Transforming Password Protocols to Compose

Céline Chevalier, Stéphanie Delaune, and Steve Kremer

October 2011

Research report LSV-11-21

Laboratoire Spécification & Vérification

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

Transforming Password Protocols to Compose

Céline Chevalier[1], Stéphanie Delaune[1], and Steve Kremer[1,2]

1 LSV, ENS Cachan & CNRS & INRIA Saclay Île-de-France, France
2 INRIA Nancy - Grand Est, France

Abstract

Formal, symbolic techniques are extremely useful for modelling and analysing security protocols. They improved our understanding of security protocols, allowed to discover flaws, and 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 of sessions. Our technique also applies to compose different password protocols allowing us to obtain both inter-protocol and inter-session composition.

1998 ACM Subject Classification D.4.6 Security and Protection

Keywords and phrases Security, cryptographic protocols, composition

1 Introduction

Password-based cryptographic protocols are a prominent means to achieve authentication or to establish authenticated, shared session keys, *e.g.* EKE [10], SPEKE [22], or the KOY protocol [23]. 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. However, such passwords are generally *weak* and may be subject to dictionary (also called guessing) attacks. In an *online* dictionary attack an adversary tries to execute the protocol for each possible password. While such 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 the second type of attacks.

It has been widely acknowledged that security protocol design is extremely error prone and rigorous security proofs are a necessity. Formal, symbolic models, in the vein of Dolev and Yao's seminal work [20], provide effective and often automated methods to find errors or prove protocols correct. While most of these methods focus on secrecy and authentication, resistance against offline guessing attacks has been considered in some works [25, 9, 16]. We will in particular focus on an elegant definition of resistance against offline guessing attacks by Corin *et al.* [16] which was introduced in the framework of the applied pi calculus [1] and for which tool support exists [11, 9].

Nowadays, state-of-the-art protocol analysis tools are able to analyse a variety of protocols. However, this analysis is generally carried out in isolation, *i.e.*, analysing one protocol at a time. This is motivated by the fact that in models like the applied pi calculus, security properties, even if shown in isolation, hold in the presence of an arbitrary (public) environment. This is similar to *universal composition* (UC) [14] in computational models. However, these arbitrary environments are public, in the sense that they don't 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 not sharing key material may be a reasonable hypothesis in some cases it is certainly not the case when we compose the same sessions of a same protocol or in a situation where the same password is used in different protocols — it is indeed unreasonable to assume that all users have different passwords for each application.

Our contributions

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. It follows however from our proofs that trace properties such as authentication will also be preserved.

Related Work

In recent years, compositional reasoning has received a lot of attention. Datta *et al.* [18] provide a general strategy whereas our composition result identifies a specific class of protocols that can be composed. In [21, 5, 17], some criteria are given to ensure that parallel and in some works sequential composition is safe. In [6] the issue of composition of sessions of a same protocol is addressed using a transformation similar to the one considered in this paper. None of these works considers password protocols and resistance to guessing attacks. Composition of different password protocols (but not of sessions of the same protocol) using a protocol identifier tag was shown in [19]. In this paper we generalize these results to allow composition of sessions of a same protocol. Moreover, the composition theorem given in [19] only applies to two protocols (and cannot be iterated). This shortcoming was overseen by the authors of [19] and we adapt their result to apply to an arbitrary number of protocols in parallel.

In computational models, Boyko *et al.* [13] 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.* [15] who propose a protocol based on KOY, which is secure in the UC model [14]. This work has been extended to active adversaries [4], group key exchange [3] and to define distributed public-key cryptography from passwords in *e.g.* [2]. 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 [6, 17]. Other forms of tagging were used to ensure termination of a verification procedure [12], safely bound the length of messages [7] or obtain decidability for the verification of some classes of protocols [26].

2 Modeling Protocols

In this section, we recall the cryptographic process calculus defined in [19] for describing protocols. This calculus is a simplified version of the applied pi calculus [1]. In particular we only consider one channel, which is public (*i.e.* under the control of the attacker) and we only consider finite processes, *i.e.* processes without replication.

2.1 Messages

A protocol consists of some agents communicating on a network. The messages sent by the agents are modelled using an abstract term algebra. For this, we assume an infinite set of *names* $\mathcal{N}$, 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. 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 *ground terms* (resp. *terms*) over $\Sigma \cup \mathcal{N}$ (resp. $\Sigma \cup \mathcal{N} \cup \mathcal{X}$). 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 $\mathcal{T}(\Sigma, \mathcal{X})$. The application of a substitution σ to a term T is written $T\sigma$. We also allow *replacement of names by terms*: the term $M\{^N/_n\}$ is the term obtained from M after replacing any occurrence of the name n by the term N (assuming that n does not occur in N). We sometimes abbreviate the sequence of terms $t_1, \dots, t_n$ by $\tilde{t}$ and write $\{^{\tilde{t}}/_{\tilde{x}}\}$ for $\{^{t_1}/x_1, \dots {}^{t_n}/x_n\}$.

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

▶ **Example 1.** Consider the signature $\Sigma = \{\mathsf{sdec}, \mathsf{senc}, \langle\,\rangle, \mathsf{proj}_1, \mathsf{proj}_2, \mathsf{exp}\}$. The function symbols sdec, senc, $\langle\,\rangle$ and exp of arity 2 represent respectively symmetric encryption and decryption, pairing as well as exponentiation. Functions proj_1 and proj_2 of arity 1 model projection of the first and the second component of a pair. As an example that will be useful for modelling the SPEKE protocol [22], we consider the equational theory E, defined by the following equations:

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

Let $T_1 = \mathsf{senc}(\mathsf{proj}_2(\langle a,b\rangle),k)$ and $T_2 = \mathsf{senc}(b,k)$. We have that the terms T_1 and T_2 are equal modulo E, written $T_1 =_{\mathsf{E}} T_2$, while obviously the syntactic equality $T_1 = T_2$ does not hold.

To represent the knowledge of an attacker (who may have observed a sequence of messages $M_1, \ldots, M_\ell$), we use the concept of *frame*. A frame $\phi = \nu\tilde{n}.\sigma$ consists of a finite set $\tilde{n} \subseteq \mathcal{N}$ of *restricted* names (those unknown to the attacker), and a substitution σ of the form $\{^{M_1}/_{z_1}, \ldots, ^{M_\ell}/_{z_\ell}\}$ where each M_i is a ground term. The variables z_i enable an attacker to refer to each M_i. The *domain* of the frame ϕ, written $\mathrm{dom}(\phi)$, is $\mathrm{dom}(\sigma) = \{z_1, \ldots, z_\ell\}$.

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

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

Intuitively, the set of deducible messages is obtained from the messages M_i in ϕ, the names that are not restricted in ϕ, and closed under equality modulo E and application of function symbols.

▶ **Example 3.** Consider the theory E given in Example 1. Let $\phi = \nu b,k.\{^{\mathsf{senc}(b,k)}/_{z_1}, ^{k}/_{z_2}\}$. We have that $\phi \vdash_{\mathsf{E}} k$, $\phi \vdash_{\mathsf{E}} b$ and $\phi \vdash_{\mathsf{E}} a$. Indeed z_2, $\mathsf{sdec}(z_1, z_2)$ and a are recipes of the terms k, b and a respectively.

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

▶ **Definition 4** (static equivalence)**.** We say that two frames $\phi_1 = \nu\tilde{n}.\sigma_1$ and $\phi_2 = \nu\tilde{n}.\sigma_2$ 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 such that $fn(M,N) \cap \tilde{n} = \emptyset$, we have that: $M\sigma_1 =_{\mathsf{E}} N\sigma_1$ if, and only if, $M\sigma_2 =_{\mathsf{E}} N\sigma_2$.

Static equivalence is useful to model the notion of security we consider in this paper, namely *resistance against guessing attacks.* To resist against a guessing attack, the protocol must be designed such that the attacker cannot decide on the basis of the data collected whether his current guess of the password is the actual password or not. Assume $\phi = \nu\tilde{w}.\phi'$ is the frame representing the information gained by the attacker by eavesdropping one or more sessions and let $\tilde{w}$ be the sequence of weak passwords. The frame ϕ is resistant to guessing attacks if the attacker cannot distinguish between a situation in which he guesses the correct passwords $\tilde{w}$ and a situation in which he guesses incorrect ones, say $\tilde{w}'$.

▶ **Definition 5** (frame resistant to guessing attacks)**.** The frame $\nu\tilde{w}.\phi'$ is *resistant to guessing attacks* against the sequence of names $\tilde{w}$ if $\nu\tilde{w}.\phi' \approx \nu\tilde{w}.\nu\tilde{w}'.\phi'\{^{\tilde{w}'}/_{\tilde{w}}\}$ where $\tilde{w}'$ is a sequence of fresh names.

This definition was proposed in [16, 9]. A slightly simpler formulation requiring $\phi' \approx \phi'\{^{\tilde{w}'}/_{\tilde{w}}\}$ (without the name restrictions) was shown equivalent in [19] and will be used in this paper.

▶ **Example 6.** Consider the following protocol where h is a unary function symbol modelling a hash function (no equation on h):

$$\text{A} \rightarrow \text{B} : \mathsf{senc}(n, w) \qquad \text{B} \rightarrow \text{A} : \mathsf{senc}(\mathsf{h}(n), w)$$

An interesting problem arises if the shared key w is a weak secret, *i.e.* vulnerable to brute-force off-line testing. Indeed, the frame representing the knowledge of the attacker at the end of a normal execution of this protocol is $\phi = \nu w.\phi' = \nu w.\nu n.\{^{\mathsf{senc}(n,w)}/_{z_1}, {}^{M}/_{z_2}\}$ where:

$$M = \mathsf{senc}(\mathsf{h}(\mathsf{sdec}(\mathsf{senc}(n, w), w)), w) =_{\mathsf{E}} \mathsf{senc}(\mathsf{h}(n), w).$$

The frame ϕ is not resistant to guessing attacks against the password w. Indeed, the test $\mathsf{h}(\mathsf{sdec}(z_1, w)) \stackrel{?}{=} \mathsf{sdec}(z_2, w)$ is a witness of the non-equivalence $\phi' \not\approx_{\mathsf{E}} \phi'\{^{w'}/_{w}\}$.

2.2 Protocol Language and Semantics

Syntax

The grammar for processes is given below. One has plain processes P, Q, R and extended processes A, B, C that allow the use of active substitutions and restrictions.

$P, Q, R :=$	plain processes	$A, B, C :=$	extended processes
0	null process	P	plain processes
$P \mid Q$	parallel composition	$A \mid B$	parallel composition
$\text{in}(x).P$	message input	$\nu n.A$	restriction
$\text{out}(M).P$	message output	$\{^{M}/_{x}\}$	active substitution
if $M = N$ then P else Q	conditional		

As usual, names and variables have scopes, which are delimited by restrictions and inputs. We write $fv(A)$, $bv(A)$, $fn(A)$, $bn(A)$ for the sets of free and bound variables (resp. names). Moreover, we consider processes such that $bn(A) \cap fn(A) = \emptyset$, $bv(A) \cap fv(A) = \emptyset$, and each name and variable is bound at most once in A. An extended process is *closed* if all free variables are in the domain of an active substitution. 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.

▶ **Example 7.** We illustrate our syntax with the SPEKE protocol (see [22] for a complete description).

$$\begin{array}{ll} \text{A} \rightarrow \text{B} : & M_1 = \mathsf{exp}(w, ra) \\ \text{B} \rightarrow \text{A} : & M_2 = \mathsf{exp}(w, rb) \\ \text{A} \rightarrow \text{B} : & M_3 = \mathsf{senc}(ca, \mathsf{exp}(\mathsf{exp}(w, rb), ra)) \\ \text{B} \rightarrow \text{A} : & M_4 = \mathsf{senc}(\langle ca, cb\rangle, \mathsf{exp}(\mathsf{exp}(w, ra), rb)) \\ \text{A} \rightarrow \text{B} : & M_5 = \mathsf{senc}(cb, \mathsf{exp}(\mathsf{exp}(w, rb), ra)) \end{array}$$

The goal of this protocol is to mutually authenticate A and B with respect to each other, provided that they share an initial secret w. This is done by a simple Diffie-Hellman exchange from a shared secret w, creating a common key $\mathsf{exp}(\mathsf{exp}(w, ra), rb) =_{\mathsf{E}} \mathsf{exp}(\mathsf{exp}(w, rb), ra)$, followed by a challenge-response transaction. The data ra, ca (resp. rb, cb) are nonces that are freshly generated by A (resp. B). In our calculus, we model one session of the protocol as $\nu w.(A \mid B)$:

$$\begin{array}{ll} A = \nu ra, ca.\text{out}(\mathsf{exp}(w, ra)).\text{in}(x_1). & B = \nu rb, cb.\text{in}(y_1).\text{out}(\mathsf{exp}(w, rb)). \\ \quad \text{out}(\mathsf{senc}(ca, ka)).\text{in}(x_2). & \quad \text{in}(y_2).\text{out}(\mathsf{senc}(\langle \mathsf{sdec}(y_2, kb), cb\rangle, kb)). \\ \quad \text{out}(\mathsf{senc}(\mathsf{proj}_2(\mathsf{sdec}(x_2, ka)), ka)) & \quad \text{in}(y_3).\ \text{if } \mathsf{sdec}(y_3, kb) = cb \text{ then } P \text{ else } 0. \end{array}$$

where $ka = \mathsf{exp}(x_1, ra)$, $kb = \mathsf{exp}(y_1, rb)$, and P models an application that is executed when B has been successfully authenticated.

An *evaluation context* is an extended process with a hole instead of an extended process. Given an extended process A we denote by $\phi(A)$ the frame obtained by replacing any embedded plain processes in it with 0.

Semantics

We here only give an informal account of the semantics and refer the reader to [19] for the complete definition. We consider a basic *structural equivalence*, denoted $\equiv$, which includes for instance $A \mid B \equiv B \mid A$, $A \mid 0 \equiv A$ and $\nu n_1, n_2.A \equiv \nu n_2, n_1.A$. In particular, 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.*,

$$A \equiv \nu\tilde{n}.(\{^{M_1}/_{z_1}\} \mid \ldots \mid \{^{M_k}/_{z_k}\} \mid P).$$

Moreover, any frame can be rewritten as $\nu n.\sigma$ matching the notion of frame introduced in Section 2.1.

Labelled operational semantics is the smallest relation $A \xrightarrow{\ell} A'$ between extended processes which is closed under structural equivalence ($\equiv$), application of evaluation context, and a few usual rules for input, output and conditional where ℓ is a label of one of the following forms:

- a label in(M), where M is a ground term such that $\phi(A) \vdash_{\mathsf{E}} M$;
- a label out(M), where M is a ground term, which corresponds to an output of M and which adds an active substitution $\{^{M}/_{z}\}$ in A';
- a label τ corresponding to a silent action (the evaluation of a conditional).

We denote by $\to$ the relation $\left\{\xrightarrow{\ell} \mid \ell \in \{in(M), out(M), \tau\},\ M \in \mathcal{T}(\Sigma)\right\}$ and by $\to^*$ its reflexive and transitive closure. Note that these semantics take the viewpoint that the attacker controls the entire network. Any message is sent to the attacker (who may or not forward it to the intended recipient) and the processes do not have any means to communicate directly.

▶ **Example 8.** We illustrate our semantics with the SPEKE protocol presented in Example 7. The derivation below represents a normal execution of the protocol. For simplicity of this example we suppose that $fv(P) = \emptyset$.

$$\begin{array}{rl}
& \nu w.(A \mid B) \\
\xrightarrow{\text{out}(\mathsf{exp}(w,ra))} & \nu w, ra, ca.(\text{in}(x_1).\text{out}(\mathsf{senc}(ca, ka)).\text{in}(x_2).\ldots \mid \{^{M_1}/_{z_1}\} \mid B) \\
\xrightarrow{\text{in}(\mathsf{exp}(w,ra))} & \nu w, ra, ca, rb, cb.(\text{in}(x_1).\text{out}(\mathsf{senc}(ca, ka)).\text{in}(x_2).\ldots \mid \{^{M_1}/_{z_1}\} \mid B') \\
\to^* & \nu w, ra, ca, rb, cb.(\{^{M_1}/_{z_1}, {}^{M_2}/_{z_2}, {}^{M_3}/_{z_3}, {}^{M_4}/_{z_4}, {}^{M_5}/_{z_5}\} \mid P)
\end{array}$$

where B' represents the remaining actions of B in which y_1 is replaced by $\mathsf{exp}(w, ra)$, and $M_1, \ldots, M_5$ are defined in Example 7. The first step is an output of M_1 performed by A. The active substitution $\{^{M_1}/_{z_1}\}$ allows the environment (*i.e.* the attacker) to access the message M_1 via the handle z_1. The handle z_1 is important since the environment cannot itself describe the term that was output, except by referring to it using z_1. Since M_1 is accessible to the environment via z_1, the next input action can be triggered: we have that $\nu w, ra, ca.\{^{M_1}/_{z_1}\} \vdash_{\mathsf{E}} \mathsf{exp}(w, ra)$ using the the recipe z_1.

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

▶ **Definition 9** (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 $\mathcal{P}$.

The process $\nu w.(A \mid B)$ described in Example 7 is a 2-party password protocol specification with roles A and B. The notion of security we will mainly concentrate on is resistance against guessing attacks.

▶ **Definition 10** (process resistant to guessing attacks)**.** Let A be an extended, closed process and $\tilde{w} \subseteq bn(A)$. We say that a process 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 $\tilde{w}$.

3 Composition Results for Password-based Protocols

In this section, we present several composition results that hold for an arbitrary equational theory E. The only requirement we have is that there exists a function symbol h, which is a free symbol in E, *i.e.* h does not occur in any equation in E. Intuitively, h models a hash function.

3.1 Disjoint State

First, we note that, as usual, composition preserves security properties as soon as protocols have disjoint states, *i.e.*, they do not share any restricted names. Intuitively, this is due to the fact that when other protocols do not share any secrets of the analyzed protocol, then the attacker can completely simulate all messages sent by these other protocols. This has been formally shown in [19].

▶ **Theorem 11.** *[19] Let $A_1, \ldots, A_k$ be k extended processes such that for all i, we have that A_i is resistant to guessing attack against w_i. We have that $A_1 \mid \cdots \mid A_k$ is resistant to guessing attack against $w_1, \ldots, w_k$.*

3.2 Joint State

As soon as two protocols share a restricted name, *e.g.* a password, composition does not necessarily preserve security properties (see [19] for an example). We will use a tagging technique to avoid confusion between messages that come from different protocols. More precisely we will tag each occurrence of a password. Intuitively, we consider protocols that are well-tagged w.r.t. a secret w: all occurrences of w are of the form $\mathsf{h}(t, w)$ for some tag t.

Composing protocols

When each process is well-tagged with a different tag, it can be shown that the processes can be safely composed. One may think of these tags as protocol identifiers, which uniquely identify which protocol is executed, and avoid messages from different protocols to interfere with each other.

▶ **Theorem 12.** *Let $\alpha_1, \ldots, \alpha_k$ be k distinct names, and $\nu w.A_1, \ldots, \nu w.A_k$ be k processes such that $\alpha_i \notin bn(A_i)$ for any $i \in \{1, \ldots, k\}$. If each $\nu w.A_i$ is resistant to guessing attack* *against w then the process $\nu w.(A_1\{^{\mathsf{h}(\alpha_1,w)}/_w\} \mid \cdots \mid A_k\{^{\mathsf{h}(\alpha_k,w)}/_w\})$ is resistant to guessing attack against w.*

Actually, this result is a small adaptation from [19] (the result was shown for $k = 2$ only). This result can also be seen as a consequence of Proposition 15 and Lemma 16 (stated in Section 4) and a theorem showing that adding tags preserves resistance against guessing attacks (this last theorem is stated and proved in [19]).

The previous result is useful to compose distinct protocols. However, when we want to compose different sessions from the same protocol, we cannot assume that participants share a distinct tag for each possible session. In the following, we define a way to dynamically establish such a session tag.

Composing sessions from the same protocol

We now define a protocol transformation which establishes a dynamic tag that will guarantee composition. 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 [8]. 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 13** (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 $\{x_i^j \mid 1 \leqslant i,j \leqslant \ell\}$ be a set of fresh variables. We define the protocol specification $\overline{\mathcal{P}} = \nu w.(\nu\tilde{m}_1, n_1.\overline{P_1} \mid \ldots \mid \nu\tilde{m}_\ell, n_\ell.\overline{P_\ell})$ as follows:

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

where $tag_i = \langle x_i^1, \langle \ldots \langle x_i^{\ell-1}, x_i^\ell\rangle\rangle\rangle$ and $x_i^i = n_i$.

We can now state our composition result for sessions of a same protocol: if a protocol resists against guessing attacks on w then any number of instances of the transformed protocol will also resist to guessing attacks on w.

▶ **Theorem 14.** *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 that is resistant to guessing attacks against w. Let $\mathcal{P}'$ be such that $\overline{\mathcal{P}} = \nu w.\mathcal{P}'$, and $\mathcal{P}'_1, \ldots \mathcal{P}'_p$ be p instances of $\mathcal{P}'$. Then we have that $\nu w.(\mathcal{P}'_1 \mid \ldots \mid \mathcal{P}'_p)$ is resistant to guessing attacks against w.*

Discussion

Note that it is possible to combine these two ways of tagging. Applying successively the two previous theorems we obtain that a tag of the form $\mathsf{h}(\langle n_1, \ldots, n_\ell\rangle, \mathsf{h}(\alpha, w))$ 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 $\mathsf{h}(\langle \alpha, \langle n_1, \ldots, n_\ell\rangle\rangle, w)$ could be used.

The notion of security we consider is resistance to guessing attacks. While generally resistance against guessing attacks is indeed a necessary condition to ensure security properties, this property is not a goal in itself. However, the way we prove our composition results allows us also to ensure that those protocols can be safely composed w.r.t. more classical trace-based security properties such as secrecy or authentication.

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. Note that even though our protocol language does not include replication, our composition results for sessions ensure security for an unbounded number of sessions. Indeed, as any attack requires only a finite number of sessions, any attack on a transformed protocol which is secure for a single instance would yield a contradiction. As deciding resistance to guessing attacks is decidable for a bounded number of sessions (for a large class of equational theories) [9] our result can also be seen as a new decidability result for an unbounded number of sessions on a class of tagged protocols.

4 Proof of our main result

The goal of this section is to give an overview of the proof of Theorem 14. This proof is done in 4 main steps.

Step 1

Assume, by contradiction, that $P = \nu w.(\mathcal{P}'_1 \mid \ldots \mid \mathcal{P}'_p)$ admits a guessing attack on w. Hence there exists an attack derivation $P \rightarrow^* Q$ for some process Q such that $\phi(Q)$ is not resistant to a guessing attack against w.

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.* a process) 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 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). 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 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. 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 exists 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*
- $\nu w_1\ldots\nu w_k.\nu\tilde{n}.A \xrightarrow{\ell_0} \nu w_1\ldots\nu w_k.\nu\tilde{n}.B$*.*

This proposition shows how to map an execution of $P \equiv \nu n_1\ldots\nu n_k\nu w.(A_1\delta_{c_i,t_i}\delta_{w_i,w} \mid \cdots \mid A_k\delta_{c_i,t_i}\delta_{w_i,w})$ (same password) to an execution of $\nu n_1\nu w_1.A_1 \mid \cdots \mid \nu n_k\nu w_k.A_k$ (different password) by maintaining a strong connection between these two derivations. Intuitively, 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.

Note that, except for w, a name that is shared between $A_j\delta_{c_i,t_i}\delta_{w_i,w}$ 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 process. 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.

This proposition is actually sufficient to establish that security properties, like authentication, are preserved by composition. However, to establish resistant against guessing attacks, we need more.

Step 3

We show that if a frame, obtained by executing several protocols that share a same password and that are tagged with terms t_i, is vulnerable to guessing attacks then the frame obtained by the corresponding execution of the protocols with different passwords and tagged with constants c_i is also vulnerable to guessing attacks.

▶ **Lemma 16.** *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* $1 \leqslant i \leqslant k$*, we also assume that* $\nu w.\psi \vdash t_i$*.*

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*.*

The proof of the lemma is technical because mapping all w_i's on the same password can introduce additional equalities between terms. However, each occurrence of the password is tagged, and the purpose of this design is to avoid the introduction of equalities between terms. Again, the lemma holds because the frames are well-tagged.

Thanks to Proposition 15 and Lemma 16 we obtain a guessing attack on the process $\nu n_1\nu w_1.A_1 \mid \cdots \mid \nu n_k\nu w_k.A_k$ against $w_1,\ldots,w_k$.

Step 4

Applying Theorem 11 (combination for disjoint state protocols), 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 2 in [19]. This leads us to a contradiction since we have assumed that $\mathcal{P}$ is resistant to guessing attacks against w.

5 Conclusion

In this paper we propose a transformation for password protocols based on a simple tagging mechanism. This transformation ensures that security is preserved when protocols are composed with other protocols which may use the same password. We show that when protocols are tagged using a simple protocol identifier, we are able to compose different protocols. Computing a dynamic session identifier allows one to also compose different sessions of a same protocol. Hence, it is sufficient to prove that a protocol is secure for one session in order to conclude security under composition.

Currently, as stated, our composition results allow to preserve resistance against offline guessing attacks. As already discussed it also follows from our proofs that trace properties would be preserved. Formalizing for instance preservation of authentication should be a rather straightforward extension. A more ambitious direction for future work would be the composition of more general, indistinguishability properties, expressed in terms of observational equivalence. We also plan to investigate sufficient conditions to ensure composition of protocols in the vein of [24] avoiding to change existing protocols.

Acknowledgements

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 01.

References

1 M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *Proc. 28th Symposium on Principles of Programming Languages (POPL'01)*, pages 104–115. ACM Press, 2001.

2 M. Abdalla, X. Boyen, C. Chevalier, 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*. Springer, 2010.

3 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)*, LNCS. Springer, 2011.

4 M. Abdalla, C. Chevalier, and D. Pointcheval. Smooth projective hashing for conditionally extractable commitments. In *Advances in Cryptology – CRYPTO'09*, volume 5677 of *LNCS*, pages 671–689. Springer, 2009.

5 S. Andova, C. Cremers, K. G. Steen, S. Mauw, S. M. lsnes, and S. Radomirović. Sufficient conditions for composing security protocols. *Information and Computation*, 206(2-4):425–459, 2008.

6 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 *LNAI*, pages 128–142. Springer, 2008.

7 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 *LNCS*, pages 376–387. Springer, 2007.

8 B. Barak, Y. Lindell, and T. Rabin. Protocol initialization for the framework of universal composability. Cryptology ePrint Archive, Report 2004/006, 2004.

9 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, 2005.

10 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.

11 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., 2004.

12 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 *LNCS*, pages 136–152. Springer, 2003.

13 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 *LNCS*, pages 156–171. Springer, 2000.

14 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.

15 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 *LNCS*, pages 404–421. Springer, 2005.

16 R. Corin, J. Doumen, and S. Etalle. Analysing password protocol security against off-line dictionary attacks. *ENTCS*, 121:47–63, 2005.

17 V. Cortier and S. Delaune. Safely composing security protocols. *Formal Methods in System Design*, 34(1):1–36, Feb. 2009.

18 A. Datta, A. Derek, J. Mitchell, and D. Pavlovic. A derivation system and compositional logic for security protocols. *Journal of Computer Security*, 13(3), 2005.

19 S. Delaune, S. Kremer, and M. D. Ryan. Composition of password-based protocols. In *Proc. 21st IEEE Computer Security Foundations Symposium (CSF'08)*, pages 239–251, Pittsburgh, PA, USA, June 2008. IEEE Computer Society Press.

20 D. Dolev and A. Yao. On the security of public key protocols. In *Proc. of the 22nd Symp. on Foundations of Computer Science*, pages 350–357. IEEE Comp. Soc. Press, 1981.

21 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. Press, 2000.

22 D. Jablon. Strong password-only authenticated key exchange. *Computer Communication Review*, 26(5):5–26, 1996.

23 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 *LNCS*, pages 475–494. Springer, 2001.

24 R. Küsters and M. Tuengerthal. Composition Theorems Without Pre-Established Session Identifiers. In *Proc. 18th ACM Conference on Computer and Communications Security (CCS'11)*. ACM Press, 2011. To appear.

25 G. Lowe. Analysing protocols subject to guessing attacks. *Journal of Computer Security*, 12(1):83–98, 2004.

26 R. Ramanujam and S. P. Suresh. Decidability of context-explicit security protocols. *Journal of Computer Security*, 13(1):135–165, 2005.

A 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 $\{^{\mathsf{h}(c_1,w_1)}/_{w_1}\} \ldots \{^{\mathsf{h}(c_k,w_k)}/_{w_k}\}$

A.1 Proof of Lemma 16

The goal of this section is to prove the following lemma.

▶ **Lemma 16.** *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 $1 \leqslant i \leqslant k$, we also assume that $\nu w.\psi \vdash t_i$.*

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.

Before proving this lemma, we introduce the following splitting functions.

▶ **Definition 17.** 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 $1 \leqslant i \leqslant k$;
- $\mathsf{f}(\mathsf{split}_\psi(M_1), \ldots, \mathsf{split}_\psi(M_\ell))$ 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 $1 \leqslant i \leqslant k$;
- $\mathsf{f}(\mathsf{split}_0(M_1), \ldots, \mathsf{split}_0(M_\ell))$ 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 [THESE-baudet07][1]. Hence, we have the following lemma.

▶ **Lemma 18.** *Let split_0 be a ground splitting function as defined in Definition 17. Let M and N be two terms. We have that:*

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

▶ **Lemma 19.** *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:*

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

[1] M. Baudet. Sécurité des protocoles cryptographiques : aspects logiques et calculatoires. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, Jan. 2007.

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 $1 \leqslant i_0 \leqslant k$. 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 16.** *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 $1 \leqslant i \leqslant k$, we also assume that $\nu w.\psi \vdash t_i$.*

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

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

Actually, it is sufficient to establish this result for all terms M and N such that $c_1, \dots, c_k$, $w_1, \dots, w_k$ 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, $c_1, \dots, c_k$, $w_1, \dots, w_k$, $t_1, \dots, t_k$. Let $w'_1, \dots, w'_k$ 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[3]. 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

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

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

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

2. The second point can be proved as follows:

$$\begin{array}{lll} (M =_\mathsf{E} N)\psi & & \\ \Rightarrow & M(\sigma\delta_{c_i,t_i}\delta_{w_i,w}) =_\mathsf{E} N(\sigma\delta_{c_i,t_i}\delta_{w_i,w}) & \text{by def. of } \psi \\ \Rightarrow & \mathsf{split}_0(M(\sigma\delta_{c_i,t_i}\delta_{w_i,w})) =_\mathsf{E} \mathsf{split}_0(N(\sigma\delta_{c_i,t_i}\delta_{w_i,w})) & \text{Lemma 18} \\ \Rightarrow & \mathsf{split}_\psi(M)\sigma =_\mathsf{E} \mathsf{split}_\psi(N)\sigma & \text{Lemma 19} \\ \Rightarrow & (\mathsf{split}_\psi(M) =_\mathsf{E} \mathsf{split}_\psi(N))\phi & \\ & \multicolumn{2}{r}{\text{since } (fn(\mathsf{split}_\psi(N)) \cup fn(\mathsf{split}_\psi(M))) \cap \tilde{n} = \emptyset} \\ \Rightarrow & (\mathsf{split}_\psi(M) =_\mathsf{E} \mathsf{split}_\psi(N))(\phi\delta_{w_i,w'_i}) & \text{since } \phi \approx \phi\delta_{w_i,w'_i} \\ \Rightarrow & \mathsf{split}_\psi(M)(\sigma\delta_{w_i,w'_i}) =_\mathsf{E} \mathsf{split}_\psi(N)(\sigma\delta_{w_i,w'_i}) & \\ \Rightarrow & \mathsf{split}_\psi(M)(\sigma\delta_{w_i,w'_i})\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'} & \\ & =_\mathsf{E} \mathsf{split}_\psi(N)(\sigma\delta_{w_i,w'_i})\delta_{w_i,w}\delta_{c_i,t'_i}\delta_{w'_i,w'} & \\ \Rightarrow & M(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'}) =_\mathsf{E} N(\sigma\delta_{c_i,t'_i}\delta_{w_i,w'}) & \text{item 1 (base case)} \\ \Rightarrow & (M =_\mathsf{E} N)(\psi\{^{w'}/_w\}) & \end{array}$$

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

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

1. To establish the first point, we distinguish two cases.
   - $\mathsf{f} = \mathsf{h}$, $\ell = 2$, $(M_1 =_{\mathsf{E}} \zeta_{i_0})\psi$, 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 $\#_w M_2 \geqslant 1$ (it is not possible to deduce w without using it explicitly). Hence, we can indeed apply our induction hypothesis in order to deduce that:

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

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

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

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. ◀

A.2 Proof of Proposition 15

▶ **Proposition 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. 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 exists 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*
- *$\nu w_1 \ldots \nu w_k.\nu\tilde{n}.A \xrightarrow{\ell_0} \nu w_1 \ldots \nu w_k.\nu\tilde{n}.B$.*

▶ **Lemma 20.** *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 $c_1, \ldots, c_k \notin \mathit{fn}(M')$, and $w \notin \mathit{fn}(M)$. By hypothesis, we have that $\zeta\tilde{\sigma} =_{\mathsf{E}} \tilde{M}$. Thus, thanks to Lemma 18, we have that $M = \mathsf{split}_0(\zeta\tilde{\sigma}) =_{\mathsf{E}} \mathsf{split}_0(\tilde{M})$. Now, thanks to Lemma 19, 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 21.** *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 $w \notin \mathit{fn}(M) \cup \mathit{fn}(N)$;*
- *$M =_{\mathsf{E}} M'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ and $N =_{\mathsf{E}} N'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ for some terms M' and N' such that $c_1, \ldots, c_k \notin \mathit{fn}(M') \cup \mathit{fn}(N')$.*

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 18, 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 15 by induction on the prooftree witnessing the derivation. Since $\rightarrow$ is closed by structural equivalence, we have first to establish a similar result for $\equiv$.

▶ **Lemma 22.** *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 $\mathit{bn}(A) = \emptyset$, $w \notin \mathit{fn}(A)$, and $A =_{\mathsf{E}} A'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ for some A' such that $c_1, \ldots, c_k \notin \mathit{fn}(A')$. 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 $w \notin \mathit{fn}(B)$, and*
- *$B =_{\mathsf{E}} B'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ with $c_1, \ldots, c_k \notin \mathit{fn}(B')$, 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

- $\overline{B} = \overline{C}[\overline{B_1}] = (C\delta_{c_i,t_i}\delta_{w_i,w})[B_1\delta_{c_i,t_i}\delta_{w_i,w}] = B\delta_{c_i,t_i}\delta_{w_i,w}$
- $B = C[B_1] =_{\mathsf{E}} C'[B'_1]\delta_{w_i,\mathsf{h}(c_i,w_i)} = B'\delta_{w_i,\mathsf{h}(c_i,w_i)}$.

◀

Now, we can prove the following proposition.

▶ **Proposition 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. 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 $w, w_1, \ldots, w_k, c_1, \ldots, c_k \notin \tilde{n}$.*

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 $\nu w.\nu\tilde{n}.(A\delta_{c_i,t_i}\delta_{w_i,w}) \rightarrow \overline{B}$. 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 20, 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 $\ell_0 =_{\mathsf{E}} \ell'\delta_{w_i,\mathsf{h}(c_i,w_i)}$;
- $A \rightarrow B$

This will allows us to conclude that $\nu w_1 \ldots \nu w_k.\nu\tilde{n}.A \rightarrow \nu w_1 \ldots \nu w_k.\nu\tilde{n}.B$. 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 $A\delta_{c_i,t_i}\delta_{w_i,w} = \text{out}(\tilde{M}).\tilde{P}$ and $\tilde{B} = \tilde{P} \mid \{^{\tilde{M}}/_x\}$ for some process $\tilde{P}$ and some term $\tilde{M}$. From this, we deduce that $A = \text{out}(M).P$ 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 $A = \text{out}(M).P =_{\mathsf{E}} A'\delta_{w_i,\mathsf{h}(c_i,w_i)}$.
  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 $B = P \mid \{^M/_x\}$, $B' = P' \mid \{^{M'}/_x\}$, $\ell_0 = \text{out}(M)$, and $\ell' = \text{out}(M')$. 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 21.
- ELSE. This case is similar to the previous one.

Inductive cases. The inductive case corresponding to application of structural equivalence directly follows from Lemma 22. 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. ◀

A.3 Proof of Theorem 14

▶ **Theorem 14.** *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 that is resistant to guessing attacks against w. Let $\mathcal{P}'$ be such that $\overline{\mathcal{P}} = \nu w.\mathcal{P}'$, and $\mathcal{P}_1', \ldots \mathcal{P}_p'$ be p instances of $\mathcal{P}'$. 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
$$P_{i,j} = \text{in}(x^1_{i,j}).\ldots\text{in}(x^{j-1}_{i,j}).\text{out}(n_{i,j}).\text{in}(x^{i+1}_{i,j}).\text{in}(x^\ell_{i,j}).P'_{i,j}$$
for some $P'_{i,j}$ ($1 \leqslant i \leqslant p, 1 \leqslant j \leqslant \ell$).
- By contradicition, suppose that P is subject to a guessing attack. Hence there exists Q such that $P \to^* Q$ and $\varphi(Q)$ admits a guessing attack. We assume w.l.o.g. that the derivation is maximal, *i.e.* there is no Q' such that $Q \to Q'$. This allows us to ensure that all the preambles have been executed.
- Hence 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 $P_r \equiv C[\text{in}(x^k_{i,j}).P'] \xrightarrow{\text{in}(M^k_{i,j})} C[P'\{^{M^k_{i,j}}/_{x^k_{i,j}}\}] \equiv P_{r+1}$. Moreover, for each i,j such that $1 \leqslant i \leqslant p, 1 \leqslant j \leqslant \ell$ there exists $y_{i,j} \in \text{dom}(\varphi(Q))$ such that $y_{i,j}\varphi(Q) = n_{i,j}$. 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 $\varphi(Q) \vdash t_{i,j}$ for all i,j such that $1 \leqslant i \leqslant p, 1 \leqslant j \leqslant \ell$. Intuitively, $t_{i,j}$ is the tag which has been computed by process $P_{i,j}$ in the attack trace.
- Let $tag_1, \ldots, tag_k$ be the different terms (modulo E) that occur in $\{t_{i,j} | 1 \leqslant i \leqslant \ell$ and $1 \leqslant j \leqslant p\}$. 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
$$\overline{A_r} = |_{i,j \text{ s.t. } t_{i,j}=tag_r} P_{i,j} \text{ and } \tilde{m}_r = (\cup_{i,j \text{ s.t. } t_{i,j}=tag_r}\tilde{m}_{i,j})$$
We have that $P \equiv \nu w.\nu\tilde{m}_1 \ldots \nu\tilde{m}_k.(\overline{A_1} \mid \ldots \mid \overline{A_k})$.
- Define the process P^* obtained from P by replacing each occurence 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 $P^* \to^* Q$. 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 P^*. We have that $P^* \to^* Q$. As $w_1, \ldots w_k, c_1, \ldots c_k$ do not occur in P^* we assume w.l.o.g. that they do not occur in any label among this derivation. Hence by iterating Proposition 15, we have that there exist two extended processes B and B' such that:
  - $Q \equiv \nu w.\nu\tilde{m}_1.\ldots.\nu\tilde{m}_k.(B\delta_{c_i,t_i}\delta_{w_i,w})$ with $bn(B) = \emptyset$ and $w \notin fn(B)$;
  - $B =_{\mathsf{E}} B'\delta_{w_i,\mathsf{h}(c_i,w_i)}$ with $c_1, \ldots, c_k \notin fn(B')$;
  - $\nu w_1, \ldots w_k, \tilde{m}_1, \ldots \tilde{m}_k.(A_1 \mid \cdots \mid A_k) \to^* \nu w_1, \ldots w_k, \tilde{m}_1, \ldots \tilde{m}_k.B$.

  Moreover as $\varphi(Q)$ admits a guessing attack on w and $\varphi(Q) \vdash tag_i$ for $1 \leqslant i \leqslant k$ we have by Lemma 16 that $\varphi(B)$ admits a guessing attack on $w_1, \ldots w_k$.
- By Theorem 11, we have for some r that $\nu w_r.\nu\tilde{m}_r.A_r$ admits a guessing attack on w_r. 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
$$Q_{i,j} = \text{in}(x^1_{i,j}).\ldots\text{in}(x^{j-1}_{i,j}).\text{out}(n_{i,j}).\text{in}(x^{i+1}_{i,j}).\text{in}(x^\ell_{i,j}).Q'_{i,j}$$
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 $\nu w_r.\nu\tilde{m}_r.(|_{i,j \text{ s.t. } t_{i,j}=tag_r} Q'_{i,j})$ admits a guessing attack on w_r. Let $\tilde{m}'_r = \tilde{m}_r \setminus \{n_{i,j} \mid t_{i,j} = tag_r\}$. We observe that $\nu\tilde{m}'_r.(|_{i,j \text{ s.t. } t_{i,j}=tag_r} Q'_{i,j}) \equiv R\{^{\mathsf{h}(c_r,w_r)}/_{w_r}\}$ 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 2 from [19] we have that there exists a guessing attack on R which implies that there exists a guessing attack on an instance of $\mathcal{P}$ yielding a contradicition. ◀