Composition of Password-based Protocols [*]

Stéphanie Delaune
LSV, ENS Cachan & CNRS & INRIA
France

Steve Kremer
LSV, ENS Cachan & CNRS & INRIA
France

Mark Ryan
School of Computer Science
Birmingham University, UK

Abstract

We investigate the composition of protocols that share a common secret. This situation arises when users employ the same password on different services. More precisely we study whether resistance against guessing attacks composes when the same password is used. We model guessing attacks using a common definition based on static equivalence in a cryptographic process calculus close to the applied pi calculus. We show that resistance against guessing attacks composes in the presence of a passive attacker. However, composition does not preserve resistance against guessing attacks for an active attacker. We therefore propose a simple syntactic criterion under which we show this composition to hold. Finally, we present a protocol transformation that ensures this syntactic criterion and preserves resistance against guessing attacks.

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 [5] and ProVerif [9] 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). But if such data is shared between protocols, then this assumption is not valid.

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 $P_1 \approx S_1$ 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.

[*] This work has been partly supported by the ARA SESUR project AVOTÉ and the following EPSRC projects EP/F033540/1 *Verifying Interoperability Requirements in Pervasive Systems*, EP/D076625/2 *UbiVal: Fundamental Approaches to Validation of Ubiquitous Computing Applications and Infrastructures*, and EP/E040829/1 *Verifying anonymity and privacy properties of security protocols*.

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

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

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 later refers to two different secrets as they have different scope. In contrast with the previously mentioned composition result, this one does not hold in general.

Additionally, the notion of security we consider in this paper is *resistance to guessing attacks*, which is not expressible as observational equivalence to some specification. Guessing attacks are a kind of dictionary attack in which the password is supposed to be weak, i.e. part of a dictionary for which a brute force attack is feasible. A guessing attack works in two phases. In a first phase the attacker eavesdrops or interacts with one or several protocol sessions. In a second *offline* phase, the attacker tries each of the possible passwords on the data collected during the first phase. To resist against a guessing attack, the protocol must be designed such that the attacker cannot discover on the basis of the data collected whether his current guess of the password is the actual password or not. If the attacker's interaction with the protocol during the first phase is limited to eavesdropping, then the attack is called *passive*; if the attacker can participate fully with the protocol, then it is *active.*

Several attempts have been made, based on the initial work of Lowe [22], to characterize guessing attacks [13, 15, 18]. In [14], 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 [7] 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) [10]. 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].

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 [14]). 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 4). 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. Nevertheless, we present a simple syntactic criterion, which we call *well-tagged*, which ensures that security composes even when the same password is reused for different protocols (Section 5). To provide an effective design methodology we also propose a simple transformation to ensure that the protocol is well-tagged. We prove that this transformation preserves resistance against guessing attacks (Section 6). Some of the proofs are omitted but can be found in [19].

Related work. The dangers of ignoring the environment (i.e. all other protocols that may be running concurrently with the protocol in question) when analysing the security of a protocol have been demonstrated in several works (e.g. [12]). The problem of secure composition has also been approached by several authors. Datta *et al.* provide a general strategy [17] whereas our composition result identifies a specific class of protocols that can be composed. In [21, 16], some criteria are given to ensure that parallel composition is safe. Andova *et al.* provide conditions to allow a broader class of composition operations [4].

However, none of these works deal with composing resistance against guessing attacks. They consider secrecy in terms of deducibility or authentication properties. To the best of our knowledge only Malladi *et al.* [23] 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 the environment has to satisfy to prevent multi-protocol guessing attacks. They identify as future work the problem solved in this paper, i.e. the development of techniques of general applicability to prevent multi-protocol guessing attacks. Moreover, their work relies on a definition of guessing attacks due to Lowe [22] which considers a particular set of cryptographic primitives. The results presented here are general and independent of the underlying equational theory. Our work is also related to Canetti *et al.*'s [11] who use a different approach and different model to study universal composability of password-based key exchange protocols.

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 $\mathit{fn}(M)$ (resp. $\mathit{fv}(M)$) 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 every 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 $M = N$ with $M, N \in \mathcal{T}(\Sigma, \mathcal{X})$ and M, N 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* $\Sigma_{\mathsf{enc}} = \{\mathsf{sdec}, \mathsf{senc}, \mathsf{adec}, \mathsf{aenc}, \mathsf{pk}, \langle\,\rangle, \mathsf{proj}_1, \mathsf{proj}_2\}$. *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 *and* proj_2 *are functional symbols of arity 1 that represent public key and projection functions on respectively the first and the second component of a pair. A typical example of an equational theory useful for cryptographic protocols is* $\mathsf{E}_{\mathsf{enc}}$*, defined by the following equations:*

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

Let $T_1 = \mathsf{sdec}(\mathsf{senc}(\mathsf{senc}(n, k_1), k_2), k_2)$ *and* $T_2 = \mathsf{senc}(n, k_1)$*. In this theory, we have that the terms* T_1 *and* T_2 *are equal modulo* $\mathsf{E}_{\mathsf{enc}}$*, written* $T_1 =_{\mathsf{E}_{\mathsf{enc}}} T_2$*, while obviously the syntactic equality* $T_1 = T_2$ *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 in which he observed them. 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. We use the concept of *frame* from the applied pi calculus [2] to represent the knowledge of the attacker. 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 $\{^{M_1}/_{x_1}, \ldots, {}^{M_\ell}/_{x_\ell}\}$. The variables enable us to refer to each M_i. We always assume that the terms M_i are ground. 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 $\{x_1, \ldots, x_\ell\}$.

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 1 (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 $\mathit{fn}(N) \cap \tilde{n} = \emptyset$ 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 $\nu\tilde{n}.\sigma \vdash_{\mathsf{E}} M$, every 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 2 *Consider the theory* $\mathsf{E}_{\mathsf{enc}}$ *given in Example 1. Let* $\phi = \nu k, s_1.\{^{\mathsf{senc}(\langle s_1,s_2\rangle,k)}/_{x_1}, {}^{k}/_{x_2}\}$. *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 $(M =_{\mathsf{E}} N)\phi$. Thus, the knowledge of the attacker can be thought of as his ability to distinguish such recipes. If two frames have identical distinguishing power, then we say that they are *statically equivalent*. Formally:

Definition 2 (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* $\phi =_{\alpha} \nu\tilde{n}.\sigma$, $\tilde{n} \cap (fn(M) \cup fn(N)) = \emptyset$, *and* $M\sigma =_{\mathsf{E}} N\sigma$.

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 3 *Consider again the equational theory* $\mathsf{E}_{\mathsf{enc}}$ *provided in Example 1. Let*

- $\phi = \nu k.\{^{\mathsf{senc}(s_0,k)}/_{x_1}, {}^{k}/_{x_2}\}$, *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* $\phi \not\approx_{\mathsf{E}_{\mathsf{enc}}} \phi'$. *However, as discussed at the beginning of Section 2.4, 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 Modelling Protocols and Guessing Attacks

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., without replication. As we will argue at the end of Section 5 this is not a restriction and our composition result carries 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 are formed from the grammar

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

such that a variable x appears in a term only if the term is in the scope of an input $\mathrm{in}(x)$. 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 $\mathrm{in}(x).P$ is ready to input on the public channel, then to run P with the actual message instead of x, while $\mathrm{out}(M).P$ is ready to output M, then to run P. Again, we omit P when P is 0.

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. 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 *frames* (extending the notion of frame introduced in Section 2.2). Given an extended process A we denote by $\phi(A)$ the frame obtained by replacing any embedded plain processes in it with 0.

Example 4 *Consider the following process:*

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

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

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	$A \mid \nu n.B$	$\equiv$	$\nu n.(A \mid B)$	$n \notin fn(A)$
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 any frame can be rewritten as $\nu n.\sigma$ matching the notion of frame introduced in Section 2.2. 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 5 *Note that in Example 4, 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 (this can always been obtained by* α*-renaming* ϕ*). If* $\phi_1 \approx \phi_2$*, then* $\phi_1 \mid \phi \approx \phi_2 \mid \phi$.

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 $in(M)$, where M is a ground term such that $\phi(A) \vdash_{\mathsf{E}} M$. This corresponds to an input of M;
- a label $out(M)$, where M is a ground term, which corresponds to an output of M;
- 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 $\quad \mathrm{in}(x).P \xrightarrow{in(M)} P\{^M/_x\}$

Out $\quad \mathrm{out}(M).P \xrightarrow{out(M)} P \mid \{^M/_x\}$
where x is a fresh variable

Then $\quad$ if $M = N$ then P else $Q \xrightarrow{\tau} P$
where $M =_{\mathsf{E}} N$

Else $\quad$ if $M = N$ then P else $Q \xrightarrow{\tau} Q$
where $M \neq_{\mathsf{E}} N$

Cont. $$\frac{A \xrightarrow{\ell} B}{C[A] \xrightarrow{\ell} C[B]}$$
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 Cont. 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. 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 $\bigcup \{ \xrightarrow{\ell} \mid \ell \in \{in(M), out(M), \tau\},\ M \in \mathcal{T}(\Sigma)\}$ and by $\rightarrow^*$ its reflexive and transitive closure.

Example 6 *We illustrate our syntax and semantics with the well-known handshake protocol.*

$$\begin{array}{rcll} \mathrm{A} & \rightarrow & \mathrm{B}: & \mathsf{senc}(n, w) \\ \mathrm{B} & \rightarrow & \mathrm{A}: & \mathsf{senc}(f(n), w) \end{array}$$

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 $f(n) = n + 1$) 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 $f(n)$. In our calculus, we model the protocol as $\nu w.(A \mid B)$ where*

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

where P models an application that is executed when B *has been successfully authenticated The derivation described in Figure 1 represents a normal execution of the protocol. For simplicity of this example we suppose that $x \notin fv(P)$.*

3.3 Guessing Attacks

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 weak 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 [7], inspired from the one of [14]. In our definition, we allow multiple shared secrets, and write $\tilde{w}$ for a sequence of such secrets.

Definition 3 *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 and $\tilde{x}$ a sequence of variables such that $\tilde{x} \cap \mathrm{dom}(\phi) = \emptyset$.

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 (in presence of an active attacker). Intuitively, a protocol A is resistant against guessing attacks on a weak password w if it is not possible for an active attacker to mount a guessing attack on it even after some interactions with the protocol during a first phase. In other words, for any process B such that $A \rightarrow^* B$ (note that the attacker can intercept and send messages during this phase), the frame $\phi(B)$ has to be resistant to guessing attack.

Definition 4 *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 7 *Consider the handshake protocol described in Example 6. 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*

$$\nu w.(A \mid B) \rightarrow^* D$$

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

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

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

This corresponds to the classical guessing attack on the handshake protocol (see [20]). After a normal execution of one session of this protocol,

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

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

Figure 1. Example 6

the attacker learns two messages, namely $\mathsf{senc}(n,w)$ *and* $\mathsf{senc}(f(n),w)$*. By decrypting these two messages with his guess x, he can easily test whether $x = w$ and thus recover the weak password w by brute-force testing.*

4 Composition Result – Passive Case

The goal of this section is to establish a composition result in the passive case for resistance against guessing attacks. We first show the equivalence of three definitions of resistance against guessing attacks: the first definition is due to Baudet [7] and the second one is due to Corin *et al.* [14]. The last definition is given in a composable way and establishes our composition result (see Corollary 1).

Proposition 1 *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 3),*
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:

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

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

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

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

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

Now, by relying on Proposition 1 (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 1 (point *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 1): 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 8 *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. 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 [8], which has also been analysed in [14].

Example 9 *The EKE protocol [8] can be informally described by the following 5 steps. A formal description of this protocol in our calculus is given in Figure 2.*

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

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.

We use the equational theory $\mathsf{E}_{\mathsf{enc}}$ *presented in Example 1 to model this protocol. An execution of this protocol in the presence of a passive attacker yields the frame* $\nu w.\phi$ *where*

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

We have that $\nu w.(\phi \mid \{^{w}/_{x}\}) \approx \nu w, w'.(\phi \mid \{^{w'}/_{x}\})$. *We have verified this static equivalence using the YAPA tool [6].*

Corin et al. [14] also analysed one session of this protocol (with a slight difference in the modelling). 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.

5 Composition Result – Active Case

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. Consider the following example.

Example 10 *Consider the processes defined in Figure 2 where the occurrence of* 0 *in* B *has been replaced by* $out(w)$. *Let* A' *and* B' *be these two 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 [10]. However,* $\nu w.(A' \mid B')$ *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 (more complicated but in the same spirit) in which each of the constituent processes admits a complete execution.

Example 11 *Consider the processes* A *and* B *defined in Figure 2 where the occurrences of* 0 *in* A *and* B *have been replaced by* $out(\mathsf{senc}(w, ra))$ *and* $in(y).out(\mathsf{sdec}(y, r))$ *respectively. Let* A_1 *and* B_1 *be these two processes. We can see* $\nu w.(A_1 \mid B)$ *and* $\nu w.(A \mid B_1)$ *as two extensions of the EKE protocol with an additional exchange. Note also that these two protocols admit a normal execution and in this sense are* executable. *We have that* $\nu w.(A_1 \mid B)$ *and* $\nu w.(A \mid B_1)$ *are resistant against guessing attacks on* w. *In particular the additional exchange does not lead to an attack. We have verified these statements*

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

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.

Figure 2. Modelling of the EKE protocol

using the tool ProVerif. However, $\nu w.(A_1 \mid B_1)$ and thus $\nu w.((A_1 \mid B) \mid (A \mid B_1))$, trivially leaks w.

This example shows that there is no hope to obtain a general composition result (even if we restrict to protocols that are executable) that holds even for a particular and relatively simple equational theory. To reach our goal, we consider a restricted class of protocols: the class of *well-tagged* protocols.

5.1 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 $\mathsf{h}(\alpha, w)$. We require that h is a hash function (i.e., has no equations in the equational theory), and α 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). In the remainder, we will consider an arbitrary equational theory E, provided there is no equation for h.

Definition 5 (well-tagged) *Let M be a term and w be a name. We say that M is* α-tagged *w.r.t. w if there exists M' such that $M'\{^{\mathsf{h}(\alpha,w)}/_{w}\} =_{\mathsf{E}} M$.*

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

Other ways of tagging a protocol exist in the literature. For example, in [16] encryption are tagged to ensure that they cannot be used to attack other protocols. That particular method would not work here; on the contrary, that kind of tagging is likely to add guessing attacks.

Example 12 *Let $A = \nu w, s.out(\mathsf{senc}(s, w))$. We have that A is resistant to guessing attacks against w. However, the corresponding well-tagged protocol, according to the definition given in [16], is not. Indeed,*

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

is not resistant to guessing attack against w. The tag α which is publicly known can be used to mount such an attack. An attacker can decrypt the message $\mathsf{senc}(\langle \alpha, s, \rangle, w)$ with his guess x and check whether the first component of the pair is the publicly known value α. Hence, he can test whether $x = w$ and recover the password w by brute force testing.

Another tagging method we considered is to replace w by $\langle \alpha, w \rangle$ (instead of $\mathsf{h}(\alpha, w)$), 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 (see Example 13). 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 13 *Consider the equational theory $\mathsf{E}_{\mathsf{enc}}$.*
Let $A = \nu w, k.out(\mathsf{senc}(w, k)).in(x).$ if
$\mathsf{proj}_1(\mathsf{dec}(x, k)) = \alpha$ then $out(w)$.
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 $\langle \alpha, w \rangle$ is clearly not.

Note that we can built a similar example without using α 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.

5.2 Composition Theorem

We show that any two well-tagged protocols that are separately resistant to guessing attacks can be safely composed provided that they use different tags. The following theorem formalizes the intuition that replacing the shared password with a hash parametrized by the password and a tag is similar to using different passwords which implies composition.

Theorem 1 (composition result) *Let A_1 and A_2 be two well-tagged processes w.r.t. w such that the process A_1 (resp. A_2) is α-tagged (resp. β-tagged) and $\nu w.(A_1 \mid A_2)$ is a process (this can be achieved by using α-renaming).*

If $\nu w.A_1$ and $\nu w.A_2$ are resistant to guessing attacks against w and $\alpha \neq \beta$, then we have that $\nu w.(A_1 \mid A_2)$ is also resistant to guessing attacks against w.

Theorem 1 is proved by contradiction in two main steps. First, we show that separately secure protocols compose safely when no secret is shared, i.e., $\nu w_1.A_1\{^{w_1}/_{w}\} \mid \nu w_2.A_2\{^{w_2}/_{w}\}$ resists against guessing attacks on w_1, w_2. This is rather easy to establish since these two protocols do not share any secret data (Proposition 2).

Proposition 2 *Let A_1 and A_2 be two extended processes such that A_1 (resp. A_2) is resistant to guessing attack against w_1 (resp. w_2) and $A_1 \mid A_2$ is a process. We have that $A_1 \mid A_2$ is resistant to guessing attack against w_1, w_2.*

Now, we show how to map an execution of $\nu w.(A_1\{^{w}/_{w_1}\} \mid A_2\{^{w}/_{w_2}\})$ (same password) to an execution of $\nu w_1.A_1 \mid \nu w_2.A_2$ (different password) by maintaining a strong connection between these two derivations. Intuitively, as A_1 is α-tagged and A_2 is β-tagged we can simply replace $\mathsf{h}(\alpha, w)$ by $\mathsf{h}(\alpha, w_1)$ and $\mathsf{h}(\beta, w)$ by $\mathsf{h}(\beta, w_2)$ in any execution.

Proposition 3 *Let A be an extended process with no occurrence of w in it and such that $w_1, w_2, \alpha, \beta \notin bn(A)$ and $A'\{^{\mathsf{h}(\alpha,w_1)}/_{w_1}\}\{^{\mathsf{h}(\beta,w_2)}/_{w_2}\} =_{\mathsf{E}} A$ for some A'. Let $\overline{B}$ be such that $\nu w.(A\{^{w}/_{w_1}\}\{^{w}/_{w_2}\}) \xrightarrow{\ell} \overline{B}$. Moreover, when $\ell = in(\tilde{M})$ we assume that $w_1, w_2 \notin fn(\tilde{M})$. Then there exists B and B' such that*

- *$\overline{B} \equiv \nu w.(B\{^{w}/_{w_1}\}\{^{w}/_{w_2}\})$ with no occurrence of w in B, and*
- *$B'\{^{\mathsf{h}(\alpha,w_1)}/_{w_1}\}\{^{\mathsf{h}(\beta,w_2)}/_{w_2}\} =_{\mathsf{E}} B$, and*
- *$\nu w_1.\nu w_2.A \rightarrow \nu w_1.\nu w_2.B$.*

Finally, we show that if a frame, obtained by executing two 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. The proof of the lemma is technical because mapping w_1 and w_2 on the same password can introduce additional equalities between terms. Again, the lemma holds because the frames are well-tagged.

Lemma 3 *Let $\phi_1 = \nu\tilde{n}.\sigma_1$ and $\phi_2 = \nu\tilde{n}.\sigma_2$ be two frames such that $\phi_1 \approx \phi_2$, $w_1, w_2, \alpha, \beta \notin \tilde{n}$ and such that $\phi_i =_{\mathsf{E}} \phi'_i\{^{\mathsf{h}(\alpha,w_1)}/_{w_1}\}\{^{\mathsf{h}(\beta,w_2)}/_{w_2}\}$ for some frame ϕ'_i $(i = 1, 2)$. Let w be a fresh name. We have that*

$$\phi_1\{^{w}/_{w_1}\}\{^{w}/_{w_2}\} \approx \phi_2\{^{w}/_{w_1}\}\{^{w}/_{w_2}\}$$

Now, we can prove Theorem 1.

Proof. We prove our composition result by contradiction. Assume that the process $\nu w.(A_1 \mid A_2)$ is not resistant to guessing attacks against w. We show that the process $\nu w_1.A_1\{^{w_1}/_{w}\} \mid \nu w_2.A_2\{^{w_2}/_{w}\}$ is not resistant to guessing attack against w_1, w_2. This means, by Proposition 2, that $\nu w_i.A_i\{^{w_i}/_{w}\}$ is not resistant to guessing attacks against w_i for $i = 1$ or $i = 2$. Thus, by α-renaming, $\nu w.A_i$ is not resistant to guessing attacks against w for $i = 1$ or $i = 2$. Hence, a contradiction.

By definition of guessing attacks, we have that there exists an extended process $\overline{A}$ such that:

- $\nu w.(A_1 \mid A_2) \rightarrow^* \overline{A}$, and
- the frame $\phi(\overline{A})$ is not resistant to guessing attacks against w.

We assume w.l.o.g. that the free names w_1, w_2, which do not occur in $\nu w.(A_1 \mid A_2)$, are not used along the derivation. By iterating Proposition 3, we have that there exist two extended processes A and A' such that:

- $\overline{A} \equiv \nu w.A\{^{w}/_{w_1}\}\{^{w}/_{w_2}\}$,
- $A'\{^{\mathsf{h}(\alpha,w_1)}/_{w_1}\}\{^{\mathsf{h}(\beta,w_2)}/_{w_2}\} =_{\mathsf{E}} A$, and
- $\nu w_1.\nu w_2.(A_1\{^{w_1}/_{w}\} \mid A_2\{^{w_2}/_{w}\}) \rightarrow^* \nu w_1.\nu w_2.A$.

It remains to show that $\phi(A) \not\approx \phi(A)\{^{w'_1}/_{w_1}\}\{^{w'_2}/_{w_2}\}$.

Suppose that $\phi(A) \approx \phi(A)\{^{w'_1}/_{w_1}\}\{^{w'_2}/_{w_2}\}$. Applying Lemma 3 with the replacements $\{^{w}/_{w_1}\}\{^{w}/_{w_2}\}$ and $\{^{w'}/_{w'_1}\}\{^{w'}/_{w'_2}\}$, we obtain that:

- $\phi(A)\{^{w}/_{w_1}\}\{^{w}/_{w_2}\} \approx \phi(A)\{^{w'_1}/_{w_1}\}\{^{w'_2}/_{w_2}\}$, and
- $\phi(A) \approx \phi(A)\{^{w'}/_{w_1}\}\{^{w'}/_{w_2}\}$.

Since $\phi(A) \approx \phi(A)\{^{w'_1}/_{w_1}\}\{^{w'_2}/_{w_2}\}$, by transitivity of $\approx$, we obtain that

$$\phi(A)\{^{w}/_{w_1}\}\{^{w}/_{w_2}\} \approx (\phi(A)\{^{w}/_{w_1}\}\{^{w}/_{w_2}\})\{^{w'}/_{w}\}.$$

Hence, $\phi(\overline{A})$ is resistant to guessing attacks against w. Thus, we obtain a contradiction and conclude the proof. □

Assuming that any attack only uses a finite number of sessions, one may note that our composition result holds for an unbounded number of sessions (even though our protocol language does not include replication). Indeed, 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 1, we have that one of the protocols admits an attack leading to a contradiction.

6 Transformation to Obtain Well-Tagged Protocols

In the previous section, we proved a composition result for protocols that resist against guessing attacks. Unfortunately, it only applies to protocols that are well-tagged. This is indeed a restriction, since 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. If $\nu w.A$ is a process resistant to guessing attacks against w, then the transformed process is defined as $\nu w.(A\{^{\mathsf{h}(\alpha,w)}/_{w}\})$: any occurrence of the password w in A is replaced by $\mathsf{h}(\alpha, w)$. In this section, we show that this transformation is safe in the sense that if a process is resistant to guessing attacks against w, then the transformed process is also resistant to guessing attack against w.

Theorem 2 *Let $A \equiv \nu w.A'$ be a process resistant to guessing attacks against w, then we have that $\nu w.(A'\{^{\mathsf{h}(\alpha,w)}/_{w}\})$ is also resistant to guessing attacks against w.*

Theorem 2 is proved by contradiction in two main steps by relying on Proposition 4 and Lemma 4. In Proposition 4, 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 4 *Let A be a process with $w, \alpha \notin bn(A)$ and $A'\{^{\mathsf{h}(\alpha,w)}/_{w}\} =_{\mathsf{E}} A$ for some A'. If $\nu w.A \rightarrow \overline{B}$, then $\overline{B} \equiv \nu w.B$ and there exists a process B' such that $B'\{^{\mathsf{h}(\alpha,w)}/_{w}\} =_{\mathsf{E}} B$ and $\nu w.A' \rightarrow \nu w.B'$.*

Then, we show that static equivalence is preserved by the transformation $\{^{\mathsf{h}(\alpha,w)}/_{w}\}$. This is crucial to do not introduce guessing attack.

Lemma 4 *Let ϕ_1 and ϕ_2 be two frames such that $\phi_1 \approx \phi_2$. Let w, α be such that $w, \alpha \notin bn(\phi_1) \cup bn(\phi_2)$. We have that*

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

Now, we are able to prove Theorem 2.

Proof. Assume that $\nu w.(A'\{^{\mathsf{h}(\alpha,w)}/_{w}\})$ is not resistant to guessing attacks on w. This means that there exists a process $\overline{B}$ such that:

- $\nu w.(A'\{^{\mathsf{h}(\alpha,w)}/_{w}\}) \rightarrow^* \overline{B}$, and
- the frame $\phi(\overline{B})$ is not resistant to guessing attacks against w.

By applying Proposition 4, we easily obtained that $\overline{B} \equiv \nu w.B$ for some process B and there exists B' such that $B'\{^{\mathsf{h}(\alpha,w)}/_{w}\} =_{\mathsf{E}} B$ and $\nu w.A' \rightarrow^* \nu w.B'$. To conclude, it remains to show that

$$\phi(B') \not\approx \phi(B')\{^{w'}/_{w}\}.$$

Assume that $\phi(B') \approx \phi(B')\{^{w'}/_{w}\}$, thanks to Lemma 4, we easily obtain that

- $\phi(B) =_{\mathsf{E}} \phi(B')\{^{\mathsf{h}(\alpha,w)}/_{w}\}$
  $\approx (\phi(B')\{^{w'}/_{w}\})\{^{\mathsf{h}(\alpha,w)}/_{w}\} = \phi(B')\{^{w'}/_{w}\}$, and
- $\phi(B') = \phi(B')\{^{\mathsf{h}(\alpha,w')}/_{w'}\}$
  $\approx (\phi(B')\{^{w'}/_{w}\})\{^{\mathsf{h}(\alpha,w')}/_{w'}\} =_{\mathsf{E}} \phi(B)\{^{w'}/_{w}\}$.

Since $\phi(B') \approx \phi(B')\{^{w'}/_{w}\}$, and by transitivity of $\approx$, 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. □

We have shown that resistance against guessing attacks is preserved by our transformation. The simplicity of our transformation should also ensure that the functionalities of the protocol are preserved as well. A rigorous proof of this would require a formal definition of what it means to "preserve the functionalities" of a protocol.

7 Conclusion

We investigated the composition of protocols that share a common secret, and answered the question of whether such composition preserves resistance to guessing attacks. In the passive case (where the attacker cannot interact with the protocol but can analyse the transcript of messages it generated), we showed that if the two protocols individually resist guessing attacks, then the composition does too. In the active case, we showed that this result does not hold in general, but we showed that one could tag the protocols in such a way that they compose without compromising the resistance to guessing attacks.

An alternative direction of research would be to investigate whether there are conditions on the equational theory and on the protocols that would make the composition result hold without tagging for the active case. 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 Ştefan Ciobâcă, 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 *Proc. 9th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'06)*, volume 3921 of *LNCS*, pages 398–412. Springer, 2006.

[2] 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.

[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, 1997.

[4] S. Andova, C. Cremers, K. G. Steen, S. Mauw, S. M. lsnes, and S. Radomirović. A framework for compositional verification of security protocols. *Information and Computation*, 206(2-4):425–459, 2007.

[5] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. Heám, 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 ap plications. In *Proc. 17th International Conference on Computer Aided Verification (CAV'05)*, volume 3576 of *LNCS*, 2005.

[6] M. Baudet. YAPA. `http://www.lsv.ens-cachan.fr/~baudet/yapa/`.

[7] 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.

[8] 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.

[9] 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. Press, 2001.

[10] 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.

[11] R. Canetti, S. Halevi, J. Katz, Y. Lindell, and P. D. MacKenzie. Universally composable password-based key exchange. In *Proc. 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'05)*, volume 3494 of *LNCS*, pages 404–421, Aarhus, Denmark, 2005. Springer.

[12] R. Canetti, C. Meadows, and P. F. Syverson. Environmental requirements for authentication protocols. In *Proc. International Symposium on Software Security – Theories and Systems (ISSS'02)*, volume 2609 of *LNCS*, pages 339–355, Tokyo, Japan, 2003. Springer.

[13] E. Cohen. Proving cryptographic protocols safe from guessing attacks. In *Proc. Foundations of Computer Security (FCS'02)*, 2002.

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

[15] 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.

[16] V. Cortier, J. Delaitre, and S. Delaune. Safely composing security protocols. In *Proc. 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07)*, LNCS. Springer, 2007. To appear.

[17] 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.

[18] 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.

[19] S. Delaune, S. Kremer, and M. Ryan. Composition of password-based protocols. Research Report LSV-08-12, Laboratoire Spécification et Vérification, ENS Cachan, France, Mar. 2008. 19 pages.

[20] L. Gong. Increasing availability and security of an authentication service. *IEEE Journal on Selected Areas in Communications*, 11(5):657–662, 1993.

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

[23] 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.