Relating two standard notions of secrecy*

Véronique Cortier[1], Michaël Rusinowitch[1], and Eugen Zălinescu[1]

Loria, UMR 7503 & INRIA Lorraine projet Cassis & CNRS, France

Abstract. Two styles of definitions are usually considered to express that a security protocol preserves the confidentiality of a data s. Reachability-based secrecy means that s should never be disclosed while equivalence-based secrecy states that two executions of a protocol with distinct instances for s should be indistinguishable to an attacker. Although the second formulation ensures a higher level of security and is closer to cryptographic notions of secrecy, decidability results and automatic tools have mainly focused on the first definition so far.
This paper initiates a systematic investigation of situations where syntactic secrecy entails strong secrecy. We show that in the passive case, reachability-based secrecy actually implies equivalence-based secrecy for signatures, symmetric and asymmetric encryption provided that the primitives are probabilistic. For active adversaries in the case of symmetric encryption, we provide sufficient (and rather tight) conditions on the protocol for this implication to hold.

1 Introduction

Cryptographic protocols are small programs designed to ensure secure communications. Since they are widely distributed in critical systems, their security is primordial. In particular, verification using formal methods attracted a lot of attention during this last decade. A first difficulty is to formally express the security properties that are expected. Even a basic property such as confidentiality admits two different acceptable definitions namely reachability-based (*syntactic*) secrecy and equivalence-based (*strong*) secrecy. Syntactic secrecy is quite appealing: it says that the secret is never accessible to the adversary. For example, consider the following protocol where the agent A simply sends a secret s to an agent B, encrypted with B's public key.

$$A \to B : \{\mathtt{s}\}_{\mathsf{pub}(B)}$$

An intruder cannot deduce s, thus s is syntactically secret. Although this notion of secrecy may be sufficient in many scenarios, in others, stronger security requirements are desirable. For instance consider a setting where s is a vote and B behaves differently depending on its value. If the actions of B are observable, s remains syntactically secret but an attacker can learn the values of the vote by watching B's actions. The design of equivalence-based secrecy is targeted at such scenarios and intuitively says that an adversary cannot observe the difference when the value of the secret changes. This

* This work has been partially supported by the ACI-SI Satin and the ACI Jeunes Chercheurs JC9005.

definition is essential to express properties like confidentiality of a vote, of a password, or the anonymity of participants to a protocol.

Although the second formulation ensures a higher level of security and is closer to cryptographic notions of secrecy, so far decidability results and automatic tools have mainly focused on the first definition. The syntactic secrecy preservation problem is undecidable in general [13], it is co-NP-complete for a bounded number of sessions [17], and several decidable classes have been identified in the case of an unbounded number of sessions [13, 10, 7, 16]. These results often come with automated tools, we mention for example ProVerif [5], CAPSL [12], and Avispa [4]. To the best of our knowledge, the only tool capable of verifying strong secrecy is the resolution-based algorithm of ProVerif [6] and only one decidability result is available: Hüttel [14] proves decidability for a fragment of the spi-calculus without recursion for framed bisimilarity, a related equivalence relation introduced by Abadi and Gordon [2]. Also in [8], Borgström *et al* propose an incomplete decision procedure based on a symbolic bisimulation.

In light of the above discussion, it may seem that the two notions of secrecy are separated by a sizable gap from both a conceptual point of view and a practical point of view. These two notions have counterparts in the cryptographic setting (where messages are bitstrings and the adversary is any polynomial probabilistic Turing machine). Intuitively, the syntactic secrecy notion can be translated into a similar reachability-based secrecy notion and the equivalence-based notion is close to indistinguishability. A quite surprising result [11] states that cryptographic syntactic secrecy actually implies indistinguishability in the cryptographic setting. This result relies in particular on the fact that the encryption schemes are probabilistic thus two encryptions of the same plaintext lead to different ciphertexts.

Motivated by the result of [11] and the large number of available systems for syntactic secrecy verification, we initiate in this paper a systematic investigation of situations where syntactic secrecy entails strong secrecy. Surprisingly, this happens in many interesting cases.

We offer results in both passive and active cases in the setting of the *applied pi calculus* [1]. We first treat in Section 2 the case of passive adversaries. We prove that syntactic secrecy is equivalent to strong secrecy. This holds for signatures, symmetric and asymmetric encryption. It can be easily seen that the two notions of secrecy are not equivalent in the case of deterministic encryption. Indeed, the secret $\mathtt{s}$ cannot be deduced from the encrypted message $\{\mathtt{s}\}_{\mathsf{pub}(B)}$ but if the encryption is deterministic, an intruder may try different values for $\mathtt{s}$ and check whether the ciphertext he obtained using B's public key is equal to the one he receives. Thus for our result to hold, we require that encryption is probabilistic. This is not a restriction since this is *de facto* the standard in almost all cryptographic applications. Next, we consider the more challenging case of active adversaries. We give sufficient conditions on the protocols for syntactic secrecy to imply strong secrecy (Section 3). Intuitively, we require that the conditional tests are not performed directly on the secret since we have seen above that such tests provide information on the value of this secret. We again exhibit several counter-examples to motivate the introduction of our conditions. An important aspect of our result is that we do not make any assumption on the number of sessions: we put no restriction on the use of replication.

The interest of our contribution is twofold. First, conceptually, it helps to understand when the two definitions of secrecy are actually equivalent. Second, we can transfer many existing results (and the armada of automatic tools) developed for syntactic secrecy. For instance, since the syntactic secrecy problem is decidable for tagged protocols for an unbounded number of sessions [16], by translating the tagging assumption to the applied-pi calculus, we can derive a first decidability result for strong secrecy for an unbounded number of sessions. Other decidable fragments might be derived from [13] for bounded messages (and nonces) and [3] for a bounded number of sessions.

2 Passive case

Cryptographic primitives are represented by functional symbols. More specifically, we consider the signature $\Sigma = \{\mathsf{enc}, \mathsf{dec}, \mathsf{enca}, \mathsf{deca}, \mathsf{pub}, \mathsf{priv}, \langle\rangle, \pi_1, \pi_2, \mathsf{sign}, \mathsf{check}, \mathsf{retrieve}\}$. $\mathcal{T}(\Sigma, \mathcal{X}, \mathcal{N})$, or simply $\mathcal{T}$, denotes the set of terms built over Σ extended by a set of constants, the infinite set of *names* $\mathcal{N}$ and the infinite set of variables $\mathcal{X}$. A term is *closed* or *ground* if it does not contain any variable. The set of names occurring in a term T is denoted by $\mathrm{fn}(T)$, the set of variables is denoted by $\mathcal{V}(T)$. The *positions* in a term T are defined recursively as usual (*i.e.* as sequences of positive integers), ϵ being the empty sequence. Denote by $\mathbb{N}_+^*$ the set of sequences of positive integers. $\mathrm{Pos}(T)$ denotes the set of positions of T and $\mathrm{Pos_v}(T)$ the set of positions of variables in T. We denote by $T|_p$ the subterm of T at position p and by $U[V]_p$ the term obtained by replacing in U the subterm at position p by V. We may simply say that a term V *is in* a term U if V is a subterm of U. We denote by $\leq_{st}$ (resp.$<_{st}$) the subterm (resp. strict) order. h_U denotes the function symbol, name or variable at position ϵ in the term U.

We equip the signature with an equational theory E:

$$\begin{cases} \pi_1(\langle z_1, z_2\rangle) = z_1 & \mathsf{deca}(\mathsf{enca}(z_1, \mathsf{pub}(z_2), z_3), \mathsf{priv}(z_2)) = z_1 \\ \pi_2(\langle z_1, z_2\rangle) = z_2 & \mathsf{check}(z_1, \mathsf{sign}(z_1, \mathsf{priv}(z_2)), \mathsf{pub}(z_2)) = \mathsf{ok} \\ \mathsf{dec}(\mathsf{enc}(z_1, z_2, z_3), z_2) = z_1 & \mathsf{retrieve}(\mathsf{sign}(z_1, z_2)) = z_1 \end{cases}$$

The function symbols $\pi_1, \pi_2, \mathsf{dec}, \mathsf{deca}, \mathsf{check}$ and $\mathsf{retrieve}$ are called *destructors*. Let $\mathcal{R}_E$ be the corresponding rewrite system (obtained by orienting the equations from left to right). $\mathcal{R}_E$ is convergent. The normal form of a term T w.r.t. $\mathcal{R}_E$ is denoted by $T{\downarrow}$. Notice that E is also stable by substitution of names. As usual, we write $U \to V$ if there exists θ, a position p in U and $L \to R \in \mathcal{R}_E$ such that $U|_p = L\theta$ and $V = U[R\theta]_p$.

The symbol $\langle \_, \_ \rangle$ represents the pairing function and π_1 and π_2 are the associated projection functions. The term $\mathsf{enc}(M, K, R)$ represents the message M encrypted with the key K. The third argument R reflects that the encryption is probabilistic: two encryptions of the same messages under the same keys are different. The symbol dec stands for decryption. The symbols enca and deca are very similar but in an asymmetric setting, where $\mathsf{pub}(a)$ and $\mathsf{priv}(a)$ represent respectively the public and private keys of an agent a. The term $\mathsf{sign}(M, K)$ represents the signature of message M with key K. check enables to verify the signature and $\mathsf{retrieve}$ enables to retrieve the signed message from the signature.[1]

[1] Signature schemes may disclose partial information on the signed message. To enforce the intruder capabilities, we assume that messages can always be retrieved out of the signature.

After the execution of a protocol, an attacker knows the messages sent on the network and also in which order they were sent. Such message sequences are organized as *frames* $\varphi = \nu\widetilde{n}.\sigma$, where $\sigma = \{{}^{M_1}/_{y_1}, \ldots, {}^{M_l}/_{y_l}\}$ is a ground acyclic substitution and $\widetilde{n}$ is a finite set of names. We denote by $\mathrm{dom}(\varphi) = \mathrm{dom}(\sigma) = \{y_1, \ldots, y_l\}$. The variables y_i enable us to refer to each message. The names in $\widetilde{n}$ are said to be *restricted* in φ. Intuitively, these names are *a priori* unknown to the intruder. The names outside $\widetilde{n}$ are said to be *free* in φ. A term M is said *public* w.r.t. a frame $\nu\widetilde{n}.\sigma$ (or w.r.t. a set of names $\widetilde{n}$) if $\mathrm{fn}(M) \cap \widetilde{n} = \emptyset$. The set of restricted names $\widetilde{n}$ might be omitted when it is clear from the context. We usually write $\nu n_1, \ldots, n_k$ instead of $\nu\{n_1, \ldots, n_k\}$.

2.1 Deducibility

Given a frame φ that represents the history of messages sent during the execution of a protocol, we define the *deduction* relation, denoted by $\varphi \vdash M$. Deducible messages are messages that can be obtained from φ by applying functional symbols and the equational theory E.

$$\frac{}{\nu\widetilde{n}.\sigma \vdash x\sigma}\ x \in \mathrm{dom}(\sigma) \qquad \frac{}{\nu\widetilde{n}.\sigma \vdash m}\ m \in \mathcal{N}\backslash\widetilde{n}$$

$$\frac{\nu\widetilde{n}.\sigma \vdash T_1 \quad \cdots \quad \nu\widetilde{n}.\sigma \vdash T_l}{\nu\widetilde{n}.\sigma \vdash f(T_1, \ldots, T_l)} \qquad \frac{\nu\widetilde{n}.\sigma \vdash T \quad T =_E T'}{\nu\widetilde{n}.\sigma \vdash T'}$$

Example 1. k and $\langle k, k'\rangle$ are deducible from the frame $\nu k, k', r.\{{}^{\mathsf{enc}(k,k',r)}/_x, {}^{k'}/_y\}$.

A message is usually said secret if it is not deducible. By opposition to our next notion of secrecy, we say that a term M is *syntactically secret* in φ if $\varphi \nvdash M$.

2.2 Static equivalence

Deducibility does not always suffice to express the abilities of an intruder.

Example 2. The set of deducible messages is the same for the frames $\varphi_1 = \nu k,n_1,n_2,r_1.\{{}^{\mathsf{enc}(n_1,k,r_1)}/_x, {}^{\langle n_1,n_2\rangle}/_y, {}^{k}/_z\}$ and $\varphi_2 = \nu k,n_1,n_2,r_1.\{{}^{\mathsf{enc}(n_2,k,r_2)}/_x, {}^{\langle n_1,n_2\rangle}/_y, {}^{k}/_z\}$, while an attacker is able to detect that the first message corresponds to distinct nonces. In particular, the attacker is able to distinguish the two "worlds" represented by φ_1 and φ_2.

We say that a frame $\varphi = \nu\widetilde{n}.\sigma$ *passes the test* (U, V) where U, V are two terms, denoted by $(U = V)\varphi$, if there exists a renaming of the restricted names in φ such that $(\mathrm{fn}(U) \cup \mathrm{fn}(V)) \cap \widetilde{n} = \emptyset$ and $U\sigma =_E V\sigma$. Two frames $\varphi = \nu\widetilde{n}.\sigma$ and $\varphi' = \nu\widetilde{m}.\sigma'$ are *statically equivalent*, written $\varphi \approx \varphi'$, if they pass the same tests, that is $\mathrm{dom}(\varphi) = \mathrm{dom}(\varphi')$ and for all terms U, V such that $(\mathcal{V}(U) \cup \mathcal{V}(V)) \subseteq \mathrm{dom}(\varphi)$ and $(\mathrm{fn}(U) \cup \mathrm{fn}(V)) \cap (\widetilde{n} \cup \widetilde{m}) = \emptyset$, we have $(U = V)\varphi$ iff $(U = V)\varphi'$.

Example 3. The frames φ_1 and φ_2 defined in Example 2 are not statically equivalent since $(\mathsf{dec}(x, z) = \pi_1(y))\varphi_1$ but $(\mathsf{dec}(x, z) \neq \pi_1(y))\varphi_2$.

Let $\mathtt{s}$ be a free name of a frame $\varphi = \nu\widetilde{n}.\sigma$. We say that $\mathtt{s}$ is *strongly secret* in φ if for every closed public terms M, M' w.r.t. φ, we have $\varphi(^{M}/_{\mathtt{s}}) \approx \varphi(^{M'}/_{\mathtt{s}})$ that is, the intruder cannot distinguish the frames obtained by instantiating the secret $\mathtt{s}$ by two terms of its choice. For simplicity we may omit $\mathtt{s}$ and write $\varphi(M)$ instead of $\varphi(^{M}/_{\mathtt{s}})$.

Of course an intended syntactical secret name $\mathtt{s}$ must be restricted, but when talking about instances of $\mathtt{s}$ we must consider it (at least) a free name (if not a variable). Hence we compare syntactic secrecy and strong secrecy regarding the same frame modulo the restriction on the secret $\mathtt{s}$. We use the notation $\nu\mathtt{s}.\varphi$ for $\nu(\widetilde{n} \cup \{\mathtt{s}\}).\sigma$, where $\varphi = \nu\widetilde{n}.\sigma$. Thus $\mathtt{s}$ is syntactically secret if $\nu\mathtt{s}.\varphi \nvdash \mathtt{s}$.

2.3 Syntactic secrecy implies strong secrecy

Syntactic secrecy is usually weaker than strong secrecy! We first exhibit some examples of frames that preserves syntactic secrecy but not strong secrecy. They all rely on different properties.

Probabilistic encryption. The frame $\psi_1 = \nu k, r.\{^{\mathsf{enc}(\mathtt{s},k,r)}/_{x}, ^{\mathsf{enc}(n,k,r)}/_{y}\}$ does not preserve the strong secrecy of $\mathtt{s}$. Indeed, $\psi_1(n) \not\approx \psi_1(n')$ since $(x = y)\psi_1(n)$ but $(x \neq y)\psi_1(n')$. This would not happen if each encryption used a distinct randomness, that is, if the encryption was probabilistic.

Key position. The frame $\psi_2 = \nu n.\{^{\mathsf{enc}(\langle n,n'\rangle,\mathtt{s},r)}/_{x}\}$ does not preserve the strong secrecy of $\mathtt{s}$. Indeed, $\psi_2(k) \not\approx \psi_2(k')$ since $(\pi_2(\mathsf{dec}(x,k)) = n')\,\psi_2(k)$ but $(\pi_2(\mathsf{dec}(x,k)) \neq n')\,\psi_2(k')$. If $\mathtt{s}$ occurs in key position in some ciphertext, the intruder may try to decrypt the ciphertext since $\mathtt{s}$ is replaced by public terms and check for some redundancy. It may occur that the encrypted message does not contain any verifiable part. In that case, the frame may preserve strong secrecy. It is for example the case for the frame $\nu n.\{^{\mathsf{enc}(n,\mathtt{s},r)}/_{x}\}$. Such cases are however quite rare in practice.

No destructors. The frame $\psi_3 = \{^{\pi_1(\mathtt{s})}/_{x}\}$ does not preserve the strong secrecy of $\mathtt{s}$ simply because $(x = k)$ is true for $\psi_3(\langle k, k'\rangle)$ while not for $\psi_3(k)$.

Retrieve rule. The $\mathsf{retrieve}(\mathsf{sign}(z_1, z_2)) = z_1$ may seem arbitrary since not all signature schemes enable to get the signed message out of a signature. It is actually crucial for our result. For example, the frame $\psi_4 = \{^{\mathsf{sign}(\mathtt{s},\mathsf{priv}(a))}/_{x}, ^{\mathsf{pub}(a)}/_{y}\}$ does not preserve the strong secrecy of $\mathtt{s}$ because $(\mathsf{check}(n,x,y) = \mathsf{ok})$ is true for $\psi_4(n)$ but not for $\psi_4(n')$.

In these four cases, the frames preserve the syntactic secrecy of $\mathtt{s}$, that is $\nu\mathtt{s}.\psi_i \nvdash \mathtt{s}$, for $1 \leq i \leq 4$. This leads us to the following definition.

Definition 1. *A frame $\varphi = \nu\widetilde{n}.\sigma$ is* well-formed *w.r.t. some name $\mathtt{s}$ if*

1. *Encryption is probabilistic, i.e. for any subterm $\mathsf{enc}(M,K,R)$ of φ, for any term $T \in \varphi$ and position p such that $T|_p = R$ we have $p = q.3$ for some q and $T|_q = \mathsf{enc}(M,K,R)$. In addition, if $\mathtt{s}$ occurs in M at a position p' such that no encryption appears along the path from the root to p' then R must be restricted, that is $R \in \widetilde{n}$. The same conditions hold for asymmetric encryption. and*
2. *$\mathtt{s}$ is not part of a key, i.e. for all $\mathsf{enc}(M,K,R)$, $\mathsf{enca}(M',K',R')$, $\mathsf{sign}(U,V)$, $\mathsf{pub}(W)$, $\mathsf{priv}(W')$ subterms of φ, $\mathtt{s} \notin \mathrm{fn}(K,K',V,W,W',R,R')$.*
3. *φ does not contain destructor symbols.*

Condition 1 requires that each innermost encryption above $\mathtt{s}$ contains a restricted randomness. This is not a restriction since $\mathtt{s}$ is meant to be a secret value and such encryptions have to be produced by honest agents and thus contain a restricted randomness.

For well-formed frames, syntactic secrecy is actually equivalent to strong secrecy.

Theorem 1. *Let φ be a well-formed frame w.r.t. $\mathtt{s}$, where $\mathtt{s}$ is a free name in φ.*

$$\nu\mathtt{s}.\varphi \not\vdash \mathtt{s} \text{ if and only if } \varphi(^{M}/_{\mathtt{s}}) \approx \varphi(^{M'}/_{\mathtt{s}})$$

for all M, M' closed public terms w.r.t. φ.

Proof. We present the skeleton of the proof; all details can be found in a technical report [18]. Let $\varphi = \nu\widetilde{n}.\sigma$ be a well-formed frame w.r.t. $\mathtt{s}$. If $\nu\mathtt{s}.\varphi \vdash \mathtt{s}$, this trivially implies that $\mathtt{s}$ is not strongly secret. Indeed, there exists a public term T w.r.t. φ such that $T\sigma =_E \mathtt{s}$ (this can be easily shown by induction on the deduction system). Let n_1, n_2 be fresh names such that $n_1, n_2 \notin \widetilde{n}$ and $n_1, n_2 \notin \mathrm{fn}(\varphi)$. Since $T\sigma(^{n_1}/_{\mathtt{s}}) =_E n_1$ the frames $\varphi(^{n_1}/_{\mathtt{s}})$ and $\varphi(^{n_2}/_{\mathtt{s}})$ are distinguishable with the test $(T = n_1)$.

We assume now that $\nu\mathtt{s}.\varphi \not\vdash \mathtt{s}$. We first show that any syntactic equality satisfied by the frame $\varphi(^{M}/_{\mathtt{s}})$ is already satisfied by φ.

Lemma 1. *Let $\varphi = \nu\widetilde{n}.\sigma$ be a well-formed frame w.r.t. a free name $\mathtt{s}$, U, V terms such that $\mathcal{V}(U), \mathcal{V}(V) \subseteq \mathrm{dom}(\varphi)$ and M a closed term, U, V and M public w.r.t. $\widetilde{n}$. If $\nu\mathtt{s}.\varphi \not\vdash \mathtt{s}$ then $U\sigma(^{M}/_{\mathtt{s}}) = V\sigma(^{M}/_{\mathtt{s}})$ implies $U\sigma = V\sigma$. Let T be a subterm of a term in σ that does not contain $\mathtt{s}$. If $\nu\mathtt{s}.\varphi \not\vdash \mathtt{s}$ then $T = V\sigma(^{M}/_{\mathtt{s}})$ implies $T = V\sigma$.*

The key lemma is that any reduction that applies to a deducible term U where $\mathtt{s}$ is replaced by some M, directly applies to U.

Lemma 2. *Let $\varphi = \nu\widetilde{n}.\sigma$ be a well-formed frame w.r.t. a free name $\mathtt{s}$ such that $\nu\mathtt{s}.\varphi \not\vdash \mathtt{s}$. Let U be a term with $\mathcal{V}(U) \subseteq \mathrm{dom}(\varphi)$ and M be a closed term in normal form, U and M public w.r.t. $\widetilde{n}$. If $U\sigma(^{M}/_{\mathtt{s}}) \to V$, for some term V, then there exists a well-formed frame $\varphi' = \nu\widetilde{n}.\sigma'$ w.r.t. $\mathtt{s}$*

- *extending φ, that is $x\sigma' = x\sigma$ for all $x \in \mathrm{dom}(\sigma)$,*
- *preserving deducible terms: $\nu\mathtt{s}.\varphi \vdash W$ iff $\nu\mathtt{s}.\varphi' \vdash W$,*
- *and such that $V = V'\sigma'(^{M}/_{\mathtt{s}})$ and $U\sigma \to V'\sigma'$ for some V' public w.r.t. $\widetilde{n}$.*

This lemma allows us to conclude the proof of Theorem 1. Fix arbitrarily two public closed terms M, M'. We can assume w.l.o.g. that M and M' are in normal form. Let $U \neq V$ be two public terms such that $\mathcal{V}(U), \mathcal{V}(V) \subseteq \mathrm{dom}(\varphi)$ and $U\sigma(^{M}/_{\mathtt{s}}) =_E V\sigma(^{M}/_{\mathtt{s}})$. Then there are $U_1, \ldots, U_k$ and $V_1, \ldots, V_l$ such that $U\sigma(^{M}/_{\mathtt{s}}) \to U_1 \to \ldots \to U_k$, $V\sigma(^{M}/_{\mathtt{s}}) \to V_1 \to \ldots \to V_l$, $U_k = U\sigma(^{M}/_{\mathtt{s}})\!\downarrow$, $V_l = V\sigma(^{M}/_{\mathtt{s}})\!\downarrow$ and $U_k = V_l$.

Applying repeatedly Lemma 2 we obtain that there exist public terms $U'_1, \ldots, U'_k$ and $V'_1, \ldots, V'_l$ and well-formed frames $\varphi^{u_i} = \nu\widetilde{n}.\sigma^{u_i}$, for $i \in \{1, \ldots, k\}$ and $\varphi^{v_j} = \nu\widetilde{n}.\sigma^{v_j}$, for $j \in \{1, \ldots, l\}$ (as in the lemma) such that $U_i = U'_i\sigma^{u_i}(^{M}/_{\mathtt{s}})$, $U'_i\sigma^{u_i} \to U'_{i+1}\sigma^{u_{i+1}}$, $V_j = V'_j\sigma^{v_j}(^{M}/_{\mathtt{s}})$ and $V'_j\sigma^{v_j} \to V'_{j+1}\sigma^{v_{j+1}}$.

We consider $\varphi' = \nu\widetilde{n}.\sigma'$ where $\sigma' = \sigma^{u_k} \cup \sigma^{v_l}$. Since only subterms of φ have been added to φ', it is easy to verify that φ' is still a well-formed frame and for every term W, $\nu\mathtt{s}.\varphi \vdash W$ iff $\nu\mathtt{s}.\varphi' \vdash W$. In particular $\nu\mathtt{s}.\varphi' \not\vdash \mathtt{s}$.

By construction we have that $U'_k\sigma^{u_k}(^{M}/_{\mathtt{s}}) = V'_l\sigma^{v_l}(^{M}/_{\mathtt{s}})$. Then, by Lemma 1, we deduce that $U'_k\sigma^{u_k} = V'_l\sigma^{v_l}$ that is $U\sigma =_E V\sigma$. By stability of substitution of names, we have $U\sigma(^{M'}/_{\mathtt{s}}) =_E V\sigma(^{M'}/_{\mathtt{s}})$. We deduce that $\varphi(^{M}/_{\mathtt{s}}) \approx \varphi(^{M'}/_{\mathtt{s}})$.

3 Active case

To simplify the analysis of the active case, we restrict our attention to pairing and symmetric encryption: the alphabet Σ is now reduced to $\Sigma = \{\mathsf{enc}, \mathsf{dec}, \langle\rangle, \pi_1, \pi_2\}$ and E is limited to the first three equations.

3.1 Modeling protocols within the applied pi calculus

The applied pi calculus [1] is a process algebra well-suited for modeling cryptographic protocols, generalizing the spi-calculus [2]. We briefly describe its syntax and semantics. This part is mostly borrowed from [1].

Processes, also called plain processes, are defined by the grammar:

P, Q := processes			
$\mathbf{0}$	null process	$\nu n.P$	name restriction
$P \mid Q$	parallel composition	$u(z).P$	message input
$!P$	replication	$\overline{u}\langle M\rangle.P$	message output
$if\ M = N\ then\ P\ else\ Q$	conditional		

where n is a name, U, V are terms, and u is a name or a variable. The null process $\mathbf{0}$ does nothing. Parallel composition executes the two processes concurrently. Replication $!P$ creates unboundedly new instances of P. Name restriction $\nu n.P$ builds a new, private name n, binds it in P and then executes P. The conditional $if\ M = N\ then\ P\ else\ Q$ behaves like P or Q depending on the result of the test $M = N$. If Q is the null process then we use the notation $[M = N].P$ instead. Finally, the process $u(z).P$ inputs a message and executes P binding the variable z to the received message, while the process $\overline{u}\langle M\rangle.P$ outputs the message M and then behaves like P. We may omit P if it is $\mathbf{0}$. In what follows, we restrict our attention to the case where u is a name since it is usually sufficient to model cryptographic protocols.

Extended processes are defined by the grammar:

A, B := extended processes			
P	plain process	$\nu n.A$	name restriction
$A \mid B$	parallel composition	$\nu x.A$	variable restriction
$\{^M/_x\}$	active substitution		

Active substitutions generalize let, in the sense that $\nu x.(\{^M/_x\}|P)$ corresponds to $let\ x = M\ in\ P$, while unrestricted, $\{^M/_x\}$ behaves like a permanent knowledge, permitting to refer globally to M by means of x. We identify variable substitutions $\{^{M_1}/_{x_1}, \ldots, {}^{M_l}/_{x_l}\}$, $l \geq 0$ with extended processes $\{^{M_1}/_{x_1}\}|\ldots|\{^{M_l}/_{x_l}\}$. In particular the empty substitution is identified with the null process.

We denote by $\mathrm{fv}(A)$, $\mathrm{bv}(A)$, $\mathrm{fn}(A)$, and $\mathrm{bn}(A)$ the sets of free and bound variables and free and bound names of A, respectively, defined inductively as usual for the pi calculus' constructs and using $\mathrm{fv}(\{^M/_x\}) = \mathrm{fv}(M) \cup \{x\}$ and $\mathrm{fn}(\{^M/_x\}) = \mathrm{fn}(M)$ for active substitutions. An extended process is *closed* if it has no free variables except those in the domain of active substitutions.

Extended processes built up from the null process (using the given constructions, that is, parallel composition, restriction and active substitutions) are called *frames*[2]. To every extended process A we associate the frame $\varphi(A)$ obtained by replacing all embedded plain processes with $\mathbf{0}$.

An *evaluation context* is an extended process with a hole not under a replication, a conditional, an input or an output.

Structural equivalence ($\equiv$) is the smallest equivalence relation on extended processes that is closed by α-conversion of names and variables, by application of evaluation contexts and such that the standard structural rules for the null process, parallel composition and restriction (such as associativity and commutativity of $|$, commutativity and binding-operator-like behavior of ν) together with the following ones hold.

$$\begin{array}{rll} \nu x.\{^{M}/_{x}\} \equiv \mathbf{0} & & \text{ALIAS} \\ \{^{M}/_{x}\} \mid A \equiv \{^{M}/_{x}\} \mid A\{^{M}/_{x}\} & & \text{SUBST} \\ \{^{M}/_{x}\} \equiv \{^{N}/_{x}\} & \text{if } M =_E N & \text{REWRITE} \end{array}$$

If $\widetilde{n}$ represents the (possibly empty) set $\{n_1, \ldots, n_k\}$, we abbreviate by $\nu\widetilde{n}$ the sequence $\nu n_1.\nu n_2 \ldots \nu n_k$. Every closed extended process A can be brought to the form $\nu\widetilde{n}.\{^{M_1}/_{x_1}\}|\ldots|\{^{M_l}/_{x_l}\}|P$ by using structural equivalence, where P is a plain closed process, $l \geq 0$ and $\{\widetilde{n}\} \subseteq \cup_i \mathrm{fn}(M_i)$. Hence the two definitions of frames are equivalent up to structural equivalence on closed extended processes. To see this we apply rule SUBST until all terms are ground (this is assured by the fact that the considered extended processes are closed and the active substitutions are cycle-free). Also, another consequence is that if $A \equiv B$ then $\varphi(A) \equiv \varphi(B)$.

Two semantics can be considered for this calculus, defined by structural equivalence and by *internal reduction* and *labeled reduction*, respectively. These semantics lead to *observational equivalence* (which is standard and not recalled here) and *labeled bisimilarity* relations. The two bisimilarity relations are equal [1]. We use here the latter since it relies on static equivalence and it allows to take implicitly into account the adversary, hence having the advantage of not using quantification over contexts.

Internal reduction is the largest relation on extended processes closed by structural equivalence and application of evaluation contexts such that:

$$\begin{array}{rcll} \overline{c}\langle x\rangle.P \mid c(x).Q & \rightarrow & P \mid Q & \text{COMM} \\ \mathit{if}\ M = M\ \mathit{then}\ P\ \mathit{else}\ Q & \rightarrow & P & \text{THEN} \\ \mathit{if}\ M = N\ \mathit{then}\ P\ \mathit{else}\ Q & \rightarrow & Q & \text{ELSE} \end{array}$$

for any ground terms M and N such that $M \neq_E N$

On the other hand, *labeled reduction* is defined by the rules of Figure 1.

Definition 2. Labeled bisimilarity *($\approx_l$) is the largest symmetric relation $\mathcal{R}$ on closed extended processes such that $A\,\mathcal{R}\,B$ implies:*

1. *$\varphi(A) \approx \varphi(B)$;*
2. *if $A \rightarrow A'$ then $B \rightarrow^* B'$ and $A'\,\mathcal{R}\,B'$, for some B';*

[2] We see later in this section why we use the same name as for the notion defined in section 2.

$c(x).P \xrightarrow{c(M)} P\{^{M}/_{x}\}$ IN $\quad \overline{c}\langle u\rangle.P \xrightarrow{\overline{c}\langle u\rangle} P$ OUT-ATOM

$\dfrac{A \xrightarrow{\overline{c}\langle u\rangle} A'}{\nu u.A \xrightarrow{\nu u.\overline{c}\langle u\rangle} A'}\ u \neq c$ OPEN-ATOM $\quad \dfrac{A \xrightarrow{\alpha} A'}{\nu u.A \xrightarrow{\alpha} \nu u.A'}\ u$ does not occur in α SCOPE

$\dfrac{A \xrightarrow{\alpha} A'}{A|B \xrightarrow{\alpha} A'|B}$ (*) PAR $\quad \dfrac{A \equiv B \quad B \xrightarrow{\alpha} B' \quad B' \equiv A'}{A \xrightarrow{\alpha} A'}$ STRUCT

where c is a name and u is a metavariable that ranges over names and variables, and the condition (*) of the rule PAR is $\mathrm{bv}(\alpha) \cap \mathrm{fv}(B) = \mathrm{bn}(\alpha) \cap \mathrm{fn}(B) = \emptyset$.

Fig. 1. Labeled reduction.

3. *if* $A \xrightarrow{\alpha} A'$ *and* $\mathrm{fv}(\alpha) \subseteq \mathrm{dom}(\varphi(A))$ *and* $\mathrm{bn}(\alpha) \cap \mathrm{fn}(B) = \emptyset$ *then* $B \rightarrow^* \xrightarrow{\alpha} \rightarrow^* B'$ *and* $A' \mathcal{R} B'$*, for some* B'*.*

We denote $A \Rightarrow B$ if $A \rightarrow B$ or $A \xrightarrow{\alpha} B$.

Definition 3. *A frame* φ *is* valid *w.r.t. a process* P *if there is* A *such that* $P \Rightarrow^* A$ *and* $\varphi \equiv \varphi(A)$.

Definition 4. *Let* P *be a closed plain process without variables as channels and* $\mathtt{s}$ *a free name of* P*, but not a channel name. We say that* $\mathtt{s}$ *is* syntactically secret *in* P *if, for every valid frame* φ *w.r.t.* P*,* $\mathtt{s}$ *is not deducible from* $\nu\mathtt{s}.\varphi$*. We say that* $\mathtt{s}$ *is* strongly secret *if for any closed terms* M, M' *such that* $\mathrm{bn}(P) \cap (\mathrm{fn}(M) \cup \mathrm{fn}(M')) = \emptyset$*,* $P(^{M}/_{\mathtt{s}}) \approx_l P(^{M'}/_{\mathtt{s}})$.

Let $\mathcal{M}_o(P)$ be the set of *outputs* of P, that is the set of terms m such that $\overline{c}\langle m\rangle$ is a message output construct for some channel name c in P, and let $\mathcal{M}_t(P)$ be the set of *operands of tests* of P, where a *test* is a couple $M = N$ occurring in a conditional and its *operands* are M and N. Let $\mathcal{M}(P) = \mathcal{M}_o(P) \cup \mathcal{M}_t(P)$ be the set of *messages* of P. Examples are provided at the end of this section.

The following lemma intuitively states that any message contained in a valid frame is an output instantiated by messages deduced from previous sent messages.

Lemma 3. *Let* P *be a closed plain process, and* A *be a closed extended process such that* $P \Rightarrow^* A$*. There are* $l \geq 0$*, an extended process* $B = \nu\widetilde{n}.\sigma_l|P_B$*, where* P_B *is some plain process, and* θ *a substitution public w.r.t.* $\widetilde{n}$ *such that:* $A \equiv B$*,* $\{\widetilde{n}\} \subseteq \mathrm{bn}(P)$*, for every operand of a test or an output* T *of* P_B *there is a message* T_0 *in* P *(a operand of a test or an output respectively), such that* $T = T_0\theta\sigma_l$*, and,* $\sigma_i = \sigma_{i-1} \cup \{^{M_i\theta_i\sigma_{i-1}}/_{y_i}\}$*, for all* $1 \leq i \leq l$*, where* M_i *is an output in* P*,* θ_i *is a substitution public w.r.t.* $\widetilde{n}$ *and* σ_0 *is the empty substitution.*

The proof is done by induction on the number of reductions in $P \Rightarrow^* A$. Intuitively, B is obtained by applying the SUBST rule (from left to right) as much as possible until there are no variables left in the plain process. Note that B is unique up to the structural

rules different from ALIAS, SUBST and REWRITE. We say that $\varphi(B)$ is the *standard frame* w.r.t. A.

As a running example we consider the Yahalom protocol:

$$\begin{array}{l} A \Rightarrow B:\ A, N_a \\ B \Rightarrow S:\ B, \{A, N_a, N_b\}_{K_{bs}} \\ S \Rightarrow A:\ \{B, K_{ab}, N_a, N_b\}_{K_{as}}, \{A, K_{ab}\}_{K_{bs}} \\ A \Rightarrow B:\ \{A, K_{ab}\}_{K_{bs}} \end{array}$$

In this protocol, two participants A and B wish to establish a shared key K_{ab}. The key is created by a trusted server S which shares the secret keys K_{as} and K_{bs} with A and B respectively. The protocol is modeled by the following process:

$$P_Y(k_{ab}) = \nu k_{as}, k_{bs}.(!P_A)|(!P_B)|(!\nu k.P_S(k))|P_S(k_{ab}) \quad \text{with}$$

$$\begin{array}{l} P_A = \nu n_a.\overline{c}\langle a, n_a\rangle.c(z_a).[b = U_b].[n_a = U_{n_a}].\overline{c}\langle \pi_2(z_a)\rangle \\ P_B = c(z_b).\nu n_b, r_b.\overline{c}\langle b, \mathsf{enc}(\langle \pi_1(z_b), \langle \pi_2(z_b), n_b\rangle\rangle, k_{bs}, r_b)\rangle.c(z'_b).[a = \pi_1(\mathsf{dec}(z'_b, k_{bs}))] \\ P_S(x) = c(z_s).\nu r_s, r'_s.\overline{c}\langle \mathsf{enc}(\langle \pi_1(z_s), \langle x, V_n\rangle\rangle, k_{as}, r_s), \mathsf{enc}(\langle V_a, x\rangle, k_{bs}, r'_s)\rangle \end{array}$$

$$\begin{array}{lll} \text{and} & U_b = \pi_1(\mathsf{dec}(\pi_1(z_a), k_{as})) & U_{n_a} = \pi_1(\pi_2(\pi_2(\mathsf{dec}(\pi_1(z_a), k_{as})))) \\ & V_a = \pi_1(\mathsf{dec}(\pi_2(z_s), k_{bs})) & V_n = \pi_2(\mathsf{dec}(\pi_2(z_s), k_{bs})). \end{array}$$

For this protocol the set of outputs and operands of tests are respectively:

$$\begin{array}{l} \mathcal{M}_o(P_Y) = \{\langle a, n_a\rangle, z_a, \pi_2(z_a), \langle b, \mathsf{enc}(\langle \pi_1(z_b), \langle \pi_2(z_b), n_b\rangle\rangle, k_{bs}, r_b)\rangle, z'_b, \\ \qquad\qquad \mathsf{enc}(\langle \pi_1(z_s), \langle x, V_n\rangle\rangle, k_{as}, r_s), \mathsf{enc}(\langle V_a, x\rangle, k_{bs}, r'_s)\} \text{ and} \\ \mathcal{M}_t(P_Y) = \{b, U_b, n_a, U_{n_a}, a, \pi_1(\mathsf{dec}(z'_b, k_{bs}))\}. \end{array}$$

3.2 Our hypotheses

In what follows, we assume $\mathtt{s}$ to be the secret. As in the passive case, destructors above the secret must be forbidden. We also restrict ourself to processes with ground terms in key position. Consider $P_1 = \nu k, r, r'.(\overline{c}\langle \mathsf{enc}(\mathtt{s}, k, r)\rangle \mid c(z).\overline{c}\langle \mathsf{enc}(a, \mathsf{dec}(z, k), r')\rangle)$. The name $\mathtt{s}$ in P_1 is syntactically secret but not strongly secret. Indeed,

$$\begin{array}{ll} P_1 \equiv & \nu k, r, r'.(\nu z.(\{^{\mathsf{enc}(\mathtt{s},k,r)}/_z\} \mid \overline{c}\langle z\rangle \mid c(z).\overline{c}\langle \mathsf{enc}(a, \mathsf{dec}(z, k), r')\rangle)) \\ \to & \nu k, r, r'.(\{^{\mathsf{enc}(\mathtt{s},k,r)}/_z\} \mid \overline{c}\langle \mathsf{enc}(a, \mathtt{s}, r')\rangle) \qquad \text{(COMM rule)} \\ \equiv & \nu k, r, r'.(\nu z'.(\{^{\mathsf{enc}(\mathtt{s},k,r)}/_z, {}^{\mathsf{enc}(a,\mathtt{s},r')}/_{z'}\} \mid \overline{c}\langle z'\rangle)) \\ \xrightarrow{\nu z'.\overline{c}\langle z'\rangle} & \nu k, r, r'.\{^{\mathsf{enc}(\mathtt{s},k,r)}/_z, {}^{\mathsf{enc}(a,\mathtt{s},r')}/_{z'}\} \stackrel{\mathsf{def}}{=} P'_1 \end{array}$$

and P'_1 does not preserve the strong secrecy of $\mathtt{s}$ (see the frame ψ_2 of Section 2.3).

Without loss of generality with respect to cryptographic protocols, we assume that terms occurring in processes are in normal form and that no destructor appears above constructors. Indeed, terms like $\pi_1(\mathsf{enc}(m, k, r))$ are usually not used to specify protocols. We also assume that tests do not contain constructors. Indeed a test $[\langle M_1, M_2\rangle = N]$ can be rewritten as $[M_1 = N_1].[M_2 = N_2]$ if $N = \langle N_1, N_2\rangle$, and $[M_1 = \pi_1(N)].[M_2 = \pi_2(N)]$ if N does not contain constructors, and will never hold otherwise. Similar rewriting applies for encryption, except for the test $[\mathsf{enc}(M_1, M_2, M_3) =$

$N]$ if N does not contain constructors. It can be rewritten in $[\mathsf{dec}(N, M_2) = M_1]$ but this is not equivalent. However since the randomness of encryption is not known to the agent, explicit tests on the randomness should not occur in general.

This leads us to consider the following class of processes. But first, we say that an occurrence q_{enc} of an encryption in a term T is an *agent encryptions* w.r.t. a set of names $\widetilde{n}$ if $t|_{q_{\mathsf{enc}}} = \mathsf{enc}(M, K, R)$ for some M, K, R and $R \in \widetilde{n}$.

Definition 5. *A process P is* well-formed *w.r.t. a name* $\mathtt{s}$ *if it is closed and if:*

1. *any occurrence of $\mathsf{enc}(M, K, R)$ in some term $T \in \mathcal{M}(P)$ is an agent encryption w.r.t. $\mathrm{bn}(P)$, and for any term $T' \in \mathcal{M}(P)$ and position p such that $T'|_p = T$ there is a position q such that $q.3 = p$ and $T'|_q = \mathsf{enc}(M, K, R)$;*
2. *for every term $\mathsf{enc}(M, K, R)$ or $\mathsf{dec}(M, K)$ occurring in P, K is ground;*
3. *any operand of a test $M \in \mathcal{M}_t$ is a name, a constant or has the form $\pi^1(\mathsf{dec}(\ldots \pi^n(\mathsf{dec}(\pi^{n+1}(z), K_l)) \ldots, K_1))$, with $l \geq 0$, where the π^i are words on $\{\pi_1, \pi_2\}$ and z is a variable;*
4. *there are no destructors above constructors, nor above* $\mathtt{s}$.

Conditional tests should not test on $\mathtt{s}$. For example, consider the following process:

$$P_3 = \nu k, r.(\overline{c}\langle \mathsf{enc}(\mathtt{s}, k, r)\rangle \mid c(z).[\mathsf{dec}(z, k) = a].\overline{c}\langle \mathsf{ok}\rangle)$$

where a is a non restricted name. $\mathtt{s}$ in P_3 is syntactically secret but not strongly secret. Indeed, $P_3 \to \nu k, r.(\{^{\mathsf{enc}(\mathtt{s},k,r)}/_z\} \mid [\mathtt{s} = a].\overline{c}\langle \mathsf{ok}\rangle)$. The process $P_3(^{a}/_{\mathtt{s}})$ reduces further while $P_3(^{b}/_{\mathtt{s}})$ does not.

That is why we have to prevent hidden tests on $\mathtt{s}$. Such tests may occur nested in equality tests. For example, let

$$\begin{aligned} P_4 = \nu k, r, r_1, r_2.(&\overline{c}\langle \mathsf{enc}(\mathtt{s}, k, r)\rangle \mid \overline{c}\langle \mathsf{enc}(\mathsf{enc}(a, k', r_2), k, r_1)\rangle \\ &\mid c(z).[\mathsf{dec}(\mathsf{dec}(z, k), k') = a].\overline{c}\langle \mathsf{ok}\rangle) \\ \to P_4' = \nu k, r, r_1, r_2.(&\{^{\mathsf{enc}(\mathtt{s},k,r)}/_z\} \mid \overline{c}\langle \mathsf{enc}(\mathsf{enc}(a, k', r_2), k, r_1)\rangle \mid [\mathsf{dec}(\mathtt{s}, k') = a].\overline{c}\langle \mathsf{ok}\rangle) \end{aligned}$$

Then $P_4(^{\mathsf{enc}(a,k',r')}/_{\mathtt{s}})$ is not equivalent to $P_4(^{n}/_{\mathtt{s}})$, since the process $P_4'(^{\mathsf{enc}(a,k',r')}/_{\mathtt{s}})$ emits the message ok while $P_4'(^{n}/_{\mathtt{s}})$ does not. This relies on the fact that the decryption $\mathsf{dec}(z, k)$ allows access to $\mathtt{s}$ in the test.

For the rest of the section we assume that z_0 is a new fixed variable.

To prevent hidden tests on the secret, we compute an over-approximation of the ciphertexts that may contain the secret, by marking with a symbol $\mathtt{x}$ all positions under which the secret may appear in clear.

We first introduce a function f_{ep} that extracts the least encryption over $\mathtt{s}$ and "clean" the pairing function above $\mathtt{s}$. Formally, we define the partial function

$$f_{ep} \colon \mathcal{T} \times \mathbb{N}_+^* \hookrightarrow \mathcal{T} \times \mathbb{N}_+^*$$

$f_{ep}(U, p) = (V, q)$ where V and q are defined as follows: $q \leq p$ is the position (if it exists) of the lowest encryption on the path p in U. If q does not exist or if p is not a maximal position in U, then $f_{ep}(U, p) = \bot$. Otherwise, V is obtained from $U|_q$ by replacing all arguments of pairs that are not on the path p with new variables. More

precisely, let $V' = U|_q$. The subterm V' must be of the form $\mathsf{enc}(M_1, M_2, M_3)$ and $p = q.i.q'$. Then V is defined by $V = \mathsf{enc}(M'_1, M'_2, M'_3)$ with $M'_j = M_j$ for $j \neq i$ and $M'_i = \mathsf{prune}(M_i, q')$ where prune is recursively defined by: $\mathsf{prune}(\langle N_1, N_2\rangle, 1.r) = \langle \mathsf{prune}(N_1, r), x_r\rangle$, $\mathsf{prune}(\langle N_1, N_2\rangle, 2.r) = \langle x_r, \mathsf{prune}(N_2, r)\rangle$ and $\mathsf{prune}(N, \epsilon) = N$. For example, $f_{ep}(\mathsf{enc}(\mathsf{enc}(\langle\langle a, b\rangle, c\rangle, k_2, r_2), k_1, r_1), 1.1.2) = (\mathsf{enc}(\langle z_\epsilon, c\rangle, k_2, r_2), 1)$.

The function f_e is the composition of the first projection with f_{ep}. With the function f_e, we can extract from the outputs of a protocol P the set of ciphertexts where $\mathtt{s}$ appears in clear below the encryption.

$$\mathcal{E}_0(P) = \{f_e(M[\mathtt{x}]_p, p) \mid M \in \mathcal{M}_o(P) \ \wedge \ M|_p = \mathtt{s}\}.$$

For example, $\mathcal{E}_0(P_Y) = \{\mathsf{enc}(\langle z_1, \langle \mathtt{x}, z_{1.2}\rangle\rangle, k_{as}, r_s), \mathsf{enc}(\langle z_1, \mathtt{x}\rangle, k_{bs}, r'_s)\}$, where P_Y is the process corresponding to the Yahalom protocol defined in previous section.

However $\mathtt{s}$ may appear in other ciphertexts later on during the execution of the protocol after decryptions and encryptions. Thus we also extract from outputs the destructor parts (which may open encryptions). Namely, we define the partial function

$$f_{dp} \colon \mathcal{T} \times \mathbb{N}_+^* \hookrightarrow \mathcal{T} \times \mathbb{N}_+^*$$

$f_{dp}(U, p) = (V, q)$ where V and q are defined as follows: $q \leq p$ is the occurrence of the highest destructor above p (if it exists). Let $r \leq p$ be the occurrence of the lowest decryption above p (if it exists). We have $U|_r = \mathsf{dec}(U_1, U_2)$. Then U_1 is replaced by the variable z_0 that is $V = (U[\mathsf{dec}(\mathrm{z}_0, U_2)]_r)|_q$. If q or r do not exist then $f_{dp}(U, p) = \bot$.

For example, $f_{dp}(\mathsf{enc}(\pi_1(\mathsf{dec}(\pi_2(y), k_1)), k_2, r_2), 1.1.1.1) = (\pi_1(\mathsf{dec}(\mathrm{z}_0, k_1)), 1)$.

The function f_d is the composition of the first projection with f_{dp}. By applying the function f_d to messages of a well-formed process P we always obtain terms D of the form $D = D_1(\ldots D_n)$ where $D_i = \pi^i(\mathsf{dec}(\mathrm{z}_0, K_i))$ with $1 \leq i \leq n$, K_i are ground terms and π^i is a (possibly empty) sequence of projections $\pi_{j_1}(\pi_{j_2}(\ldots(\pi_{j_l})\ldots))$.

With the function f_d, we can extract from the outputs of a protocol P the meaningful destructor part:

$$\mathcal{D}_o(P) = \{f_d(M, p) \mid M \in \mathcal{M}_o(P) \ \wedge \ p \in \mathrm{Pos_v}(M)\}.$$

For example, $\mathcal{D}_o(P_Y) = \{\pi_2(\mathsf{dec}(\mathrm{z}_0, k_{bs})), \pi_1(\mathsf{dec}(\mathrm{z}_0, k_{bs}))\}$.

We are now ready to mark (with $\mathtt{x}$) all the positions where the secret might be transmitted (thus tested). We also define inductively the sets $\mathcal{E}_i(P)$ as follows. For each element E of $\mathcal{E}_i$ we can show that there is an unique term in normal form denoted by $\overline{E}$ such that $\mathcal{V}(\overline{E}) = \{\mathrm{z}_0\}$ and $\overline{E}(E)\!\downarrow = \mathtt{x}$. For example, let $E_1 = \mathsf{enc}(\langle z_1, \langle \mathtt{x}, z_2\rangle\rangle, k_{as}, r_s)$, then $\overline{E_1} = \pi_1(\pi_2(\mathsf{dec}(\mathrm{z}_0, k_{as})))$. We define

$$\begin{aligned}\overline{\mathcal{E}_i}(P) &= \{U \mid \exists E \in \mathcal{E}_i(P), U \leq_{st} \overline{E} \text{ and } \exists q \in \mathrm{Pos}(U), h_{U|_q} = \mathsf{dec}\},\\ \mathcal{E}_{i+1}(P) &= \{M'[\mathtt{x}]_q \mid \exists M \in \mathcal{M}_o(P), p \in \mathrm{Pos_v}(M) \text{ s.t. } f_{ep}(M, p) = (M', p'),\\ &\qquad f_{dp}(M', p'') = (D, q), p = p'.p'', \text{ and } D_1 \in \overline{\mathcal{E}_i}(P)\}.\end{aligned}$$

For example,

$\overline{\mathcal{E}_0}(P_Y) = \{\pi_1(\pi_2(\mathsf{dec}(\mathrm{z}_0, k_{as}))), \pi_2(\mathsf{dec}(\mathrm{z}_0, k_{as})), \mathsf{dec}(\mathrm{z}_0, k_{as}),$
$\qquad \pi_2(\mathsf{dec}(\mathrm{z}_0, k_{bs})), \mathsf{dec}(\mathrm{z}_0, k_{bs})\}$
$\mathcal{E}_1(P_Y) = \{\mathsf{enc}(\langle z_1, \langle z_{1.2}, \mathtt{x}\rangle\rangle, k_{as})\}$
$\overline{\mathcal{E}_1}(P_Y) = \{\pi_2(\pi_2(\mathsf{dec}(\mathrm{z}_0, k_{as}))), \pi_2(\mathsf{dec}(\mathrm{z}_0, k_{as})), \mathsf{dec}(\mathrm{z}_0, k_{as})\}$
and $\mathcal{E}_i(P_Y) = \emptyset$ for $i \geq 2$.

Note that $\mathcal{E}(P) = \cup_{i\geq 0}\mathcal{E}_i(P)$ is finite up-to renaming of the variables since for every $i \geq 1$, every term $M \in \mathcal{E}_i(P)$, $\mathrm{Pos}(M)$ is included in the (finite) set of positions occurring in terms of $\mathcal{M}_0$.

We can now define an over-approximation of the set of tests that may be applied over the secret.

$$\mathcal{M}_t^{\mathtt{s}}(P) = \{M \in \mathcal{M}_t(P) \mid \exists p \in \mathrm{Pos}_{\mathrm{v}}(M) \text{ s.t. } D = D_1(\dots D_n) = f_{dp}(M,p) \neq \bot,$$
$$\text{and } \exists E \in \mathcal{E}(P), \exists i \text{ s.t. } D_i = \pi^i(\mathsf{dec}(z_0, K)), E = \mathsf{enc}(U, K, R) \text{ and } \mathtt{x} \in D_i(E)\downarrow\}$$

For example, $\mathcal{M}_t^{\mathtt{s}}(P_Y) = \{\pi_1(\pi_2(\pi_2(\mathsf{dec}(\pi_1(z_a), k_{as}))))\}$.

Definition 6. *We say that a well-formed process P w.r.t.* $\mathtt{s}$ does not test over $\mathtt{s}$ *if the following conditions are satisfied:*

1. *for all $E \in \mathcal{E}(P)$, for all $D = D_1(\dots D_n) \in \mathcal{D}_o(P)$, if $D_i = \pi^i(\mathsf{dec}(\mathrm{z}_0), K)$ and $e = \mathsf{enc}(U, K, R)$ and $\mathtt{x} \in D_i(E)\downarrow$ then $i = 1$ and $\overline{E} \not\leq_{st} D_1$,*
2. *if $M = N$ or $N = M$ is a test of P and $M \in \mathcal{M}_t^{\mathtt{s}}(P)$ then N is a restricted name.*

Note that $\mathcal{E}(P)$ can be computed in polynomial time from P and that whether P *does not test over* $\mathtt{s}$ is decidable. We show in the next section that the first condition is sufficient to ensure that frames obtained from P are well-formed. It ensures in particular that there are no destructors right above $\mathtt{s}$. If some D_i cancels some encryption in some E and $\mathtt{x} \in D_i(E)\downarrow$ then all its destructors should reduce in the normal form computation (otherwise some destructors (namely projections from D_i) remain above $\mathtt{x}$). Also we have $i = 1$ since otherwise a D_i may have consumed the lowest encryption above $\mathtt{x}$, thus the other decryption may block, and again there would be destructors left above $\mathtt{x}$.

The second condition requires that whenever a operand of a test $M = N$ is potentially dangerous (that is M or $N \in \mathcal{M}_t^{\mathtt{s}}(P)$) then the other operand should be a restricted name.

3.3 Main result

We are now ready to prove that syntactic secrecy is actually equivalent to strong secrecy for protocols that are well-formed and do not test over the secret.

Theorem 2. *Let P be well-formed process w.r.t. a free name $\mathtt{s}$, which is not a channel name, such that P does not test over $\mathtt{s}$. We have $\nu\mathtt{s}.\varphi \not\vdash \mathtt{s}$ for any valid frame φ w.r.t. P if and only if $P(M/\mathtt{s}) \approx_l P(M'/\mathtt{s})$, for all ground terms M, M' public w.r.t. $\mathrm{bn}(P)$.*

Proof. Again, we only provide a sketch of the proof. Showing that strong secrecy implies syntactic secrecy is simple so we concentrate here on the converse implication. Let P be well-formed process w.r.t. a free name $\mathtt{s}$ with no test over $\mathtt{s}$ and assume that P is syntactically secret w.r.t. $\mathtt{s}$.

Let M, M' be to public terms w.r.t. $\mathrm{bn}(P)$. To prove that $P(M/\mathtt{s})$ and $P(M'/\mathtt{s})$ are labeled bisimilar, we need to show that each move of $P(M/\mathtt{s})$ can be matched by $P(M'/\mathtt{s})$ such that the corresponding frames are bisimilar (and conversely). By hypothesis, P is syntactically secret w.r.t. $\mathtt{s}$ thus for any valid frame φ w.r.t. P, we have $\nu\mathtt{s}.\varphi \not\vdash \mathtt{s}$. In order to apply our previous result in the passive setting (Theorem 1), we need to show

that all the valid frames are well-formed. However, frames may now contain destructors in particular if the adversary sends messages that contain destructors. Thus we first need to extend our definition of well-formedness for frames.

Definition 7. *We say that a frame $\varphi = \nu\widetilde{n}.\sigma$ is* extended well-formed *w.r.t.* $\mathtt{s}$ *if for every occurrence $q_{\mathtt{s}}$ of* $\mathtt{s}$ *in $T\downarrow$, where $T = x\sigma$ for some $x \in \mathrm{dom}(\sigma)$, there exists an agent encryption w.r.t. $\widetilde{n}$ above* $\mathtt{s}$*. Let $q_{\mathsf{enc}} < q_{\mathtt{s}}$ the occurrence of the lowest encryption. It must verify that $h_{T|_q} = \langle\rangle$, for all positions q with $q_{\mathsf{enc}} < q < q_{\mathtt{s}}$.*

This definition ensures in particular that there is no destructor directly above $\mathtt{s}$.

Theorem 1 can easily be generalized to extended well-formed frames.

Proposition 1. *Let φ be an extended well-formed frame w.r.t.* $\mathtt{s}$*, where* $\mathtt{s}$ *is a free name in φ. Then $\nu\mathtt{s}.\varphi \not\vdash \mathtt{s}$ iff $\varphi(^{M}/_{\mathtt{s}}) \approx \varphi(^{M'}/_{\mathtt{s}})$ for all M, M' closed public terms w.r.t. φ.*

The first step of the proof of Theorem 2 is to show that any frame produced by the protocol is an extended well-formed frame. We actually prove directly a stronger result, crucial in the proof: the secret $\mathtt{s}$ always occurs under an honest encryption and this subterm is an instance of a term in $\mathcal{E}(P)$.

Lemma 4. *Let P be a well-formed process with no test over* $\mathtt{s}$ *and $\varphi = \nu\widetilde{n}.\sigma$ be a valid frame w.r.t. P such that $\nu\mathtt{s}.\varphi \not\vdash \mathtt{s}$. Consider the corresponding standard frame $\nu\widetilde{n}.\overline{\sigma} = \nu\widetilde{n}.\{^{M_i}/_{y_i} \mid 1 \leq i \leq l\}$. For every i and every occurrence $q_{\mathtt{s}}$ of* $\mathtt{s}$ *in $M_i\downarrow$, we have $f_e(M_i\downarrow, q_{\mathtt{s}}) = E[^{W}/_{\mathtt{x}}]$ for some $E \in \mathcal{E}(P)$ and some term W. In addition $\nu\widetilde{n}.\sigma_i\downarrow$ is an extended well-formed frame w.r.t.* $\mathtt{s}$*.*

The lemma is proved by induction on i and relies deeply on the construction of $\mathcal{E}(P)$.

The second step of the proof consists in showing that any successful test in the process $P(^{M}/_{\mathtt{s}})$ is also successful in P and thus in $P(^{M'}/_{\mathtt{s}})$.

Lemma 5. *Let P be a well-formed process with no test over* $\mathtt{s}$*, $\varphi = \nu\widetilde{n}.\sigma$ a valid frame for P such that $\nu\mathtt{s}.\varphi \not\vdash \mathtt{s}$ and θ a public substitution. If $T_1 = T_2$ is a test in P, then $T_1\theta\sigma(^{M}/_{\mathtt{s}}) =_E T_2\theta\sigma(^{M}/_{\mathtt{s}})$ implies $T_1\theta\sigma =_E T_2\theta\sigma$.*

This lemma is proved by case analysis, depending on whether $T_1, T_2 \in \mathcal{M}_t^{\mathtt{s}}$ and whether $\mathtt{s}$ occurs or not in $\mathrm{fn}(T_1\theta\sigma)$ and $\mathrm{fn}(T_2\theta\sigma)$.

To prove that $P(^{M}/_{\mathtt{s}})$ and $P(^{M'}/_{\mathtt{s}})$ are labeled bisimilar, we introduce the following relation $\mathcal{R}$ between extended processes defined as follows: $A\,\mathcal{R}\,B$ if there is an extended process A_0 and terms M, M' such that $P \Rightarrow^* A_0$, $A = A_0(^{M}/_{\mathtt{s}})$ and $B = A_0(^{M'}/_{\mathtt{s}})$. Then we show that $\mathcal{R}$ satisfies the three points of the definition of labeled bisimilarity using in particular Lemma 5. Hence we have also $\mathcal{R} \subseteq \approx_l$. Since we have clearly that $P(^{M}/_{\mathtt{s}})\,\mathcal{R}\,P(^{M'}/_{\mathtt{s}})$, it follows that $P(^{M}/_{\mathtt{s}}) \approx_l P(^{M'}/_{\mathtt{s}})$.

3.4 Examples

We have seen in Section 3.2 that P_Y is a well-formed process w.r.t. k_{ab} and does not test over k_{ab}. Applying Theorem 2, if P_Y preserves the syntactic secrecy of k_{ab}, we can deduce that the Yahalom protocol preserves the strong secrecy of k_{ab} that is

$$P_Y(^{M}/_{k_{ab}}) \approx_l P_Y(^{M'}/_{k_{ab}})$$

for any public terms M, M' w.r.t. $\mathrm{bn}(P_Y)$. We did not formally prove that the Yahalom protocol preserves the syntactic secrecy of k_{ab} but this was done with several tools in slightly different settings (e.g.[9, 15]).

We have also verified that the Needham-Schroeder symmetric key protocol and the Wide-Mouthed-Frog protocol are both well-formed process w.r.t. k_{ab} and do not test over k_{ab}, where k_{ab} is the exchanged key. Again, the syntactic secrecy of k_{ab} has been proved by several tools (e.g. [9]) in slightly different settings for both protocols. Using Theorem 2, we can deduce that they both preserve the strong secrecy of k_{ab}.

References

1. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *28th Symp. on Principles of Programming Languages (POPL'01)*, pages 104–115. ACM, 2001.
2. M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *4th Conf. on Computer and Communications Security (CCS'97)*, pages 36–47. ACM, 1997.
3. R. Amadio and D. Lugiez. On the reachability problem in cryptographic protocols. In *12th Conf. on Concurrency Theory (CONCUR'00)*, volume 1877 of *LNCS*, pages 380–394, 2000.
4. The AVISPA Project. http://www.avispa-project.org/.
5. B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Computer Security Foundations Workshop (CSFW'01)*, pages 82–96. IEEE Comp. Soc. Press, 2001.
6. B. Blanchet. Automatic Proof of Strong Secrecy for Security Protocols. In *IEEE Symposium on Security and Privacy (SP'04)*, pages 86–100, 2004.
7. B. Blanchet and A. Podelski. Verification of cryptographic protocols: Tagging enforces termination. In *Foundations of Software Science and Computation Structures (FoSSaCS'03)*, volume 2620 of *LNCS*, April 2003.
8. J. Borgström, S. Briais, and U. Nestmann. Symbolic bisimulations in the spi calculus. In *Int. Conf. on Concurrency Theory (CONCUR'04)*, volume 3170 of *LNCS*. Springer, 2004.
9. L. Bozga, Y. Lakhnech, and M. Périn. HERMES: An automatic tool for verification of secrecy in security protocols. In *15th Conference on Computer Aided Verification (CAV'03)*, volume 2725 of *LNCS*, pages 219–222. Springer, 2003.
10. H. Comon-Lundh and V. Cortier. New decidability results for fragments of first-order logic and application to cryptographic protocols. In *Rewriting Techniques and Applications (RTA'2003)*, LNCS 2706, pages 148–164. Springer-Verlag, 2003.
11. V. Cortier and B. Warinschi. Computationally Sound, Automated Proofs for Security Protocols. In *European Symposium on Programming (ESOP'05)*, volume 3444 of *LNCS*, pages 157–171. Springer, April 2005.
12. G. Denker, J. Millen, and H. Rueß. The CAPSL Integrated Protocol Environment. Technical Report SRI-CSL-2000-02, SRI International, Menlo Park, CA, 2000.
13. N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Workshop on Formal Methods and Security Protocols*, 1999.
14. H. Hüttel. Deciding framed bisimilarity. In *INFINITY'02*, August 2002.
15. L. C. Paulson. Relations between secrets: Two formal analyses of the Yahalom protocol. *Journal of Computer Security*, 9(3):197–216, 2001.
16. R. Ramanujam and S.P.Suresh. Tagging makes secrecy decidable for unbounded nonces as well. In *23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'03)*, Mumbai, 2003.
17. M. Rusinowitch and M. Turuani. Protocol Insecurity with Finite Number of Sessions and Composed Keys is NP-complete. *Theoretical Computer Science*, 299:451–475, 2003.
18. Eugen Zalinescu, Véronique Cortier, and Michaël Rusinowitch. Relating two standard notions of secrecy. Research Report 5908, INRIA, Avril 2006.