta, A. Derek, J. Mitchell, and D. Pavlovic. A derivation system and compositional logic for security protocols. *Journal of Computer Security*, 13(3):423–482, 2005.
28. S. Delaune and F. Jacquemard. Decision procedures for the security of protocols with probabilistic encryption against offline dictionary attacks. *Journal of Automated Reasoning*, 36(1-2):85–124, Jan. 2006.
29. J. D. Guttman and F. J. Thayer. Protocol independence through disjoint encryption. In *Proc. 13th Computer Security Foundations Workshop (CSFW'00)*, pages 24–34. IEEE Comp. Soc., 2000.
30. F. Hao and P. Y. A. Ryan. Password authenticated key exchange by juggling. In *Proc. 16th International Security Protocols Workshop*, volume 6615 of *Lecture Notes in Computer Science*, pages 159–171. Springer, 2008.
31. F. Hao and P. Y. A. Ryan. How to sync with alice. In *19th International Security Protocols Workshop*, volume 7114 of *Lecture Notes in Computer Science*, pages 170–178. Springer, 2011.
32. D. Jablon. Strong password-only authenticated key exchange. *Computer Communication Review*, 26(5):5–26, 1996.
33. J. Katz, R. Ostrovsky, and M. Yung. Efficient password-authenticated key exchange using human-memorable passwords. In *Advances in Cryptology – EUROCRYPT'01*, volume 2045 of *Lecture Notes in Computer Science*, pages 475–494. Springer, 2001.
34. G. Lowe. Analysing protocols subject to guessing attacks. *Journal of Computer Security*, 12(1):83–98, 2004.
35. S. Malladi, J. Alves-Foss, and S. Malladi. What are multi-protocol guessing attacks and how to prevent them. In *Proc. 11th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2002)*, pages 77–82. IEEE Comp. Soc., 2002.
36. S. Patel. Number theoretic attacks on secure password schemes. In *Proc. IEEE Symposium on Security and Privacy (S&P'97)*, pages 236–247. IEEE Computer Society, 1997.
37. R. Ramanujam and S. P. Suresh. Decidability of context-explicit security protocols. *Journal of Computer Security*, 13(1):135–165, 2005.
38. Trusted Computing Group. TPM Specification version 1.2. Parts 1–3, revision 103. `http://www.trustedcomputinggroup.org/resources/tpm_main_specification`, 2007.

A Disjoint case

To establish this proposition, we first prove some lemmas about deduction and static equivalence.

Lemma 9 *Let $\phi \equiv \nu\tilde{n}.\sigma$ be a frame, t be a ground term that is not deducible from ϕ, M be a ground term deducible from ϕ, y be a variable not in $\mathrm{dom}(\phi)$, and m be a name not in $bn(\phi)$. Then, we have that t is neither deducible from $\nu m.\phi$, nor from $\nu\tilde{n}.(\sigma \mid \{^M/_y\})$.*

Proof We prove the two points separately.

We have that t is not deducible from the frame $\nu m.\phi$. We prove this result by contradiction. Assume that it is not the case. This means that there exists U such that $fn(U) \cap \tilde{n} = \emptyset$, $m \notin fn(U)$, and $U\sigma =_{\mathsf{E}} t$. We easily deduce that U is also a recipe for t w.r.t. the frame ϕ, contradiction.

We have that t is not deducible from the frame $\nu\tilde{n}.(\sigma \mid \{^M/_y\})$. Let ζ be a recipe of M, i.e. a term such that $fn(\zeta) \cap \tilde{n} = \emptyset$, $w \notin fn(\zeta)$, and $\zeta\sigma =_{\mathsf{E}} M$. We now prove the result by contradiction. Assume that t is deducible from the frame $\nu\tilde{n}.(\sigma \mid \{^M/_y\})$. This means that there exists U such that $fn(U) \cap \tilde{n} = \emptyset$, and $U(\sigma \mid \{^M/_y\}) =_{\mathsf{E}} t$. Let $U' = U\{^\zeta/_y\}$. We have that $fn(U') \cap \tilde{n} = \emptyset$, and $U'\sigma = (U\{^\zeta/_y\})\sigma =_{\mathsf{E}} (U\{^M/_y\})\sigma = U(\sigma \mid \{^M/_y\}) =_{\mathsf{E}} t$. Thus, t is deducible from $\nu\tilde{n}.\sigma$ using the recipe U', contradiction. □

Lemma 10 *Let $\phi \equiv \nu w.\nu\tilde{n}.\sigma$ be a frame resistant to guessing attacks against w, M be a ground term deducible from ϕ, y be a variable not in $\mathrm{dom}(\phi)$, and m be a name not in $bn(\phi)$. Then we have that the frames $\nu m.\phi$ and $\nu w.\nu\tilde{n}.(\sigma \mid \{^M/_y\})$ are resistant to guessing attacks against w.*

Proof We prove the two points separately.

The frame $\nu m.\phi$ is resistant to guessing attacks against w. We prove this result by contradiction. Assume that it is not the case. This means that

$$\nu w.\nu m.\nu\tilde{n}.(\sigma \mid \{^w/_x\}) \not\approx \nu w'.\nu w.\nu m.\nu\tilde{n}.(\sigma \mid \{^{w'}/_x\})$$

where w' is a fresh name, and x a variable that does not occur in $\mathrm{dom}(\sigma)$. By definition of $\approx$, this means that there exist M and N such that $(fn(M) \cup fn(N)) \cap \tilde{n} = \emptyset$, and $w, w', m \notin fn(M) \cup fn(N)$ with $(M\{^w/_x\} =_{\mathsf{E}} N\{^w/_x\})\sigma$ and $(M\{^{w'}/_x\} \neq_{\mathsf{E}} N\{^{w'}/_x\})\sigma$ (or conversely). Actually, the same test (M, N) can be used to show that ϕ is not resistant to guessing attacks against w.

The frame $\nu w.\nu\tilde{n}.(\sigma \mid \{^M/_y\})$ is resistant to guessing attacks against w. Let ζ be a recipe of M, i.e. a term such that $fn(\zeta) \cap \tilde{n} = \emptyset$, $w \notin fn(\zeta)$, $fv(\zeta) \subseteq \mathrm{dom}(\sigma)$, and $\zeta\sigma =_{\mathsf{E}} M$. Moreover, we assume that $w' \notin fn(\zeta)$. By hypothesis, we have that $\nu w.\nu\tilde{n}.(\sigma \mid \{^w/_x\}) \approx \nu w'.\nu w.\nu\tilde{n}.(\sigma \mid \{^{w'}/_x\})$ where w' is a fresh name and x a variable that does not occur in $\mathrm{dom}(\sigma)$. Our goal is to show that:

$$\nu w.\nu\tilde{n}.(\sigma \mid \{^M/_y\} \mid \{^w/_x\}) \approx \nu w'.\nu w.\nu\tilde{n}.(\sigma \mid \{^M/_y\} \mid \{^{w'}/_x\}).$$

Let U, V be two terms such that $(fn(U) \cup fn(V)) \cap \tilde{n} = \emptyset$, $w, w' \notin (fn(U) \cup fn(V))$, and $(U =_{\mathsf{E}} V)(\sigma \mid \{^M/_y\} \mid \{^w/_x\})$. Let $U' = U\{^\zeta/_y\}$ and $V' = V\{^\zeta/_y\}$. First, we have that $(fn(U') \cup fn(V')) \cap \tilde{n} = \emptyset$ and $w, w' \notin (fn(U') \cup fn(V'))$. Moreover, we have that:

- $U(\sigma \mid \{^M/_y\} \mid \{^w/_x\}) =_{\mathsf{E}} U'(\sigma \mid \{^w/_x\})$, and
- $V(\sigma \mid \{^M/_y\} \mid \{^w/_x\}) =_{\mathsf{E}} V'(\sigma \mid \{^w/_x\})$.

Thanks to our hypothesis, we deduce that $(U' =_{\mathsf{E}} V')(\sigma \mid \{^{w'}/_x\})$ and $(U\{^\zeta/_y\} =_{\mathsf{E}} V\{^\zeta/_y\})(\sigma \mid \{^{w'}/_x\})$, i.e. $(U =_{\mathsf{E}} V)(\sigma \mid \{^M/_y\} \mid \{^{w'}/_x\}$. The other direction can be shown in a similar way. □

Proposition 1 *Let $A_1, \ldots, A_k$ be k extended processes such that $A \stackrel{\mathsf{def}}{=} A_1 \mid \ldots \mid A_k$ is also an extended process, and $w_i \in bn(A_i)$ for each $i \in \{1, \ldots, k\}$.*

1. *Let t be a ground term that occurs as a subterm in A_i for some $i \in \{1, \ldots, k\}$. If A_i preserves secrecy of t, then A preserves secrecy of t.*
2. *Let $\Phi = ev(\tilde{x}) \Rightarrow_{(\mathsf{inj})} ev'(\tilde{x})$ be a correspondence property (injective or not). If Φ holds on each A_i, then Φ holds on A.*
3. *If each A_i is resistant to guessing attack against w_i, then A is resistant to guessing attack against $w_1, \ldots, w_k$.*

Proof We prove this composition result by contradiction. Assume that the process A admits an attack. Let $A_i \equiv \nu w_i.\nu\tilde{n}_i.P_i$ for each $\in \{1, \ldots, k\}$, $\tilde{w} = w_1, \ldots, w_k$, and $\tilde{n} = \tilde{n}_1, \ldots \tilde{n}_k$. By definition of an attack, we have that there exists a trace:

$$A \stackrel{\mathsf{def}}{=} A_1 \mid \ldots \mid A_k \xrightarrow{\ell_1} B_1 \ldots \xrightarrow{\ell_n} B_n$$

with $B_n = \nu\tilde{w}.\nu\tilde{n}.(P'_1 \mid \sigma_1 \mid \ldots \mid P'_k \mid \sigma_k)$. Intuitively, the active substitutions in σ_i comes from A_i and P'_i is the remaining part of P_i. In addition, depending on the security property under study, we have that:

1. *(secrecy)* We know that $\phi(B_n) \vdash_{\mathsf{E}} t$ for some t that occurs as a subterm of A_{i_0} with $i_0 \in \{1, \ldots, k\}$. Actually, since $A \rightarrow^* B_n$, we have also that $A_{i_0} \rightarrow^* \nu w_{i_0}.\nu\tilde{n}_{i_0}.(P'_{i_0} \mid \sigma_{i_0})$. Moreover, by hypothesis, we know that $\nu w_{i_0}.\nu\tilde{n}_{i_0}.\sigma_{i_0} \not\vdash_{\mathsf{E}} t$. Relying on Lemma 9, we deduce that t is not deducible from $\nu\tilde{w}.\nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k)$, i.e. $\phi(B_n) \not\vdash_{\mathsf{E}} t$, contradiction.
2. *(correspondence property)* there exists j_0 and a substitution σ such that $\ell_{j_0} =_{\mathsf{E}} ev(\tilde{x}\sigma)$ and $\ell_j \neq_{\mathsf{E}} ev'(\tilde{x}\sigma)$ for any $j \leq j_0$. Let $i_0 \in \{1, \ldots, k\}$ be such that the action ℓ_{j_0} has been performed by A_{i_0}. Actually, since $A \rightarrow^* B_n$ through the labels $\ell_1, \ldots, \ell_n$, we have also that $A_{i_0} \rightarrow^* \nu w_{i_0}.\nu\tilde{n}_{i_0}.(P'_{i_0} \mid \sigma_{i_0})$ using the labels $\ell_{j_1}, \ldots, \ell_{j_p}$ a subword of $\ell_1, \ldots, \ell_n$ (i.e. the sequence $\ell_1, \ldots, \ell_n$ can be obtained from $\ell_{j_1}, \ldots, \ell_{j_p}$ by inserting some element in it). Moreover, we have that ℓ_{j_0} occurs in $\ell_{j_1}, \ldots, \ell_{j_p}$. From this, it is now quite easy to see that Φ does not hold on A_i, contradiction.
   We consider now the case of an injective correspondance property. We know that there exist j_0 and σ such that:

   $$\#\{j \mid ev(x_1\sigma, \ldots, x_k\sigma) = \ell_j \text{ with } j \leq j_0\} > \#\{j \mid ev'(x_1\sigma, \ldots, x_k\sigma) = \ell_j \text{ with } j \leq j_0\}.$$

   In particular, this means that there exists $i_0 \in \{1, \ldots, k\}$ such that:

   $$\begin{array}{l}\#\{j \mid ev(x_1\sigma, \ldots, x_k\sigma) = \ell_j \text{ with } j \leq j_0 \text{ and } \ell_j \text{ is an action performed by } A_{i_0}\} \\ > \quad \#\{j \mid ev'(x_1\sigma, \ldots, x_k\sigma) = \ell_j \text{ with } j \leq j_0 \text{ and } \ell_j \text{ is an action performed by } A_{i_0}\}.\end{array}$$

   As before, we have that $A_{i_0} \rightarrow^* \nu w_{i_0}.\nu\tilde{n}_{i_0}.(P'_{i_0} \mid \sigma_{i_0})$ using the labels $\ell_{j_1}, \ldots, \ell_{j_p}$ (these labels correspond to the actions that are performed by A_{i_0} in the sequence $\ell_1, \ldots, \ell_n$). Using the relation given above, it is quite easy to see that Φ does not hold on A_{i_0}. This allows us to conclude.
3. *(guessing attack)* the frame $\phi(B_n)$ is not resistant to guessing attacks against $\tilde{w}$. Actually, since $A \rightarrow^* B_n$, we have also that $A_i \rightarrow^* \nu w_i.\nu\tilde{n}_i.(P'_i \mid \sigma_i)$ for each $i \in \{1, \ldots, k\}$. Moreover, by hypothesis, we know that $\nu w_i.\nu\tilde{n}_i.\sigma_i$ is resistant to guessing attacks against w_i. Relying on Lemma 10, we obtain the following equivalences:

   $$\begin{array}{c}\nu w_1.\nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k) \approx \nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k) \\ \nu w_2.\nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k) \approx \nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k) \\ \vdots \\ \nu w_k.\nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k) \approx \nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k)\end{array}$$

Applying Lemma 1 (*item 1*), we deduce that:

$$\begin{array}{c}\nu w_1.\nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k) \approx \nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k)\\ \nu w_1.\nu w_2.\nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k) \approx \nu w_1.\nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k)\\ \vdots\\ \nu w_1.\ldots.\nu w_k.\nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k) \approx \nu w_1.\ldots.\nu w_{k-1}.\nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k)\end{array}$$

By transitivity of $\approx$, we deduce that $\nu\tilde{w}.\nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k) \approx \nu\tilde{n}.(\sigma_1 \mid \ldots \mid \sigma_k)$. This means that $\phi(B_n)$ is not resistant to guessing attacks against $\tilde{w}$, contradiction. □

B Transformation

The goal of this section is to prove Theorem 1.

B.1 Proof of Lemma 5

Before to prove Lemma 5, we introduce the following cutting function.

Definition 12 Given a frame ϕ, a term $U = \mathsf{h}(U_1, U_2)$ and a name a, the cutting function cut w.r.t. ϕ, U and a is defined recursively as $\mathsf{cut}_\phi(u) = u$ when u is a name or a variable and:

$$\mathsf{cut}_\phi(f(T_1,\ldots,T_k)) = \begin{cases} a & \text{if } f = \mathsf{h},\ k = 2,\ (U_1 =_\mathsf{E} T_1)\phi \text{ and } (U_2 =_\mathsf{E} T_2)\phi \\ f(\mathsf{cut}_\phi(T_1),\ldots,\mathsf{cut}_\phi(T_k)) & \text{otherwise}\end{cases}$$

When $\mathrm{dom}(\phi) = \emptyset$, we denote it at cut_0. In this case, the function cut_0 is a replacement modulo E as defined in [13]. Hence, we have the following lemma.

Lemma 11 *Let $U = \mathsf{h}(U_1, U_2)$ be a term and a be a name. We have that:*

$$M =_\mathsf{E} N \ \Rightarrow\ \mathsf{cut}_0(M) =_\mathsf{E} \mathsf{cut}_0(N) \text{ for any term } M \text{ and } N.$$

Lemma 12 *Let $\phi =_\alpha \nu\tilde{n}.\sigma$ be a frame. Let $w, \overline{w}$ and c be three names such that $w, c \notin \tilde{n}$ and $\overline{w}$ is a fresh name. Let cut be the cutting function w.r.t. $\phi\{^{\mathsf{h}(c,\overline{w})}/_w\}$, $\mathsf{h}(c,\overline{w})$, w and cut_0 be the cutting function w.r.t. $\mathsf{h}(c,\overline{w})$ and w. Let M be a term such that $fn(M) \cap \tilde{n} = \emptyset$. We have that*

$$\mathsf{cut}_0(M(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\})) = \mathsf{cut}(M)\sigma.$$

Proof We prove this result by structural induction on M. If M is a name or a variable such that $M \notin \mathrm{dom}(\phi)$, we have that

$$\mathsf{cut}_0(M(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\})) = \mathsf{cut}(M)\sigma = M.$$

Now, assume that M is a variable, say x, such that $x \in \mathrm{dom}(\phi)$. Let $T = x\sigma$. Note that $\overline{w}$ does not occur in T since $\overline{w}$ is fresh w.r.t. σ. Hence, we have that[2]:

$$\mathsf{cut}_0(M(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\})) = \mathsf{cut}_0(T\{^{\mathsf{h}(c,\overline{w})}/_w\}) = T = x\sigma = \mathsf{cut}(M)\sigma.$$

Now, we deal with the induction step: $M = f(M_1,\ldots,M_k)$. We distinguish two cases:

1. $f = \mathsf{h}$, $k = 2$, $(M_1 =_\mathsf{E} c)(\phi\{^{\mathsf{h}(c,\overline{w})}/_w\})$ and $(M_2 =_\mathsf{E} \overline{w})(\phi\{^{\mathsf{h}(c,\overline{w})}/_w\})$. In such a case, we have that $\mathsf{cut}(M)\sigma = w$. Moreover, we have also that $M_1\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\} =_\mathsf{E} c$ and $M_2\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\} =_\mathsf{E} \overline{w}$. Hence, we have that

$$\mathsf{cut}_0(M(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\})) = \mathsf{cut}_0(\mathsf{h}(M_1(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\}), M_2(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\}))) = w.$$

[2] The second step can be easily shown by structural induction on T.

2. Otherwise, we have that $\mathsf{cut}(f(M_1,\ldots,M_k)) = f(\mathsf{cut}(M_1),\ldots,\mathsf{cut}(M_k))$. Hence, we have that $\mathsf{cut}_0(M(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\})) = f(\mathsf{cut}_0(M_1(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\})),\ldots,\mathsf{cut}_0(M_k(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\})))$. Indeed, otherwise we will have that $f = \mathsf{h}$, $(M_1 =_{\mathsf{E}} c)(\phi\{^{\mathsf{h}(c,\overline{w})}/_w\})$ and also that $(M_2 =_{\mathsf{E}} \overline{w})(\phi\{^{\mathsf{h}(c,\overline{w})}/_w\})$. This situation corresponds to our first case. Hence, we have that

$$\begin{array}{ll}
& \mathsf{cut}_0(M(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\})) \\
= & f(\mathsf{cut}_0(M_1(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\})),\ldots,\mathsf{cut}_0(M_k(\sigma\{^{\mathsf{h}(c,\overline{w})}/_w\}))) \\
= & f(\mathsf{cut}(M_1)\sigma,\ldots,\mathsf{cut}(M_k)\sigma) \qquad \text{by induction hypothesis} \\
= & f(\mathsf{cut}(M_1),\ldots,\mathsf{cut}(M_k))\sigma \\
= & \mathsf{cut}(M)\sigma
\end{array}$$

This allows us to conclude the proof. □

Lemma 5 *Let ϕ_1 and ϕ_2 be two frames such that $\phi_1 \approx \phi_2$. Let w, c be such that $w, c \notin bn(\phi_1) \cup bn(\phi_2)$. We have that*

$$\phi_1\{^{\mathsf{h}(c,w)}/_w\} \approx \phi_2\{^{\mathsf{h}(c,w)}/_w\}.$$

Proof We will show that $\phi_1\{^{\mathsf{h}(c,\overline{w})}/_w\} \approx \phi_2\{^{\mathsf{h}(c,\overline{w})}/_w\}$ for some fresh names $\overline{w}$. This will allow us to conclude that $\phi_1\{^{\mathsf{h}(c,w}/_w\} \approx \phi_2\{^{\mathsf{h}(c,w)}/_w\}$ by simply renaming $\overline{w}$ with w. For this we have to show that for all terms M and N, we have that: $(M =_{\mathsf{E}} N)\phi_1\{^{\mathsf{h}(c,\overline{w})}/_w\} \Rightarrow (M =_{\mathsf{E}} N)\phi_2\{^{\mathsf{h}(c,\overline{w})}/_w\}$ (and conversely). Actually, the 2nd implication can be proved in a similar way, so we will focuss on the first one.

Actually, it is sufficient to establish this result for all terms M and N such that $w \notin fn(M) \cup fn(N)$ since w does not occur in $\phi_1\{^{\mathsf{h}(c,\overline{w})}/_w\}$ and $\phi_2\{^{\mathsf{h}(c,\overline{w})}/_w\}$. Let σ_1 and σ_2 be two substitutions such that $\phi_1 =_\alpha \nu\tilde{n}_1.\sigma_1$ and $\phi_2 =_\alpha \nu\tilde{n}_2.\sigma_2$ for some sequences of names $\tilde{n}_1$ and $\tilde{n}_2$ such that $(fn(M) \cup fn(N)) \cap (\tilde{n}_1 \cup \tilde{n}_2) = \emptyset$. Moreover, we can assume that $w, \overline{w}, c \notin \tilde{n}_1 \cup \tilde{n}_2$. Hence, we have that $\phi_1\{^{\mathsf{h}(c,\overline{w})}/_w\} =_\alpha \nu\tilde{n}_1.\sigma_1\{^{\mathsf{h}(c,\overline{w})}/_w\}$, and $\phi_2\{^{\mathsf{h}(c,\overline{w})}/_w\} =_\alpha \nu\tilde{n}_2.\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\}$.

Let cut be the cutting function w.r.t. $\phi_1\{^{\mathsf{h}(c,\overline{w})}/_w\}$, $\mathsf{h}(c,\overline{w})$ and w, and cut_0 be the cutting function w.r.t. $\mathsf{h}(c,\overline{w})$ and w. We show by induction on $\max(|M|,|N|)$[3] that

1. $(\mathsf{cut}(M)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} =_{\mathsf{E}} M(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\})$, and
2. $(M =_{\mathsf{E}} N)(\phi_1\{^{\mathsf{h}(c,\overline{w})}/_w\}) \Rightarrow (M =_{\mathsf{E}} N)(\phi_2\{^{\mathsf{h}(c,\overline{w})}/_w\})$.

Base case: $\max(|M|,|N|) = 1$

1. If M is a name (note that $M \neq w$) or a variable such that $M \notin \mathrm{dom}(\phi_2)$, we have that $(\mathsf{cut}(M)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} = M$ and $M(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\}) = M$. If M is a variable, say x, such that $x \in \mathrm{dom}(\phi_2)$, then we have that

$$(\mathsf{cut}(M)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} = (x\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} = x(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\}) = M(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\}).$$

2. The second point can be proved as follows:

$$\begin{array}{lll}
& (M =_{\mathsf{E}} N)(\phi_1\{^{\mathsf{h}(c,\overline{w})}/_w\}) & \\
\Rightarrow & M(\sigma_1\{^{\mathsf{h}(c,\overline{w})}/_w\}) =_{\mathsf{E}} N(\sigma_1\{^{\mathsf{h}(c,\overline{w})}/_w\}) & \\
\Rightarrow & \mathsf{cut}_0(M(\sigma_1\{^{\mathsf{h}(c,\overline{w})}/_w\})) =_{\mathsf{E}} \mathsf{cut}_0(N(\sigma_1\{^{\mathsf{h}(c,\overline{w})}/_w\})) & \text{by Lemma 11} \\
\Rightarrow & \mathsf{cut}(M)\sigma_1 =_{\mathsf{E}} \mathsf{cut}(N)\sigma_1 & \text{by Lemma 12} \\
\Rightarrow & (\mathsf{cut}(M) =_{\mathsf{E}} \mathsf{cut}(N))\phi_1 & \\
\Rightarrow & (\mathsf{cut}(M) =_{\mathsf{E}} \mathsf{cut}(N))\phi_2 & \text{since } \phi_1 \approx \phi_2 \\
\Rightarrow & \mathsf{cut}(M)\sigma_2 =_{\mathsf{E}} \mathsf{cut}(N)\sigma_2 & \\
\Rightarrow & (\mathsf{cut}(M)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} =_{\mathsf{E}} (\mathsf{cut}(N)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} &
\end{array}$$

[3] The size $|M|$ of a term M is defined by $|u| = 1$ when u is a name or a variable and $|f(M_1,\ldots,M_k)| = 1 + \sum_{i=1}^{k} |M_i|$.

The last step comes from the fact that $=_{\mathsf{E}}$ is closed by substitutions of terms for names. Since, $|M| = |N| = 1$, we can apply our previous result to obtain that:
$(\mathsf{cut}(M)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} =_{\mathsf{E}} M(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\})$, $(\mathsf{cut}(N)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} =_{\mathsf{E}} N(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\})$.
We have that $M(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\}) =_{\mathsf{E}} N(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\})$, thus $(M =_{\mathsf{E}} N)(\phi_2\{^{\mathsf{h}(c,\overline{w})}/_w\})$.

Induction step: $\max(|M|, |N|) \geq 2$. We assume w.l.o.g. that $|M| \geq |N|$, so $M = f(M_1, \ldots, M_k)$.

1. To establish the first point, we distinguish two cases:
   - $f = \mathsf{h}$, $k = 2$, $(M_1 =_{\mathsf{E}} c)(\phi_1\{^{\mathsf{h}(c,\overline{w})}/_w\})$ and $(M_2 =_{\mathsf{E}} \overline{w})(\phi_1\{^{\mathsf{h}(c,\overline{w})}/_w\})$. In such a case, we have that $\mathsf{cut}(M) = w$, thus $(\mathsf{cut}(M)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} = \mathsf{h}(c, \overline{w})$. Since $|M_1| + |c| < |M| + |N|$ and $|M_2| + |\overline{w}| < |M| + |N|$, we have that
   $$(M_1 =_{\mathsf{E}} c)(\phi_2\{^{\mathsf{h}(c,\overline{w})}/_w\}) \text{ and } (M_2 =_{\mathsf{E}} \overline{w})(\phi_2\{^{\mathsf{h}(c,\overline{w})}/_w\})$$
   Hence, we have that
   $$M(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\}) = \mathsf{h}(M_1(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\}), M_2(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\})) =_{\mathsf{E}} \mathsf{h}(c, \overline{w})$$
   - Otherwise, we have that $\mathsf{cut}(M) = f(\mathsf{cut}(M_1), \ldots, \mathsf{cut}(M_k))$. Thus,
   $$\begin{array}{ll} & (\mathsf{cut}(M)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} \\ = & (f(\mathsf{cut}(M_1), \ldots, \mathsf{cut}(M_k))\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} \\ = & f((\mathsf{cut}(M_1)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\}, \ldots, (\mathsf{cut}(M_k)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\}) \\ =_{\mathsf{E}} & f(M_1(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\}), \ldots, M_k(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\})) \quad \text{by induction hypothesis} \\ = & f(M_1, \ldots, M_k)(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\}) \\ = & M(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\}) \end{array}$$

2. To prove the second point, it is easy to establish (as in the base case) that
   $$(M =_{\mathsf{E}} N)(\phi_1\{^{\mathsf{h}(c,\overline{w})}/_w\}) \Rightarrow (\mathsf{cut}(M)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} =_{\mathsf{E}} (\mathsf{cut}(N)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\}$$
   Thanks to our previous result, we have that $(\mathsf{cut}(M)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} =_{\mathsf{E}} M(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\})$, and $(\mathsf{cut}(N)\sigma_2)\{^{\mathsf{h}(c,\overline{w})}/_w\} =_{\mathsf{E}} N(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\})$. We conclude that $M(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\}) =_{\mathsf{E}} N(\sigma_2\{^{\mathsf{h}(c,\overline{w})}/_w\})$, and thus $(M =_{\mathsf{E}} N)(\phi_2\{^{\mathsf{h}(c,\overline{w})}/_w\})$.

This allows us to conclude the proof. □

B.2 Proof of Proposition 3

The two following lemmas will be useful to deal with the cases of an input (Lemma 3) and a conditional (Lemma 4) in the proof of Proposition 3.

Lemma 3 *Let ϕ be a frame such that $c, w \notin bn(\phi)$ and $\phi'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} \phi$ for some ϕ'. If $\nu w.\phi \vdash_{\mathsf{E}} M$ then there exists M' such that $M'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} M$ and $\nu w.\phi' \vdash_{\mathsf{E}} M'$.*

Proof Let $\phi = \nu\tilde{n}.\sigma$ and $\phi' = \nu\tilde{n}.\sigma'$ for some sequence of names $\tilde{n}$ and some substitutions σ and σ'. We have that $\sigma'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} \sigma$. Let M be such that $\nu w.\phi \vdash_{\mathsf{E}} M$, i.e. there exists ζ such that $fn(\zeta) \cap (\tilde{n} \cup \{w\}) = \emptyset$ and $\zeta\sigma =_{\mathsf{E}} M$. Let $M' = \zeta\sigma'$. We have that $\nu w.\phi' \vdash_{\mathsf{E}} M'$ and also that $M'\{^{\mathsf{h}(c,w)}/_w\} = (\zeta\sigma')\{^{\mathsf{h}(c,w)}/_w\} = \zeta(\sigma'\{^{\mathsf{h}(c,w)}/_w\}) =_{\mathsf{E}} \zeta\sigma =_{\mathsf{E}} M$. □

Lemma 4 *Let M, N, M' and N' be four terms such that $M =_{\mathsf{E}} M'\{^{\mathsf{h}(c,w)}/_w\}$ and $N =_{\mathsf{E}} N'\{^{\mathsf{h}(c,w)}/_w\}$. Then, we have that*

$$M =_{\mathsf{E}} N \text{ if, and only if, } M' =_{\mathsf{E}} N'$$

Proof As $=_{\mathsf{E}}$ is closed by substitutions of terms for names $M' =_{\mathsf{E}} N'$ implies $M =_{\mathsf{E}} N$. Now, let M and N be two terms such that $M =_{\mathsf{E}} N$. We have that $M'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} N'\{^{\mathsf{h}(c,w)}/_w\}$. Thus, according to Lemma 11, we have that

$$\mathsf{cut}_0(M'\{^{\mathsf{h}(c,w)}/_w\}) =_{\mathsf{E}} \mathsf{cut}_0(N'\{^{\mathsf{h}(c,w)}/_w\})$$

where cut_0 represents the cutting function w.r.t. $\mathsf{h}(c,w)$ and w. Now, it is easy to establish, by structural induction on M' that $\mathsf{cut}_0(M'\{^{\mathsf{h}(c,w)}/_w\}) = M'$. This allows us to conclude. □

We will prove Proposition 3 by induction on the prooftree witnessing the derivation. First, we establish a similar result for $\equiv$.

Lemma 13 *Let A be a process such that $w \notin bn(A)$ and $A'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} A$ for some A'. Suppose that $A \equiv B$ for some process B. Then $w \notin bn(B)$ and there exists a process B' such that $B'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} B$ and $A' \equiv B'$.*

Proof We prove this result by induction on the proof tree showing that $A \equiv B$. All the base cases are easy to prove. The only interesting inductive case is the case of an application of an evaluation context. Suppose that the proof tree showing that $A \equiv B$ ends with an instance of such a rule, i.e.

$$\frac{A_1 \equiv B_1}{C[A_1] \equiv C[B_1]}$$

where $A = C[A_1]$ and $B = C[B_1]$. By hypothesis, we know that there exists A' such that $A'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} C[A_1]$. Hence we have that $A' = C'[A'_1]$ where $C'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} C$ and $A'_1\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} A_1$ for some evaluation context C' and some process A'_1. Hence we can apply our induction hypothesis and we obtain that $w \notin bn(B_1)$ and there exists B'_1 such that $B'_1\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} B_1$, and $A'_1 \equiv B'_1$. We have that $w \notin bn(C[B_1])$. Let $B' = C'[B'_1]$. We have that $(C'[B'_1])\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} C[B_1] = B$ and $A' \equiv B'$. □

Now, we can prove the following proposition.

Proposition 3 *Let A be a process with $c, w \notin bn(A)$ and $A'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} A$ for some A'. If $\nu w.A \xrightarrow{\ell} \overline{B}$, then $\overline{B} \equiv \nu w.B$ and there exists a process B' and a label ℓ' such that $B'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} B$, $\ell'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} \ell$, and $\nu w.A' \xrightarrow{\ell'} \nu w.B'$.*

Proof We have that $\nu w.A \xrightarrow{\ell} \overline{B}$ and it is easy to see that $w \in bn(\overline{B})$. According to our calculus, we can always by using structural equivalence move a restriction in front of the process, thus we have that $\overline{B} \equiv \nu w.B$ for some process B. It is easy to see that $A \xrightarrow{\ell} B$ and when $\ell = in(M)$, we have that $\nu w.\phi(A) \vdash_{\mathsf{E}} M$. As $\nu w.\phi(A) \vdash_{\mathsf{E}} M$, by Lemma 3, we have that $\nu w.\phi(A') \vdash_{\mathsf{E}} M'$ for some M' such that $M'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} M$. This allows us to ensure that, in the case of an input, the side condition corresponding to an application of evaluation context is satisfied. Now, we show that there exists B' and ℓ' such that $B'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} B$, $\ell'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} \ell$, and $A' \to B'$ by induction on the proof tree showing that $A \xrightarrow{\ell} B$. This will allows us to conclude that $\nu w.A' \xrightarrow{\ell'} \nu w.B'$.

Base cases.

- In. In such a case, $A = \mathrm{in}(x).P$, $B = P\{^M/_x\}$. We have that $A' = \mathrm{in}(x).P'$ and $P'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} P$. Let $B' = P'\{^{M'}/_x\}$ and $\ell' = \mathrm{in}(M')$. We have that $\ell'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} \ell$, $B'\{^{\mathsf{h}(c,w)}/_w\} = (P'\{^{M'}/_x\})\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} P\{^M/_x\} = B$, and $A' \xrightarrow{\ell'} B'$.
- Out. We suppose that $A = \mathrm{out}(M).P$ and $B = P \mid \{^M/_x\}$. We have that $A' = \mathrm{out}(M').P'$ where $P'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} P$ and $M'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} M$. Let $B' = P' \mid \{^{M'}/_x\}$ and $\ell' = \mathrm{out}(M')$. We have $\ell'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} \ell$, $B'\{^{\mathsf{h}(c,w)}/_w\} = (P' \mid \{^{M'}/_x\})\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} B$, and $A' \xrightarrow{\ell'} B'$.

– Event. We suppose that $A = \mathrm{ev}(\tilde{M}).P$ and $B = P$. We have that $A' = \mathrm{ev}(\tilde{M}').P'$ where $P'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} P$ and $\tilde{M}'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} M$. Let $B' = P'$ and $\ell' = ev(\tilde{M}')$. We have $\ell'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} \ell$, $B'\{^{\mathsf{h}(c,w)}/_w\} = P'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} B$, and $A' \xrightarrow{\ell'} B'$.
– Then. We suppose that $A =$ "if $M_1 = M_2$ then P else Q" and $B = P$. By definition of $=_{\mathsf{E}}$ we have that $A' =$ "if $M_1' = M_2'$ then P' else Q'" where $P'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} P$, $Q'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} Q$ and $M_i'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} M_i$ $(i = 1, 2)$. Let $B' = P'$ and $\ell' = \tau$. As $M_1 =_{\mathsf{E}} M_2$, by Lemma 4 we have that $M_1' =_{\mathsf{E}} M_2'$. Hence, we indeed have that $\ell'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} \ell$, $B'\{^{\mathsf{h}(c,w)}/_w\} = P'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} P = B$, and $A' \to B'$.
– Else. This case is similar to the previous one.

Inductive cases. The inductive case corresponding to an application of structural equivalence directly follows from Lemma 13. Hence, it remains to show the case of an application of an evaluation context. Suppose that the proof $A \xrightarrow{\ell} B$ finishes by an application of the following rule

$$\frac{A_1 \xrightarrow{\ell} B_1}{C[A_1] \xrightarrow{\ell} C[B_1]}$$

where $A = C[A_1]$ and $B = C[B_1]$. By hypothesis, we know that there exists A' such that $A'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} A$. By definition of $=_{\mathsf{E}}$ we have that $A' = C'[A_1']$ where $C'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} C$ and $A_1'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} A_1$ for some evaluation context C' and some process A_1'. Hence we can apply our induction hypothesis to obtain that there exist B_1' and ℓ' such that $\ell'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} \ell$, $B_1'\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} B_1$, and $A_1' \xrightarrow{\ell'} B_1'$. Let $B' = C'[B_1']$. We have that $B'\{^{\mathsf{h}(c,w)}/_w\} = (C'[B_1'])\{^{\mathsf{h}(c,w)}/_w\} =_{\mathsf{E}} B$, and $A' \xrightarrow{\ell'} B'$. This last result is obtained by application of the evaluation context C' on $A_1' \xrightarrow{\ell'} B_1'$. □

C Composition

In this section we will use the following notations. Given terms $t_1, \ldots, t_k$ and distinct names $c_1, \ldots, c_k, w_1, \ldots, w_k$, and w that do not occur in $t_1, \ldots, t_k$, we denote by $\delta_{w_i,w}$ the replacement $\{^{w}/_{w_1}\} \ldots \{^{w}/_{w_k}\}$, by δ_{c_i,t_i} the replacement $\{^{t_1}/_{c_1}\} \ldots \{^{t_k}/_{c_k}\}$, and by $\delta_{w_i,\mathsf{h}(c_i,w_i)}$ the replacement $\{^{\mathsf{h}(c_1,w_1)}/_{w_1}\} \ldots \{^{\mathsf{h}(c_k,w_k)}/_{w_k}\}$

C.1 Proof of Lemma 8

Before proving Lemma 8, we introduce the following splitting functions.

Definition 13 Let $\psi = \nu\tilde{n}.\sigma$ be a frame such that $w \notin \tilde{n}$. Let $t_1, \ldots, t_k$ be distinct ground terms modulo E. Let $c_1, \ldots, c_k, w_1, \ldots, w_k$ be distinct fresh names.

Splitting function. Let M be a term such that $fn(M) \cap \tilde{n} = \emptyset$. The *splitting function* split_ψ w.r.t. ψ, w, $c_1, \ldots, c_k$, $w_1, \ldots, w_k, t_1, \ldots, t_k$ is defined recursively as $\mathsf{split}_\psi(M) = M$ when M is a name or a variable and $\mathsf{split}_\psi(\mathsf{f}(M_1, \ldots, M_\ell))$ is equal to:

– $\mathsf{h}(c_i, w_i)$ if $\mathsf{f} = \mathsf{h}$, $\ell = 2$, $M_1\sigma =_{\mathsf{E}} t_i$ and $M_2\sigma =_{\mathsf{E}} w$ with $1 \le i \le k$;
– $\mathsf{f}(\mathsf{split}_\psi(M_1), \ldots, \mathsf{split}_\psi(M_\ell))$ otherwise.

Ground splitting function. Let M be a term. The *ground splitting function* split_0 w.r.t. w, $c_1, \ldots, c_k$, $w_1, \ldots, w_k, t_1, \ldots, t_k$ is defined recursively as $\mathsf{split}_0(M) = M$ when M is a name or a variable and $\mathsf{split}_0(\mathsf{f}(M_1, \ldots, M_\ell))$ is equal to:

– $\mathsf{h}(c_i, w_i)$ if $\mathsf{f} = \mathsf{h}$, $\ell = 2$, $M_1 =_{\mathsf{E}} t_i$ and $M_2 =_{\mathsf{E}} w$ with $1 \le i \le k$;
– $\mathsf{f}(\mathsf{split}_0(M_1), \ldots, \mathsf{split}_0(M_\ell))$ otherwise.

As soon as $t_1, \ldots, t_k$ are distinct terms modulo E, the function split_0 is a replacement modulo E as defined in [13]. Hence, we have the following lemma.

Lemma 14 *Let* split_0 *be a ground splitting function as defined in Definition 13. Let* M *and* N *be two terms. We have that:*

$$M =_{\mathsf{E}} N \;\Rightarrow\; \mathsf{split}_0(M) =_{\mathsf{E}} \mathsf{split}_0(N)$$

Lemma 15 *Let* $t_1, \ldots, t_k$ *be distinct ground terms modulo* E *and* $c_1, \ldots, c_k, w_1, \ldots, w_k$ *be distinct fresh names, i.e., not occurring in* $fn(t_1, \ldots, t_k)$*. Let* $\phi = \nu\tilde{n}.\sigma$ *be a frame such that* $c_1, \ldots, c_k, w_1, \ldots, w_k, w \notin \tilde{n}$*,* $w \notin fn(\sigma)$*, and* $\sigma =_{\mathsf{E}} \sigma_0\delta_{w_i,\mathsf{h}(c_i,w_i)}$ *for some substitution* σ_0*. Let* split_ψ *(resp.* split_0*) be the splitting function (resp. ground splitting function) w.r.t.* $\psi = \nu\tilde{n}.(\sigma\delta_{c_i,t_i}\delta_{w_i,w})$*,* $w, c_1, \ldots, c_k, w_1, \ldots, w_k, t_1, \ldots, t_k$*. Let* M *be a term such that* $fn(M) \cap \tilde{n} = \emptyset$*. We have that:*

$$\mathsf{split}_0(M(\sigma\delta_{c_i,t_i}\delta_{w_i,w})) =_{\mathsf{E}} \mathsf{split}_\psi(M)\sigma.$$

Proof We prove this result by structural induction on M. If M is a name or a variable such that $M \notin \mathrm{dom}(\psi) = \mathrm{dom}(\sigma)$, the result trivially holds. Now, assume that M is a variable, say x, such that $x \in \mathrm{dom}(\psi)$ and let $T = x\sigma$. We have that $T =_{\mathsf{E}} T'\{^{\mathsf{h}(c_1,w_1)}/_{w_1}\} \ldots \{^{\mathsf{h}(c_k,w_k)}/_{w_k}\}$ for some T', and w does not occur in T. Hence, we have that:

$$\begin{array}{rcl} \mathsf{split}_0(x\psi) & = & \mathsf{split}_0(x(\sigma\delta_{c_i,t_i}\delta_{w_i,w})) \\ & =_{\mathsf{E}} & \mathsf{split}_0(T\delta_{c_i,t_i}\delta_{w_i,w}) \\ & =_{\mathsf{E}} & T \\ & = & \mathsf{split}_\psi(x)\sigma \end{array}$$

Now, we can deal with the induction step, i.e. $M = \mathsf{f}(M_1, \ldots, M_\ell)$. We distinguish two cases:

1. $\mathsf{f} = \mathsf{h}$, $\ell = 2$, $M_1(\sigma\delta_{c_i,t_i}\delta_{w_i,w}) =_{\mathsf{E}} t_{i_0}$, and $M_2(\sigma\delta_{c_i,t_i}\delta_{w_i,w}) =_{\mathsf{E}} w$ with $1 \le i_0 \le k$. In such a case, we have that $\mathsf{split}_\psi(M) = \mathsf{h}(c_{i_0}, w_{i_0})$, and

   $$M(\sigma\delta_{c_i,t_i}\delta_{w_i,w}) = \mathsf{h}(M_1(\sigma\delta_{c_i,t_i}\delta_{w_i,w}), M_2(\sigma\delta_{c_i,t_i}\delta_{w_i,w})) =_{\mathsf{E}} \mathsf{h}(t_{i_0}, w)$$

   Hence, we have that

   $$\begin{array}{rcl} \mathsf{split}_0(M(\sigma\delta_{c_i,t_i}\delta_{w_i,w})) & =_{\mathsf{E}} & \mathsf{split}_0(\mathsf{h}(t_{i_0}, w)) \\ & =_{\mathsf{E}} & \mathsf{h}(c_{i_0}, w_{i_0}) \\ & = & \mathsf{split}_\psi(M)\sigma \end{array}$$

2. Otherwise, we have that $\mathsf{split}_\psi(\mathsf{f}(M_1, \ldots, M_\ell)) = \mathsf{f}(\mathsf{split}_\psi(M_1), \ldots, \mathsf{split}_\psi(M_\ell))$, and thus we have also that:

   $$\mathsf{split}_0(M(\sigma\delta_{c_i,t_i}\delta_{w_i,w})) = \mathsf{f}(\mathsf{split}_0(M_1(\sigma\delta_{c_i,t_i}\delta_{w_i,w})), \ldots, \mathsf{split}_0(M_1(\sigma\delta_{c_i,t_i}\delta_{w_i,w}))).$$

   Hence, relying on our induction hypothesis, we have that:

   $$\begin{array}{rcl} \mathsf{split}_0(M(\sigma\delta_{c_i,t_i}\delta_{w_i,w})) & =_{\mathsf{E}} & \mathsf{f}(\mathsf{split}_\psi(M_1)\sigma, \ldots, \mathsf{split}_\psi(M_\ell)\sigma) \\ & = & \mathsf{split}_\psi(M)\sigma \end{array}$$

This allows us to conclude.

Lemma 8 *Let* $t_1, \ldots, t_k$ *be distinct ground terms modulo* E*. Let* $c_1, \ldots, c_k, w_1, \ldots, w_k$ *be distinct fresh names, and* $\phi = \nu\tilde{n}.\sigma$ *be a frame such that* $c_1, \ldots, c_k, w_1, \ldots, w_k \notin \tilde{n}$*, and* $\sigma =_{\mathsf{E}} \sigma_0\delta_{w_i,\mathsf{h}(c_i,w_i)}$ *for some substitution* σ_0*. Let* w *be a fresh name, and* $\psi = \nu\tilde{n}.(\sigma\delta_{c_i,t_i}\delta_{w_i,w})$*. For each* $1 \le i \le k$*, we also assume that* $\nu w.\psi \vdash_{\mathsf{E}} t_i$*.*

If $\nu\tilde{w}.\phi$ *is resistant to guessing attacks against* $\tilde{w} = \{w_1, \ldots, w_k\}$*, then* $\nu w.\psi$ *is resistant to guessing attacks against* w*.*

Proof To prove this, we have to establish that $\psi \approx \psi\{^{w'}/_{w}\}$ where w' is a fresh name. Hence, we have to show that for all terms M and N such that $fn(M, N) \cap \tilde{n} = \emptyset$, we have that:

1. $(M =_{\mathsf{E}} N)\psi \Rightarrow (M =_{\mathsf{E}} N)(\psi\{^{w'}/_{w}\})$; and
2. $(M =_{\mathsf{E}} N)(\psi\{^{w'}/_{w}\}) \Rightarrow (M =_{\mathsf{E}} N)\psi$[4].

Actually, it is sufficient to establish this result for all terms M and N such that $c_1, \ldots, c_k$, $w_1, \ldots, w_k$ do not occur in M and N. This comes from the fact that these names do not occur in ψ and $\psi\{^{w'}/_{w}\}$. Moreover, we can assume w.l.o.g. that $\tilde{n} \cap (fn(M) \cup fn(N)) = \emptyset$. Lastly, we will consider the first item (the other one can be proved in a similar way) and thus we can assume that $w' \notin (fn(M) \cup fn(N))$.

Let split_ψ (resp. split_0) be the splitting function (resp. ground splitting function) w.r.t. $\psi = \nu\tilde{n}.(\sigma\delta_{c_i,t_i}\delta_{w_i,w})$, w, $c_1, \ldots, c_k$, $w_1, \ldots, w_k$, $t_1, \ldots, t_k$. Let $w'_1, \ldots, w'_k$ be distinct fresh names (we assume w.l.o.g. that they do not occur in M and N). We denote by $\#_w M$ the number of occurrences of w in M, and by $\#M$ the size of M[5]. We denote by $|M|$ the measure $(\#_w M, \#M)$ and we use the lexicographic ordering. We show by induction on $\max(|M|, |N|)$ that:

1. $[\mathsf{split}_\psi(M)(\sigma\delta_{w_i,w'_i})]\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'} =_{\mathsf{E}} M(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'})$
2. $(M =_{\mathsf{E}} N)\psi \;\Rightarrow\; (M =_{\mathsf{E}} N)(\psi\{^{w'}/_{w}\})$

where

- $\delta_{w_i,w'_i} = \{^{w'_1}/_{w_1}\} \ldots \{^{w'_k}/_{w_k}\}$;
- $t'_i = t_i\{^{w'}/_{w}\}$ for $1 \leq i \leq k$, and $\delta_{c_i,t'_i} = \{^{t'_1}/_{c_1}\} \ldots \{^{t'_k}/_{c_k}\}$;
- $\delta_{w_i,w'} = \{^{w'}/_{w_1}\} \ldots \{^{w'}/_{w_k}\}$; and
- $\delta_{w'_i,w'} = \{^{w'}/_{w'_1}\} \ldots \{^{w'}/_{w'_k}\}$.

Base case: $\max(|M|, |N|) \leq (1, 1)$. This means that M (resp. N) do not contain any occurrence of w, or M (resp. N) is equal to w.

1. In both cases, we have that $\mathsf{split}_\psi(M) = M$. This comes from the fact that w is not deducible from $\nu w.\psi$ since all the occurrences of w are under an h. Hence, we have that:

$$\begin{array}{ll} & [\mathsf{split}_\psi(M)(\sigma\delta_{w_i,w'_i})]\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'} \\ =_{\mathsf{E}} & M(\sigma\delta_{w_i,w'_i}\delta_{c_i,t'_i}\delta_{w'_i,w'}) \qquad \text{since } w_i, c_i, w'_i \notin fn(M) \\ =_{\mathsf{E}} & M(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'}) \qquad \text{since } w'_i \notin fn(\sigma) \end{array}$$

2. The second point can be proved as follows:

$$\begin{array}{lll} (M =_{\mathsf{E}} N)\psi & & \\ \Rightarrow & M(\sigma\delta_{c_i,t_i}\delta_{w_i,w}) =_{\mathsf{E}} N(\sigma\delta_{c_i,t_i}\delta_{w_i,w}) & \text{by def. of } \psi \\ \Rightarrow & \mathsf{split}_0(M(\sigma\delta_{c_i,t_i}\delta_{w_i,w})) =_{\mathsf{E}} \mathsf{split}_0(N(\sigma\delta_{c_i,t_i}\delta_{w_i,w})) & \text{Lemma 14} \\ \Rightarrow & \mathsf{split}_\psi(M)\sigma =_{\mathsf{E}} \mathsf{split}_\psi(N)\sigma & \text{Lemma 15} \\ \Rightarrow & (\mathsf{split}_\psi(M) =_{\mathsf{E}} \mathsf{split}_\psi(N))\phi & \\ & \qquad \text{since } (fn(\mathsf{split}_\psi(N)) \cup fn(\mathsf{split}_\psi(M))) \cap \tilde{n} = \emptyset & \\ \Rightarrow & (\mathsf{split}_\psi(M) =_{\mathsf{E}} \mathsf{split}_\psi(N))(\phi\delta_{w_i,w'_i}) & \text{since } \phi \approx \phi\delta_{w_i,w'_i} \\ \Rightarrow & \mathsf{split}_\psi(M)(\sigma\delta_{w_i,w'_i}) =_{\mathsf{E}} \mathsf{split}_\psi(N)(\sigma\delta_{w_i,w'_i}) & \\ \Rightarrow & \mathsf{split}_\psi(M)(\sigma\delta_{w_i,w'_i})\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'} & \\ & =_{\mathsf{E}} \mathsf{split}_\psi(N)(\sigma\delta_{w_i,w'_i})\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'} & \\ \Rightarrow & M(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'}) =_{\mathsf{E}} N(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'}) & \text{item 1 (base case)} \\ \Rightarrow & (M =_{\mathsf{E}} N)(\psi\{^{w'}/_{w}\}) & \end{array}$$

Induction step: $\max(|M|, |N|) \geq (1, 2)$. We assume w.l.o.g. that $|M| \geq |N|$, thus $M = \mathsf{f}(M_1, \ldots, M_\ell)$. As for each $1 \leq i \leq k$ we have that $\nu w.\psi \vdash t_i$ there exist ζ_i such that $fn(\zeta_i) \cap (\{c_1, \ldots, c_k, w_1, \ldots, w_k, w\} \cup \tilde{n}) = \emptyset$ and $\zeta_i(\sigma\delta_{c_i,t_i}\delta_{w_i,w}) =_{\mathsf{E}} t_i$.

1. To establish the first point, we distinguish two cases.

[4] The notation $M\psi$ simply means $M\sigma$ where σ is the substitution involved in the frame, i.e. $\psi = \nu\tilde{n}.\sigma$.

[5] The size $\#M$ of a term M is defined by $\#M = 1$ when M is a name or a variable and $\#\mathsf{f}(M_1, \ldots, M_\ell) = 1 + \sum_{i=1}^{\ell} \#M_i$.

– $\mathsf{f} = \mathsf{h}$, $\ell = 2$, $(M_1 =_{\mathsf{E}} \zeta_{i_0})\psi$, and $(M_2 =_{\mathsf{E}} w)\psi$ for some $i_0 \in \{1, \ldots, k\}$. Applying our induction hypothesis, we deduce that $(M_1 =_{\mathsf{E}} \zeta_{i_0})(\psi\{^{w'}/_{w}\})$ and $(M_2 =_{\mathsf{E}} w)(\psi\{^{w'}/_{w}\})$. Note that $\#_w\zeta_{i_0} = 0$ and $\#_w M_2 \geq 1$ (it is not possible to deduce w without using it explicitly). Hence, we can indeed apply our induction hypothesis in order to deduce that:

$$\begin{aligned}
[\mathsf{split}_\psi(M)(\sigma\delta_{w_i,w'_i})]\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'} &=_{\mathsf{E}} \mathsf{h}(c_{i_0}, w_{i_0})\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'} \\
&=_{\mathsf{E}} \mathsf{h}(t'_{i_0}, w) \\
&=_{\mathsf{E}} \mathsf{h}(M_1(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'}), M_2(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'})) \\
&=_{\mathsf{E}} M(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'})
\end{aligned}$$

– Otherwise, $\mathsf{split}_\psi(M) = \mathsf{f}(\mathsf{split}_\psi(M_1), \ldots, \mathsf{split}\psi(M_\ell))$, and thus we have that:

$$\begin{aligned}
&[\mathsf{split}_\psi(M)(\sigma\delta_{w_i,w'_i})]\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'} \\
&=_{\mathsf{E}} [\mathsf{f}(\mathsf{split}_\psi(M_1), \ldots, \mathsf{split}_\psi(M_\ell))(\sigma\delta_{w_i,w'_i})]\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'} \\
&=_{\mathsf{E}} \mathsf{f}(\mathsf{split}_\psi(M_1)(\sigma\delta_{w_i,w'_i})\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'}, \ldots, \mathsf{split}_\psi(M_\ell)(\sigma\delta_{w_i,w'_i})\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'}) \\
&=_{\mathsf{E}} \mathsf{f}(M_1(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'}), \ldots, M_\ell(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'})) \\
&=_{\mathsf{E}} \mathsf{f}(M_1, \ldots, M_\ell)(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'}) \\
&=_{\mathsf{E}} M(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'})
\end{aligned}$$

2. This point can be proved as in the base case.

The second implication, $(M =_{\mathsf{E}} N)(\psi\{^{w'}/_{w}\}) \Rightarrow (M =_{\mathsf{E}} N)\psi$ can be proved in a similar way. This allows us to conclude the proof. □

C.2 Proof of Proposition 4

Lemma 6 *Let $t_1, \ldots, t_k$ be distinct ground terms modulo E and $c_1, \ldots, c_k, w_1, \ldots, w_k$ be distinct fresh names. Let $\phi = \nu\tilde{n}.\sigma$, $\tilde{\phi} = \nu\tilde{n}.\tilde{\sigma}$ and $\phi' = \nu\tilde{n}.\sigma'$ be three frames such that $w \notin fn(\sigma)$, and $w, w_1, \ldots, w_k, c_1, \ldots, c_k \notin \tilde{n}$. Moreover, we assume that $\sigma\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{\sigma}$, $\sigma =_{\mathsf{E}} \sigma'\delta_{w_i,\mathsf{h}(c_i,w_i)}$, and $c_1, \ldots, c_k \notin fn(\sigma')$. If $\nu w.\tilde{\phi} \vdash_{\mathsf{E}} \tilde{M}$ and $\{w_1, \ldots, w_k, c_1, \ldots, c_k\} \cap fn(\tilde{M}) = \emptyset$ for some ground term $\tilde{M}$ then there exist ground terms M, M' such that $c_1, \ldots, c_k \notin fn(M')$, $w \notin fn(M)$, $M\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{M}$, $M =_{\mathsf{E}} M'\delta_{w_i,\mathsf{h}(c_i,w_i)}$, and $\nu w_1 \ldots \nu w_k.\phi \vdash_{\mathsf{E}} M$.*

Proof Let $\tilde{M}$ be a ground term such that $\nu w.\tilde{\phi} \vdash_{\mathsf{E}} \tilde{M}$ and $\{w_1, \ldots, w_k, c_1, \ldots, c_k\} \cap fn(\tilde{M}) = \emptyset$. Thus, there exists a term ζ such that $fn(\zeta) \cap (\tilde{n} \cup \{w, w_1, \ldots, w_k, c_1, \ldots, c_k\}) = \emptyset$, $fv(\zeta) \subseteq \mathrm{dom}(\tilde{\sigma})$, and $\zeta\tilde{\sigma} =_{\mathsf{E}} \tilde{M}$. Let $M' = \zeta\sigma'$ and $M = \mathsf{split}_0(\zeta\tilde{\sigma})$ where split_0 is the ground splitting function w.r.t. w, $c_1, \ldots, c_k$, $w_1, \ldots, w_k$, $t_1, \ldots, t_k$. We have that $c_1, \ldots, c_k \notin fn(M')$, and $w \notin fn(M)$. By hypothesis, we have that $\zeta\tilde{\sigma} =_{\mathsf{E}} \tilde{M}$. Thus, thanks to Lemma 14, we have that $M = \mathsf{split}_0(\zeta\tilde{\sigma}) =_{\mathsf{E}} \mathsf{split}_0(\tilde{M})$. Now, thanks to Lemma 15, we deduce that $\mathsf{split}_{\tilde{\phi}}(\zeta)\sigma =_{\mathsf{E}} M$ where $\mathsf{split}_{\tilde{\phi}}$ is the splitting function w.r.t. $\tilde{\phi}$, w, $c_1, \ldots, c_k$, $w_1, \ldots, w_k$, $t_1, \ldots, t_k$. Actually, since $\tilde{\sigma} = \sigma\delta_{c_i,t_i}\delta_{w_i,w}$ and $\sigma =_{\mathsf{E}} \sigma'\delta_{w_i,\mathsf{h}(c_i,w_i)}$, we have that w only appears under h and hence is not deducible from $\nu w.\tilde{\phi}$. This allows us to show that $\mathsf{split}_{\tilde{\phi}}(\zeta) = \zeta$. Hence, we have that $\zeta\sigma =_{\mathsf{E}} M$. Lastly, we have that

– $M =_{\mathsf{E}} \zeta\sigma =_{\mathsf{E}} (\zeta\sigma')\delta_{w_i,\mathsf{h}(c_i,w_i)} = M'\delta_{w_i,\mathsf{h}(c_i,w_i)}$, and
– $M\delta_{c_i,t_i}\delta_{w_i,w} =_{\mathsf{E}} [(\zeta\sigma')\delta_{w_i,\mathsf{h}(c_i,w_i)}]\delta_{c_i,t_i}\delta_{w_i,w} = (\zeta\sigma)\delta_{c_i,t_i}\delta_{w_i,w} = \zeta\tilde{\sigma}$.

This allows us to conclude the proof. □

Lemma 7 *Let $t_1, \ldots, t_k$ be distinct ground terms modulo E and $c_1, \ldots, c_k, w_1, \ldots, w_k$ be distinct fresh names. Let M, N, $\tilde{M}$ and $\tilde{N}$ be four terms such that*

- *$\tilde{M} = M\delta_{c_i,t_i}\delta_{w_i,w}$ and $\tilde{N} = N\delta_{c_i,t_i}\delta_{w_i,w}$ with $w \notin fn(M) \cup fn(N)$;*
- *$M =_{\mathsf{E}} M'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ and $N =_{\mathsf{E}} N'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ for some terms M' and N' such that $c_1, \ldots, c_k \notin fn(M') \cup fn(N')$.*

Then, we have that $M =_{\mathsf{E}} N$ if and only if $\tilde{M} =_{\mathsf{E}} \tilde{N}$.

Proof As $=_{\mathsf{E}}$ is closed under substitution of terms for names $M =_{\mathsf{E}} N$ implies $\tilde{M} =_{\mathsf{E}} \tilde{N}$. Now, let M and N be two terms such that $\tilde{M} =_{\mathsf{E}} \tilde{N}$ where $\tilde{M} = M\delta_{c_i,t_i}\delta_{w_i,w}$ and $\tilde{N} = N\delta_{c_i,t_i}\delta_{w_i,w}$. Thus, according to Lemma 14, we have that

$$\mathsf{split}_0(M\delta_{c_i,t_i}\delta_{w_i,w}) =_{\mathsf{E}} \mathsf{split}_0(N\delta_{c_i,t_i}\delta_{w_i,w})$$

where split_0 represents the splitting function w.r.t. w, $c_1, \ldots, c_k$, $w_1, \ldots, w_k$, $t_1, \ldots, t_k$. Now, it is easy to establish, by structural induction on M and N and by relying on the fact that $M =_{\mathsf{E}} M'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ for some term M', and $N =_{\mathsf{E}} N'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ for some term N', that:

$$\mathsf{split}_0(M\delta_{c_i,t_i}\delta_{w_i,w}) =_E M \quad \text{and} \quad \mathsf{split}_0(N\delta_{c_i,t_i}\delta_{w_i,w}) =_E N.$$

This allows us to conclude.

We will prove Proposition 4 by induction on the prooftree witnessing the derivation. First, we establish a similar result for $\equiv$.

Lemma 16 *Let $t_1, \ldots, t_k$ be distinct ground terms modulo E and $c_1, \ldots, c_k, w_1, \ldots, w_k$ be distinct fresh names. Let A be an extended process such that $bn(A) = \emptyset$, $w \notin fn(A)$, and $A =_{\mathsf{E}} A'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ for some A' such that $c_1, \ldots, c_k \notin fn(A')$. Suppose that $A\delta_{c_i,t_i}\delta_{w_i,w} \equiv \overline{B}$ for some process $\overline{B}$. Then there exist some processes B and B' such that*

- *$\overline{B} = B\delta_{c_i,t_i}\delta_{w_i,w}$ with $w \notin fn(B)$, and*
- *$B =_{\mathsf{E}} B'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ with $c_1, \ldots, c_k \notin fn(B')$, and*
- *$A \equiv B$.*

Proof Let $\overline{A} = A\delta_{c_i,t_i}\delta_{w_i,w}$. We prove this result by induction on the proof tree showing that $\overline{A} \equiv \overline{B}$. All the base cases that we have to check, i.e. Par-0, Par-C and Par-A, are easy to prove. The only interesting inductive case is the case of an application of an evaluation context. Suppose that the proof tree showing that $\overline{A} \equiv \overline{B}$ ends with an instance of such a rule, i.e.

$$\frac{\overline{A_1} \equiv \overline{B_1}}{\overline{C}[\overline{A_1}] \equiv \overline{C}[\overline{B_1}]}$$

where $\overline{A} = \overline{C}[\overline{A_1}]$ and $\overline{B} = \overline{C}[\overline{B_1}]$. Note that the evaluation context will not contain any ν operator since otherwise $bn(A) \neq \emptyset$. As $\overline{A} = A\delta_{c_i,t_i}\delta_{w_i,w}$ we have that there exist A_1, C such that $A_1\delta_{c_i,t_i}\delta_{w_i,w} = \overline{A_1}$ and $C\delta_{c_i,t_i}\delta_{w_i,w} = \overline{C}$. Moreover there exists A' such that $C[A_1] = A =_{\mathsf{E}} A'\delta_{w_i,\mathsf{h}(c_i,w_i)}$. Hence there also exist C', A_1' such that $C =_{\mathsf{E}} C'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ and $A_1 =_{\mathsf{E}} A_1'\delta_{w_i,\mathsf{h}(c_i,w_i)}$. We can therefore apply our induction hypothesis and we obtain that there exist processes B_1, B_1' such that

- *$\overline{B_1} = B_1\delta_{c_i,t_i}\delta_{w_i,w}$;*
- *$B_1 =_{\mathsf{E}} B_1'\delta_{w_i,\mathsf{h}(c_i,w_i)}$;*
- *$A_1 \equiv B_1$.*

Let $B = C[B_1]$ and $B' = C'[B_1']$. We indeed have that

- *$\overline{B} = \overline{C}[\overline{B_1}] = (C\delta_{c_i,t_i}\delta_{w_i,w})[B_1\delta_{c_i,t_i}\delta_{w_i,w}] = B\delta_{c_i,t_i}\delta_{w_i,w}$*
- *$B = C[B_1] =_{\mathsf{E}} C'[B_1']\delta_{w_i,\mathsf{h}(c_i,w_i)} = B'\delta_{w_i,\mathsf{h}(c_i,w_i)}$.*

This allows us to conclude the proof. □

Now, we can prove the following proposition.

Proposition 4 *Let $t_1, \ldots, t_k$ be distinct ground terms modulo E and $c_1, \ldots, c_k, w_1, \ldots, w_k$ be distinct fresh names. Let $\nu\tilde{n}.A$ be an extended process such that $bn(A) = \emptyset$, $w \notin fn(A)$, and $A =_{\mathsf{E}} A'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ for some A' such that $c_1, \ldots, c_k \notin fn(A')$. Moreover, we assume that $w, w_1, \ldots, w_k, c_1, \ldots, c_k \notin \tilde{n}$.*

Let $\overline{B}$ be such that $\nu w.\nu\tilde{n}.(A\delta_{c_i,t_i}\delta_{w_i,w}) \xrightarrow{\ell} \overline{B}$. Moreover, when $\ell = in(\tilde{M})$ we assume that $w_1, \ldots, w_k, c_1, \ldots, c_k \notin fn(\tilde{M})$. Then there exist extended processes B, B', and labels ℓ_0, ℓ' such that:

- *$\overline{B} \equiv \nu w.\nu\tilde{n}.(B\delta_{c_i,t_i}\delta_{w_i,w})$ with $bn(B) = \emptyset$ and $w \notin fn(B)$, $\ell = \ell_0\delta_{c_i,t_i}\delta_{w_i,w}$, and*
- *$B =_{\mathsf{E}} B'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ with $c_1, \ldots, c_k \notin fn(B')$, $\ell_0 =_{\mathsf{E}} \ell'\delta_{w_i,\mathsf{h}(c_i,w_i)}$, and*
- *$\nu w_1 \ldots \nu w_k.\nu\tilde{n}.A \xrightarrow{\ell_0} \nu w_1 \ldots \nu w_k.\nu\tilde{n}.B$.*

Proof We have $\nu w.\nu\tilde{n}.(A\delta_{c_i,t_i}\delta_{w_i,w}) \xrightarrow{\ell} \overline{B}$. It is easy to see that $w \in bn(\overline{B})$ and $\tilde{n} \subseteq bn(\overline{B})$. Indeed, according to our calculus, we can always by using structural equivalence move a restriction in front of the process. Thus we have that $\overline{B} \equiv \nu w.\nu\tilde{n}.\tilde{B}$ for some process $\tilde{B}$ such that $bn(\tilde{B}) = \emptyset$. Let ℓ be the label involved in $\nu w.\nu\tilde{n}.(A\delta_{c_i,t_i}\delta_{w_i,w}) \to \overline{B}$. It is easy to see that $A\delta_{c_i,t_i}\delta_{w_i,w} \xrightarrow{\ell} \tilde{B}$ and when $\ell = in(\tilde{M})$, we have that $\nu w.\nu\tilde{n}(\phi(A)\delta_{c_i,t_i}\delta_{w_i,w}) \vdash_{\mathsf{E}} \tilde{M}$. Moreover, by hypothesis, we have that $w_1, \ldots, w_k, c_1, \ldots, c_k \notin fn(\tilde{M})$. By Lemma 6, we deduce that $\nu w_1 \ldots \nu w_k.\nu\tilde{n}.\phi(A) \vdash_{\mathsf{E}} M$ for some M such that $M\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{M}$ and we also know that there exists M' such that $M =_{\mathsf{E}} M'\delta_{w_i,\mathsf{h}(c_i,w_i)}$. This allows us, in particular, to ensure that, in the case of an input, the side condition corresponding to an application of evaluation context is satisfied. Now, we show by induction on the proof tree showing that $A\delta_{c_i,t_i}\delta_{w_i,w} \xrightarrow{\ell} \tilde{B}$ that there exist processes B,B', and labels ℓ_0, ℓ' such that

- $\tilde{B} = B\delta_{c_i,t_i}\delta_{w_i,w}$ with $w \notin fn(B)$, and $\ell = \ell_0\delta_{c_i,t_i}\delta_{w_i,w}$;
- $B =_{\mathsf{E}} B'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ with $c_1, \ldots, c_k \notin fn(B')$, and $\ell_0 =_{\mathsf{E}} \ell'\delta_{w_i,\mathsf{h}(c_i,w_i)}$;
- $A \to B$

This will allows us to conclude that $\nu w_1 \ldots \nu w_k.\nu\tilde{n}.A \to \nu w_1 \ldots \nu w_k.\nu\tilde{n}.B$. Note that since $bn(\tilde{B}) = \emptyset$, we have also that $bn(B) = \emptyset$.

Base cases.

- In. In such a case, we have $A\delta_{c_i,t_i}\delta_{w_i,w} = \mathrm{in}(x).\tilde{P}$ and $\tilde{B} = \tilde{P}\{^{\tilde{M}}/_x\}$ for some process $\tilde{P}$ and some term $\tilde{M}$. From this, we deduce that $A = \mathrm{in}(x).P$ for some process P such that $P\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{P}$. We have also that $A = \mathrm{in}(x).P =_{\mathsf{E}} A'\delta_{w_i,\mathsf{h}(c_i,w_i)}$.
  Thus, there exists P' with $c_1, \ldots, c_k \notin fn(P')$ such that $P =_{\mathsf{E}} P'\delta_{w_i,\mathsf{h}(c_i,w_i)}$. Moreover, we have already seen that there exists M and M' such that
  - $M\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{M}$, and
  - $M =_{\mathsf{E}} M'\delta_{w_i,\mathsf{h}(c_i,w_i)}$.

  Let $B = P\{^{M}/_x\}$, $B' = P'\{^{M'}/_x\}$, $\ell_0 = \mathrm{in}(M)$, and $\ell' = \mathrm{in}(M')$. It is easy to check that the three conditions hold.
- Out. In such a case, we have $A\delta_{c_i,t_i}\delta_{w_i,w} = \mathrm{out}(\tilde{M}).\tilde{P}$ and $\tilde{B} = \tilde{P} \mid \{^{\tilde{M}}/_x\}$ for some process $\tilde{P}$ and some term $\tilde{M}$. From this, we deduce that $A = \mathrm{out}(M).P$ for some term M and some process P such that $M\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{M}$, and $P\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{P}$. We have also that $A = \mathrm{out}(M).P =_{\mathsf{E}} A'\delta_{w_i,\mathsf{h}(c_i,w_i)}$.
  Thus, there exist M' and P' such that $M =_{\mathsf{E}} M'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ and $P =_{\mathsf{E}} P'\delta_{w_i,\mathsf{h}(c_i,w_i)}$. Moreover, we have that $c_1, \ldots, c_k \notin fn(M') \cup fn(P')$. Let $B = P \mid \{^{M}/_x\}$, $B' = P' \mid \{^{M'}/_x\}$, $\ell_0 = \mathrm{out}(M)$, and $\ell' = \mathrm{out}(M')$. It is easy to check that the three conditions hold.
- Event. In such a case, we have $A\delta_{c_i,t_i}\delta_{w_i,w} = \mathrm{ev}(\tilde{M}).\tilde{P}$ and $\tilde{B} = \tilde{P} \mid \{^{\tilde{M}}/_x\}$ for some process $\tilde{P}$ and some terms $\tilde{M}$. From this, we deduce that $A = \mathrm{ev}(M).P$ for some terms M and some process P such that $M\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{M}$, and $P\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{P}$. We have also that $A = \mathrm{ev}(M).P =_{\mathsf{E}} A'\delta_{w_i,\mathsf{h}(c_i,w_i)}$.

Thus, there exist M' and P' such that $M =_{\mathsf{E}} M'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ and $P =_{\mathsf{E}} P'\delta_{w_i,\mathsf{h}(c_i,w_i)}$. Moreover, we have that $c_1,\ldots,c_k \notin fn(M') \cup fn(P')$. Let $B = P$, $B' = P'$, $\ell_0 = \mathrm{ev}(M)$, and $\ell' = \mathrm{ev}(M')$. It is easy to check that the three conditions hold.

- Then. In such a case, we have $A\delta_{c_i,t_i}\delta_{w_i,w} = \text{if } \tilde{M}_1 = \tilde{M}_2 \text{ then } \tilde{P} \text{ else } \tilde{Q}$ for some terms $\tilde{M}_1$ and $\tilde{M}_2$ and some processes $\tilde{P}$ and $\tilde{Q}$ such that $\tilde{M}_1 =_{\mathsf{E}} \tilde{M}_2$ and $\tilde{B} = \tilde{P}$. From this, we deduce that $A = \text{if } M_1 = M_2 \text{ then } P \text{ else } Q$ for some terms M_1, M_2 and some processes P, Q such that $M_i\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{M}_i$ $(i = 1, 2)$, $P\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{P}$, and $Q\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{Q}$. We have also that $A = \text{if } M_1 = M_2 \text{ then } P \text{ else } Q =_{\mathsf{E}} A'\delta_{w_i,\mathsf{h}(c_i,w_i)}$.
  Thus, there exist M'_1, M'_2, P' and Q' such that:
  - $M_i =_{\mathsf{E}} M'_i\delta_{w_i,\mathsf{h}(c_i,w_i)}$ $(i = 1, 2)$,
  - $P =_{\mathsf{E}} P'\delta_{w_i,\mathsf{h}(c_i,w_i)}$, and
  - $Q =_{\mathsf{E}} Q'\delta_{w_i,\mathsf{h}(c_i,w_i)}$.

  Moreover, we have that $c_1,\ldots,c_k \notin fn(M'_1) \cup fn(M'_2) \cup fn(P') \cup fn(Q')$. Let $B = P$, $B' = P'$, and $\ell_0 = \ell = \tau$. It is easy to see that the two first conditions hold. For the last one, we have to show that $M_1 =_{\mathsf{E}} M_2$. This can be easily done thanks to Lemma 7.
- Else. This case is similar to the previous one.

Inductive cases. The inductive case corresponding to application of structural equivalence directly follows from Lemma 16. It remains to show the case of an application of an evaluation context. In such a case, we have $A\delta_{c_i,t_i}\delta_{w_i,w} \xrightarrow{\ell} \tilde{B}$ finishes by an application of the following rule

$$\frac{\tilde{A}_1 \xrightarrow{\ell} \tilde{B}_1}{\tilde{C}[\tilde{A}_1] \xrightarrow{\ell} \tilde{C}[\tilde{B}_1]}$$

where $A\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{C}[\tilde{A}_1]$ and $\tilde{B} = \tilde{C}[\tilde{B}_1]$. From this, we deduce that $A = C[A_1]$ for some context C and some process A_1 such that $C\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{C}$ and $A_1\delta_{c_i,t_i}\delta_{w_i,w} = \tilde{A}_1$. We have $A = C[A_1] =_{\mathsf{E}} A'\delta_{w_i,\mathsf{h}(c_i,w_i)}$. Thus, there exist C' and A'_1 such that $C =_{\mathsf{E}} C'\delta_{w_i,\mathsf{h}(c_i,w_i)}$, and $A_1 =_{\mathsf{E}} A'_1\delta_{w_i,\mathsf{h}(c_i,w_i)}$. Hence we can apply our induction hypothesis to obtain that there exist B'_1, B_1, ℓ_0, and ℓ' such that

- $\tilde{B}_1 \equiv B_1\delta_{c_i,t_i}\delta_{w_i,w}$ with $w \notin fn(B_1)$, and $\ell = \ell_0\delta_{c_i,t_i}\delta_{w_i,w}$;
- $B_1 =_{\mathsf{E}} B'_1\delta_{w_i,\mathsf{h}(c_i,w_i)}$ with $c_1,\ldots,c_k \notin fn(B'_1)$, and $\ell_0 =_{\mathsf{E}} \ell'\delta_{w_i,\mathsf{h}(c_i,w_i)}$;
- $A_1 \to B_1$.

Let $B = C[B_1]$ and $B' = C'[B'_1]$. The three conditions hold and this allows us to conclude the proof. □

C.3 Proof of Theorem 3

Theorem 3 *Let $\mathcal{P} = \nu w.(\nu\tilde{m}_1.P_1 \mid \ldots \mid \nu\tilde{m}_\ell.P_\ell)$ be a password protocol specification and $\mathcal{P}'$ be such that $\overline{\mathcal{P}} = \nu w.\mathcal{P}'$, and $\mathcal{P}'_1,\ldots\mathcal{P}'_p$ be p instances of $\mathcal{P}'$.*

1. *Let t be a ground term that occurs as a subterm in $\mathcal{P}'_i$ for some $i \in \{1,\ldots,p\}$. If $\nu w.\mathcal{P}'_i$ preserves secrecy of t, then we have that $\nu w.(\mathcal{P}'_1 \mid \ldots \mid \mathcal{P}'_p)$ preserves secrecy of $t\{^{\mathsf{h}(t_i,w)}/_w\}$.*
2. *Let $\Phi = ev(\tilde{x}) \Rightarrow_{(\mathsf{inj})} ev(\tilde{x})$ be a correspondence property (injective or not). If Φ holds on $\mathcal{P}$, then Φ holds on $\nu w.(\mathcal{P}'_1 \mid \ldots \mid \mathcal{P}'_p)$.*
3. *If $\mathcal{P}$ is resistant to guessing attacks against w, then we have that $\nu w.(\mathcal{P}'_1 \mid \ldots \mid \mathcal{P}'_p)$ is resistant to guessing attacks against w.*

Proof We suppose w.l.o.g. that $\mathcal{P}'_i = \nu\tilde{m}_{i,1}\nu n_{i,1}.P_{i,1} \mid \ldots \mid \nu\tilde{m}_{i,\ell}\nu n_{i,\ell}.P_{i,\ell}$ where

$$P_{i,j} = \mathrm{in}(x^1_{i,j}).\ldots.\mathrm{in}(x^{j-1}_{i,j}).\mathrm{out}(n_{i,j}).\mathrm{in}(x^{j+1}_{i,j}).\ldots.\mathrm{in}(x^{\ell}_{i,j}).P'_{i,j}$$

for some $P'_{i,j}$ $(1 \le i \le p, 1 \le j \le \ell)$.

By contradiction, suppose that $P = \nu w.(\mathcal{P}'_1 \mid \ldots \mid \mathcal{P}'_p)$ admits an attack. Throughout the proof we refer to an attack as being either an attack on secrecy, on a correspondence property or a guessing attack. Hence there exists Q such that $P \rightarrow^* Q$ is the derivation exhibiting this attack. We assume w.l.o.g. that the derivation is maximal, i.e. there is no Q' such that $Q \rightarrow Q'$. This allows us to ensure that all the preambles have been executed. We are going to show that there exists an attack on $\mathcal{P}$ contradicting the hypothesis.

Step 1. We will first regroup the different roles of the protocol instances according to their tag. For this we need to identify the tag $t_{i,j}$ that is computed by $P_{i,j}$ during the attack derivation. We have that $P \xrightarrow{\ell_1} P_1 \xrightarrow{\ell_2} \cdots P_{q-1} \xrightarrow{\ell_q} P_q = Q$ and for each $x^k_{i,j}$ such that $j \neq k$ there exists r such that $P_r \equiv C[\mathrm{in}(x^k_{i,j}).P'] \xrightarrow{\mathrm{in}(M^k_{i,j})} C[P'\{^{M^k_{i,j}}/_{x^k_{i,j}}\}] \equiv P_{r+1}$. Moreover, for each i, j such that $1 \leq i \leq p, 1 \leq j \leq \ell$ there exists $y_{i,j} \in \mathrm{dom}(\phi(Q))$ such that $y_{i,j}\phi(Q) = n_{i,j}$. Let $M^j_{i,j} = n_{i,j}$. We define $t_{i,j} = \langle M^1_{i,j}, \langle \ldots \langle M^{\ell-1}_{i,j}, M^\ell_{i,j} \rangle \rangle \rangle$. We note that $\phi(Q) \vdash t_{i,j}$ for all i, j such that $1 \leq i \leq p, 1 \leq j \leq \ell$. Intuitively, $t_{i,j}$ is the tag which has been computed by process $P_{i,j}$ in the attack derivation.

Next we regroup the roles in P according to the tag they used. Let $tag_1, \ldots, tag_k$ be the different terms (modulo E) that occur in $\{t_{i,j} | 1 \leq i \leq \ell \text{ and } 1 \leq j \leq p\}$. By definition, the terms $tag_1, \ldots, tag_k$ are distinct modulo E. We group the different processes of P according to the value of the tag in the derivation, i.e., we define

$$\overline{A_r} = \nu \tilde{m}_r.\, |_{i,j \text{ s.t. } t_{i,j} = tag_r} P_{i,j} \text{ where } \tilde{m}_r = (\cup_{i,j \text{ s.t. } t_{i,j} = tag_r} \tilde{m}_{i,j}, n_{i,j})$$

We have that $P \equiv \nu w.(\overline{A_1} \mid \ldots \mid \overline{A_k})$ and we let $\tilde{m}$ stand for the sequence $\nu \tilde{m}_1 \ldots \nu \tilde{m}_k$.

Step 2. The aim of this step is to show that an attack on a transformed protocol also exists on a protocol that is tagged with constants (instead of the constructed tag) and different passwords (instead of the same password).

We first instantiate the tag of each role $P_{i,j}$ by the tag that has been computed in the attack derivation. Define the process $\overline{P}_0$ obtained from P by replacing each occurrence of a non-instantiated tag $\langle x^1_{i,j}, \langle \ldots n_{i,j} \ldots \langle x^{\ell-1}_{i,j}, x^\ell_{i,j} \rangle \rangle \rangle$ in $\overline{A}_r$ by the ground term tag_r. It is easy to see that $\overline{P}_0 \rightarrow^* Q$. Moreover, by construction each $\overline{A_i}$ is of the form $A_i \delta_{c_i, tag_i} \delta_{w_i, w}$ with $A_i = A'_i \delta_{w, \mathsf{h}(c_i, w_i)}$ for some A_i, A'_i and $c_1, \ldots c_k, w_1, \ldots w_k$ which do not occur in $\overline{P}_0$. As $w_1, \ldots w_k, c_1, \ldots c_k$ do not occur in $\overline{P}_0$ we assume w.l.o.g. that they do not occur in any label among this derivation.

Let $\overline{P}_n = Q$ and $P_0 = (\nu w_1.A_1 \mid \ldots \mid \nu w_k.A_k)$. By iterating Proposition 4 we have that there exist two extended processes P_n, P'_n and two sequences of labels $\ell^0_1, \ldots, \ell^0_n$ and $\ell'_1, \ldots, \ell'_n$ such that:

- $\overline{P}_n \equiv \nu w.\nu \tilde{m}.(P_n \delta_{c_i, t_i} \delta_{w_i, w})$ with $bn(P_n) = \emptyset$, $w \notin fn(P_n)$, and $\overline{\ell_j} = \ell^0_j \delta_{c_i, t_i} \delta_{w_i, w}$ for any $j \in \{1, \ldots, n\}$;
- $P_n =_{\mathsf{E}} P'_n \delta_{w_i, \mathsf{h}(c_i, w_i)}$ with $c_1, \ldots, c_k \notin fn(P'_n)$, and $\ell^0_j =_{\mathsf{E}} \ell'_j \delta_{w_i, \mathsf{h}(c_i, w_i)}$ for any $j \in \{1, \ldots, n\}$, and
- $P_0 \xrightarrow{\ell^0_1} \ldots \xrightarrow{\ell^0_n} \nu w_1. \ldots . \nu w_k . \nu \tilde{m}.P_n$.

Exactly as in the proof of Theorem 2, using Lemmas 6, 7 and 8 we show that the derivation $P_0 \xrightarrow{\ell^0_1} \ldots \xrightarrow{\ell^0_n} \nu w_1. \ldots . \nu w_k . \nu \tilde{m}.P_n$ also admits an attack.

Step 3. In the final step we are going to show that the attack already existed on an instance of $\mathcal{P}$ contradicting the hypothesis.

By Proposition 1, we have for some r that $\nu w_r.\nu \tilde{m}_r.A_r$ admits an attack. We have that $A_r = |_{i,j \text{ s.t. } t_{i,j} = tag_r} Q_{i,j}$ and the $Q_{i,j}$s are of the form

$$Q_{i,j} = \mathrm{in}(x^1_{i,j}). \ldots \mathrm{in}(x^{j-1}_{i,j}).\mathrm{out}(n_{i,j}).\mathrm{in}(x^{j+1}_{i,j}). \ldots \mathrm{in}(x^\ell_{i,j}).Q'_{i,j}$$

for some $Q'_{i,j}$ such that $x^1_{i,j}, \ldots, x^{j-1}_{i,j}, n_{i,j}, x^{i+1}_{i,j}, x^{\ell}_{i,j}$ do not occur in $Q'_{i,j}$. Hence, we also have that $\nu w_r.\nu \tilde{m}_r.(\big|_{i,j \ s.t. \ t_{i,j}=tag_r} Q'_{i,j})$ admits an attack. Let $\tilde{m}'_r = \tilde{m}_r \setminus \{n_{i,j} \mid t_{i,j} = tag_r\}$. We observe that $\nu \tilde{m}'_r.(\big|_{i,j \ s.t. \ t_{i,j}=tag_r} Q'_{i,j}) \equiv R\{^{\mathsf{h}(c_r,w_r)}/_{w_r}\}$ for some process R such that $\nu w_r.R$ is an instance of $\nu w.(\nu \tilde{m}_{i_1}.P_{i_1} \mid \ldots \mid \nu \tilde{m}_{i_q}.P_{i_q})$ and $\{P_{i_1}, \ldots, P_{i_q}\} \subseteq \{P_1, \ldots P_\ell\}$ (multiset inclusion). Note that this holds because in the transformed protocol each of the roles generates a new nonce, and hence each of the $Q_{i,j}$s can be associated to at most one of the role of $\mathcal{P}$ (two instances of the same role would necessarily generate different tags).

Thanks to Theorem 1 we have that there exists an attack on R which implies that there exists an attack on an instance of $\mathcal{P}$ yielding a contradiction. □