On the existence of an effective and complete inference system for cryptographic protocols

Liana Bozga, Cristian Ene and Yassine Lakhnech

Liana.Bozga@imag.fr , Cristian.Ene@imag.fr , Yassine.Lakhnech@imag.fr ,
VERIMAG , 2 av. de Vignate, 38610 Grenoble, France

Abstract. A central question in the domain of program semantics and program verification is the existence of a complete inference system for assertions of the form $\pi \models \varphi$ meaning that program π satisfies property φ. A stronger version of this question asks for an effective (decidable) complete inference system. We investigate these questions for cryptographic protocols focusing on authentication and confidentiality properties. While it is not difficult to see that a complete and effective inference system cannot exist when an unbounded number of sessions are considered, we prove that such a system exists for bounded protocols. More, precisely 1.) we provide a complete weakest pre-condition calculus for bounded cryptographic protocols and 2.) we show that assertions needed for completeness of the calculus are expressible in a decidable second order logic on terms.

1 Introduction

A central question in the domain of program semantics and program verification is the existence of a complete (and sound) inference system for assertions of the form $\pi \models \varphi$ meaning that program π satisfies property φ. A stronger version of this question asks for an effective (decidable) complete inference system. This is the question of the relationship between the truth of formulae of the form $\pi \models \varphi$ and their provability. For While-programs (or counter machines), for instance, it has been proved that it is possible to design an inference system such that provability implies truth (i.e., soundness) but impossible to have a sound system that is also complete and effective, i.e., it is impossible to have a decidable inference system such that truth implies provability (see [9] for a complete survey). Roughly speaking, the reason is that one can describe transitive closures using while programs while this is not possible in general in 1st-order logics except when Peano arithmetic is included. In other words, one has to sacrifice effectiveness (e.g., by including Peano arithmetic in the logic), or completeness and accept that some valid formulae $\pi \models \varphi$ cannot be proved or even expressed. This situation of While-programs led to the what is called Cook's relative completeness: *is it possible to have a complete inference system for programs, if we assume all facts of the underlying logic as axioms, i.e., all facts about the considered data are given?* The main question we address in this paper is the following: *what is the situation for cryptographic protocols?*

Beyond the theoretic relevance of this question, it has several practical consequences. Indeed, if one can provide a complete inference system for cryptographic protocols this can serve as a basis to develop compositional proof theories as well as refinement theories. The latter would be of great interest as the problem of composing cryptographic protocols (CP for short), i.e., which properties are preserved when CPs are composed, as well as the relationship between the abstract specification of a CP and its real implementation remain two insufficiently investigated subjects (cf. [15]). Moreover, a decidable complete inference system provides a symbolic decision procedure.

In this paper, we introduce a complete and effective inference system for bounded cryptographic protocols. Let us explain what we mean. A session of a cryptographic protocol can be specified as a sequence of sending and receiving messages. One can consider either fixed bounded number of sessions or an unbounded one. In the first case, we speak about bounded protocols but in both cases the size of the messages is unbounded. It is not difficult to encode a counter machine as an unbounded CP. Hence, we know that it is not possible to have an effective complete (and sound) inference system. We show that such a system exists for bounded protocols. This provides an alternative proof of the decidability of secrecy for bounded CPs and covers more properties than in existing work. We introduce a logic, called SPL for Security Properties Logic, for describing security properties and develop a calculus for computing the weakest condition that has to be satisfied by the initial configurations of the protocol in order to guarantee that a property described by an SPL formula is satisfied. We prove soundness and completeness for the introduced calculus. Then, we study the decidability of SPL and show that although the satisfiability (existence of a model) of SPL formulae is, in general, undecidable, it is decidable for its existential fragment, i.e., the satisfiability of formulae of the form $\exists X.\varphi$, where φ is quantifier-free can be decided effectively (Section 6). Now, it turns out that interesting security properties are expressible in the universal fragment of the logic (see Section 3.4) and that the weakest precondition of a universal formula is expressible as a universal formula (Section 5). Hence, given a protocol π and a property φ, using the calculus one can compute a formula $wp(\pi, \varphi)$ such that there is an attack starting for an initial state satisfying ψ iff $\neg wp(\pi, \varphi) \wedge \psi$ is satisfiable. Thus, if ψ is given in the existential fragment, which is the interesting situation, one can effectively check whether $\neg wp(\pi, \varphi) \wedge \psi$ is satisfiable.

Related work The results of this paper provide an algorithm for checking security properties (confidentiality and authentication) of cryptographic protocols. It has several interesting aspects:

1. it covers other properties than confidentiality (secrecy); indeed while other methods rely on an ad hoc reduction of authentication properties to secrecy, our method is directly applicable.
2. as initial configuration are described by formulae of the introduced logic, it can deal with infinite non-regular sets of messages initially known by the intruder.

3. we believe that our method is more easily amenable to extended intruder models. We demonstrate this by considering cipher block chaining.

While several methods have been designed for the verification of a fixed number of sessions [18, 1, 3, 16, 12, 7, 8] to our knowledge it has not been previously proven that a decidable and complete inference system for cryptographic protocols exists.

2 Preliminaries

Let X be a countable set of variables and let F^i be a countable set of function symbols of arity i, for every $i \in \mathbb{N}$. Let $F = \bigcup_{i\in\mathbb{N}} F^i$. The set of *terms over X and F*, denoted by $\mathcal{T}(X, F)$. We denote by $\leq$ the *subterm* relation $\leq$ on $\mathcal{T}(X, F)$. As usual, function symbols of arity 0 are called constant symbols. *Ground terms* are terms with no variables. We denote by $\mathcal{T}(F)$ the set of ground terms over F. For any $t_1, t_2 \in \mathcal{T}(X, F)$, we denote with $\mu(t_1, t_2)$ the most general unifier (shortly mgu) of t_1 and t_2, if it exists. More precisely, by $\mu(t_1, t_2)$ we denote the representation of the mgu of t_1 and t_2 as a conjunction of equalities of the form $x = t$, if it exists. If it does not exist then $\mu(t_1, t_2)$ should be the constant $false$ (falsum). We write $t_1 \sim t_2$, if t_1, t_2 can be unified. Also, for any substitution $\sigma : X \to \mathcal{T}(X, F)$, we denote by $t\sigma$ the application to t of the homomorphic extension of σ to terms. Given a set $\tilde{x}$ of variables, we denote by $\Gamma(\tilde{x})$ the set consisting of ground substitutions with domain $\tilde{x}$. We also write $\Gamma(x)$ instead of $\Gamma(\{x\})$.

A tree tr is a function from a finite subset of ω^* to $X \cup F$ such that $tr(u) \in F^n$ iff $u \cdot j \in dom(tr)$, for every $j \in \{0, \cdots, n-1\}$. Henceforth, we tacitly identify the term t with $Tr(t)$. The elements of $dom(t)$ are called *positions* in t. We use $\prec$ to denote the prefix relation on ω^*. We write $t(p)$ to denote the symbol at position p in t and $t_{|p}$ to denote the subterm of t at position p, which corresponds to the tree $t_{|p}(x) = t(p \cdot x)$ with $x \in dom(t_{|p})$ iff $p \cdot x \in dom(t)$. Given a term t and positions p and q, we say that $t_{|p}$ dominates $t_{|q}$ if $p \prec q$.

If $w_1, w_2 \in \Sigma^*$ are words over an alphabet Σ, then we denote by $w_2^{-1}w_1$ the word obtained from w_1 after removing the prefix w_2.

3 The Protocol and Intruder Model

We describe in this section the model of cryptographic protocols used in this work. We mention that this model is by now a standard one used, for instance, in [4]. We begin by describing the messages involved in a protocol model.

3.1 Messages

The set of messages is denoted by $\mathcal{M}$ and contains ground terms constructed from constant symbols and the function symbols **encr** : $\mathcal{M} \times \mathcal{K} \to \mathcal{M}$ and **pair** : $\mathcal{M} \times \mathcal{M} \to \mathcal{M}$, where $\mathcal{K}$ is a set of keys. Constant symbols are also called atomic messages and are defined as follows:

1. *Principal names* are used to refer to the principals in a protocol. The set of all principals is $\mathcal{P}$.
2. *Nonces* can be thought as randomly generated numbers. As no one can predict their values, they are used to convince for the freshness of a message. We denote by $\mathcal{N}$ the set of nonces.
3. *Keys* are used to encrypt messages. If k is a key then we use k^{-1} to denote its inverse. Moreover, we use $pbk(A)$ to denote the public key of A.

For the sake of simpicity we leave out signature and hash functions but we can easily handle them in our model.

Let $\mathcal{A} = \mathcal{P} \cup \mathcal{N} \cup \mathcal{K}$ and $\mathcal{F} = \mathcal{A} \cup \{\mathbf{encr}, \mathbf{pair}\}$. As usual, we write (m_1, m_2) for $\mathbf{pair}(m_1, m_2)$ and $\{m\}_k$ instead of $\mathbf{encr}(m, k)$. *Message terms* are the elements of $\mathcal{T}(\mathcal{X}, \mathcal{F})$, that is, terms over the atoms $\mathcal{A}$, a set of variables $\mathcal{X}$ and the binary function symbols **encr** and **pair**. *Messages* are ground terms in $\mathcal{T}(\mathcal{X}, \mathcal{F})$, i.e, $\mathcal{M} = \mathcal{T}(\mathcal{F})$. For conciseness, we write $\mathcal{T}$ instead of $\mathcal{T}(\mathcal{X}, \mathcal{F})$.

3.2 The intruder model

We describe how an intruder can create new messages from already known messages. We use the most commonly used model, introduced by Dolev and Yao [10], which is given by a formal system $\vdash$.

The intruder capabilities for intercepting messages and sending (fake) messages are fixed by the operational semantics. Thus, the *derivability of a message* m from a set E of messages, denoted by $E \vdash m$, is described by the following axiom and rules:

- If $m \in E$ then $E \vdash m$.
- If $E \vdash m_1$ and $E \vdash m_2$ then $E \vdash \mathbf{pair}(m_1, m_2)$. This rule is called pairing.
- If $E \vdash m$ and $E \vdash k \in \mathcal{K}$ then $E \vdash \mathbf{encr}(m, k)$. This is called encryption.
- If $E \vdash \mathbf{pair}(m_1, m_2)$ then $E \vdash m_1$ and $E \vdash m_2$. This is called projection.
- If $E \vdash \mathbf{encr}(m, k)$, $E \vdash k'$ and k and k' are inverses then $E \vdash m$. This is called decryption.

The last two rules are called *decomposition rules*, the others are called *composition rules*. A derivation of a message that does not use the last two rules is denoted by $E \vdash_c m$. We write $E \vdash M$, if $E \vdash m$ holds for every $m \in M$.

For a term t, we use the notation $E \not\vdash t$ to denote that no instance of t is derivable from E, that is, for no substitution $\sigma : \mathcal{X} \to \mathcal{M}$, we have $E \vdash t\sigma$.

We now define *critical* and *non-critical* positions in a message. The idea is that since there is no way to deduce from an encrypted message the key with which it has been encrypted, the key position in messages of the form $\mathbf{encr}(m, k)$ is not critical[1]. Formally, given a term t, a position p in t is called *non-critical*, if there is a position q such that $p = q \cdot 2$ and $t(q) = \mathbf{encr}$; otherwise it is called *critical*. We will also use the notation $s \in_c m$ to denote that s appears in m at a critical position, i.e., there exists $p \in dom(m)$ such that p is critical and $m_{|p} = s$.

[1] For the insider, the critical position corresponds, for instance, to the subterm relation in the strand space model [11, 20].

3.3 Process model

Definition 1 (actions and protocols). *Actions are defined by:*

$$\alpha ::= l \xrightarrow{!t} l' \mid l \xrightarrow{?t(\tilde{x})} l' \mid l \xrightarrow{x:=t} l' \mid l \xrightarrow{x=t} l'$$

where $t \in \mathcal{T}$ is a term, l, l' are labels and $\tilde{x} \subseteq var$ is a set of variables. An action is an output, an input, an assignment or just an equality test. In the case of an input, $\tilde{x}$ denotes the variables that are instantiated by the action. The set of actions is denoted by $\mathcal{A}ct$.

A protocol is represented by a set of sequences of actions. More precisely, a protocol Π is given by:

$$\Pi = \sum_{i=1}^{n} \alpha_1^i \cdots \alpha_{n_i}^i.$$

where $\alpha_j^i = \ell_j^i \xrightarrow{\beta_j^i} \ell_{j+1}^i$ for some β_j^i with $j \in \{1, \ldots, n_i\}$. Here, the labels ℓ represent control points and $\sum$ is the usual non-deterministic choice. This corresponds to the representation of a fixed set sessions put in parallel by their possible interleavings. Usually, we use the more intuitive notation:

$$\Pi = \sum_{i=1}^{n} \ell_0^i \beta_0^i \cdots \ell_{n_i}^i \beta_{n_i}^i \ell_{n_i+1}^i.$$

Definition 2 (operational semantics). *A configuration of a protocol run is given by triple (σ, E, ℓ_j^i) consisting of a substitution σ, a set of messages E and a control point ℓ_j^i. The operational semantics is defined as a labelled transitional system over the set of configurations $\mathcal{C}onf$. The transition relation*

$$(\sigma, E, \ell_j^i) \xrightarrow{\alpha} (\sigma', E', \ell_{j+1}^i)$$

is defined as follows:

- $(\sigma, E, \ell_j^i) \xrightarrow{\alpha} (\sigma, E \cup \{t\sigma\}, \ell_{j+1}^i)$, *if $j \leq n_i$ and $\alpha = \ell_j^i \xrightarrow{!t} \ell_{j+1}^i$. That is, sending the message $t\sigma$ amounts to adding $t\sigma$ to the knowledge of the intruder.*
- *for $\rho \in \Gamma(x)$ with $E\sigma \vdash t(\sigma \oplus \rho)$, we have $(\sigma, E, \ell_j^i) \xrightarrow{\alpha} (\sigma \oplus \rho, E, \ell_{j+1}^i)$, if $j \leq n_i$ and $\alpha = \ell_j^i \xrightarrow{?t(\tilde{x})} \ell_{j+1}^i$. That is, $?t$ corresponds to receiving any message that matches with $?t\sigma$ and is known by the intruder.*
- $(\sigma, E, \ell_j^i) \xrightarrow{\alpha} (\sigma \oplus [x \mapsto t\sigma], E, \ell_{j+1}^i)$, *if $j \leq n_i$ and α is the assignment $\ell_j^i \xrightarrow{x:=t} \ell_{j+1}^i$. The effect of an assignment is as usual.*
- $(\sigma, E, \ell_j^i) \xrightarrow{\alpha} (\sigma, E, \ell_{j+1}^i)$, *if $\sigma(x) = t\sigma$, $j \leq n_i$, and α is the test $\ell_j^i \xrightarrow{x=t} \ell_{j+1}^i$. The action $x = t$ behaves as a filter.*

The initial configuration is given by a substitution σ_0, a set of terms E_0 such that the variables in E_0 do not appear in the protocol description and a control point $\ell_0 \in \{\ell_0^1, \cdots, \ell_{n_i}^n\}$.

To illustrate the definitions we consider the Needham-Schroeder public-key protocol, NS for short. The protocol is designed to ensure principal authentication: at the end of the protocol, the two participants A and B should be convinced about the identity of their respective correspondent.

A session S between participants A and B of NS protocol is:

$$\begin{array}{ll} A \to B: & \{A, N_a\}_{pbk(B)} \\ B \to A: & \{N_a, N_b\}_{pbk(A)} \\ A \to B: & \{N_b\}_{pbk(B)} \end{array}$$

The keys $pbk(A)$ and $pbk(B)$ are the public keys of the participant A respectively of the participant B and the nonce N_a and N_b are fresh values generated by A respectively B.

In fact each participant of the protocol may be seen as a sequential process. The participant A sends his identity A and a fresh nonce N_a encrypted with the public key of B. Then, if A receives back a message encrypted with his public key that contains the nonce N_a, and an other value x he knows that the message was created by B. In fact, only B could obtain the nonce N_a by decrypting the first message. Finally, to complete the protocol, A sends back the value x encrypted with the public key of B.

On the other side, when B receives a participant name y and a value z encrypted with his public key, he sends the value z and a fresh nonce N_b encrypted with the public key of the participant y. Then, if B receives back a message encrypted with his public key that contains the nonce N_b, he knows that the message was created by y. In fact, only y could obtain the nonce N_b by decrypting the message sent by B.

The next table shows how we represent each participant. The labels represent the local control points of the process.

A:	B:
$0 : !\{A, N_a\}_{pbk(p)}$	$0 : ?\{y, z\}_{pbk(B)}$
$1 : ?\{N_a, x\}_{pbk(A)}$	$1 : !\{z, N_b\}_{pbk(y)}$
$2 : !\{x\}_{pbk(p)}$	$2 : ?\{N_b\}_{pbk(B)}$
$3 :$	$3 :$

We write A^S to specify the process A of the session S. We write $v^{(S)}$ to specify a variable, a nonce or a participant v involved in a session S.

Let consider two parallel sessions $S1(a, b)||S2(a, c)$. An execution trace has control points of the form $(pc_A^{(S1)}, pc_B^{(S1)}, pc_A^{(S2)}, pc_B^{(S2)})$ which correspond to the local control points of A^{S1} respectively B^{S1}, A^{S2}, B^{S2}. Traces are obtained by interleaving of actions in the two sessions. For example, one possible trace, where the session $S2$ starts before the session $S1$ ends, is:

$$\begin{array}{ll}
(0,0,0,0)\ !\{a, {N_a}^{(S1)}\}_{pbk(b)} & (1,0,0,0)\ !\{a, {N_a}^{(S2)}\}_{pbk(c)} \\
(1,0,1,0)\ ?\{y^{(S1)}, z^{(S1)}\}_{pbk(b)} & (1,1,1,0)\ ?\{y^{(S2)}, z^{(S2)}\}_{pbk(c)} \\
(1,1,1,1)\ !\{z^{(S1)}, {N_b}^{(S1)}\}_{pbk(y^{(S1)})} & (1,2,1,1)\ ?\{{N_a}^{(S1)}, x^{(S1)}\}_{pbk(a)} \\
(2,2,1,1)\ !\{x^{(S1)}\}_{pbk(b)} & (3,2,1,1)\ ?\{{N_b}^{(S1)}\}_{pbk(b)} \\
(3,3,1,1)\ !\{z^{(S2)}, {N_b}^{(S2)}\}_{pbk(y^{(S2)})} & (3,3,1,2)\ ?\{{N_a}^{(S2)}, x^{(S2)}\}_{pbk(a)} \\
(3,3,2,2)\ !\{x^{(S2)}\}_{pbk(c)} & (3,3,3,2)\ ?\{{N_b}^{(S2)}\}_{pbk(c)} \\
(3,3,3,3) &
\end{array}$$

In this example the initial substitution is $[A^{(S1)} = a; p^{(S1)} = b; B^{(S1)} = b; A^{(S2)} = a; p^{(S2)} = c; B^{(S2)} = c]$.

3.4 Expressing security properties

In this subsection, we introduce an intuitive logic, which allows us to express security properties about cryptographic protocols. The prurpose is to recall these properties and show how they can be described. The set of formulas $\mathcal{F}_0$, is defined in Table 1, x is a meta-variable that ranges over the set $\mathcal{V}$ of first-order variables. First-order variables range over messages; t is a meta-variable over terms.

$$\mathcal{F}_0 \ni \varphi, \psi ::= \mathit{Secret}(t) \mid x = t \mid pc = \ell \mid \top \mid \bot \mid \varphi \wedge \psi \mid \neg\varphi \mid \forall x \varphi$$

Table 1. The set of formulas $\mathcal{F}_0$

The proposition $\mathit{Secret}(t)$ expresses secrecy in the following (usual) sense: is true in a configuration (σ, E, ℓ), if $t\sigma$ cannot be derived by the intruder from $E\sigma$. The proposition $pc = \ell$ is true, if the program counter equals ℓ.

Definition 3 (semantics). *The interpretation ofa formula is given by the set of its models, i.e., configurations $\mathcal{C}onf$ that satisfy the formula:*

- $[\![\mathit{Secret}(t)]\!] = \{(\sigma, E, \ell) \mid E\sigma \not\vdash t\sigma\}$
- $[\![x = t]\!] = \{(\sigma, E, \ell) \mid \sigma(x) = \sigma(t)\}$.
- $[\![pc = \ell]\!] = \{(\sigma, E, \ell) \mid (\sigma, E, \ell) \text{ is a configuration}\}$
- $[\![\top]\!] = \mathcal{C}onf$
- $[\![\bot]\!] = \emptyset$
- $[\![\varphi \wedge \psi]\!] = [\![\varphi]\!] \cap [\![\psi]\!]$
- $[\![\varphi \vee \psi]\!] = [\![\varphi]\!] \cup [\![\psi]\!]$
- $[\![\neg\varphi]\!] = \mathcal{C}onf \setminus [\![\varphi]\!]$
- $[\![\forall x \varphi]\!] = \bigcap_{\{\rho \mid \rho \in \Gamma(x)\}} [\![\varphi[\rho(x)/x]]\!]$.

There are many definitions of authentication that we can find in the literature [5, 22, 14, 19, 17]. We show here, by means of an example, how the introduced logic allows to specify the authentication properties discussed in [14]. Obviously, we can express semantic secrecy in our logic using the predicate *Secret*. Consider the NS protocol, introduced in subsection 3.3.

Aliveness of the initiator is guaranteed to the participant b in session $S1$: if b completes a run of the protocol in session $S1$, as responder, with one participant, let say x, then $y^{(S1)} = x$ and the participant x has previously been running the protocol (not necessarily with b neither not necessarily as initiator).

$$pc_B^{(S1)} = 3 \Rightarrow ((y^{(S1)} = a \wedge (pc_A^{(S1)} \neq 0 \vee pc_A^{(S2)} \neq 0)) \vee (y^{(S1)} = c \wedge pc_B^{(S2)} \neq 0) \vee y^{(S1)} = b)$$

Weak agreement of the initiator is guaranteed to the responder b in session $S1$: if b completes a run of the protocol in session $S1$, as responder, with one participant, let say x, then $y^{(S1)} = x$ and the participant x has previously been running the protocol **with** b (not necessarily as initiator).

$$\begin{aligned} pc_B^{(S1)} = 3 \Rightarrow ((&y^{(S1)} = a \wedge pc_A^{(S1)} \neq 0 \wedge p^{(S1)} = b) \vee (y^{(S1)} = a \wedge pc_A^{(S2)} \neq 0 \wedge p^{(S2)} = b) \vee \\ &(y^{(S1)} = c \wedge pc_B^{(S2)} \neq 0 \wedge y^{(S2)} = b) \vee y^{(S1)} = b) \end{aligned}$$

The definition of non-injective agreement and agreemenr are discussed in Appendix A.

It should be clear that given an authentication property and a bounded CP, one can systematically derive a formula expressing the property. Also, interesting to notice that the formulae that express these properties do not use the predicate $Secret(t)$. But then, what is about the general belief that verifying authentication can be reduced to verifying secrecy properties? The weakest precondition calculus we develop in Section 5 clearly (and rigorously) shows where secrecy intervenes.

4 The SPL logic

In this section, we present a more expressive logic, the SPL logic, that embeds the logic introduced in the previous section. SPL is used in Section 5 as the underlying logic for the weakest precondition calculus.

Henceforth, let $K \subseteq \mathcal{K}$ be a fixed but arbitrary set of keys, such that $\emptyset \neq K \neq \mathcal{K}$.

4.1 A syntactic characterization of secrecy

A major problem we face for developing a complete inference system for cryptographic protocols is that secrecy. i.e., $E \not\vdash m$, is not expressive enough. For instance, consider the protocol $?\{x\}_k; !x$ and the property $E \not\vdash (s_1, s_2)$. What should be the weakest precondition that ensures this property at the end of this protocol? In this section, we introduce a modality that allows to express weakest preconditions and provides a syntactic characterization of secrecy.

Intuitively, this modality is a predicate that asserts that given the intruder's knowledge E, a term s is protected by a key in K in any message the intruder can derive from E.

A pair $(\{t\}_k, r)$, where t is a term, $k \in K$ and r a critical position in $\{t\}_k$ is called a *term transducer (TT for short).* Intuitively, the pair $(\{t\}_k, r)$ can be seen as function that takes as argument a term that matches with $\{t\}_k$ and returns as result the term $\{t\}_{k_{|r}}$. Notice that the decomposition rules in the intruder model can be considered as a set of term transducers the intruder can apply to get new terms. As w ill become clear later, a run of a CP provides the intruder with new term transducer she (he) can apply to learn new terms.

We are now ready to introduce the main modality of the logic:

Definition 4. *Let m and s be two messages and let $w \in (\mathcal{M} \times \mathcal{P}os)^*$ be a sequence of term transducers.*

We define the predicate $m\langle w\rangle s$, which we read "s is w-protected in m", recursively on the structure of m and length of w:

- *m is atomic and $m \neq s$, or*
- *$m = \mathbf{pair}(m_1, m_2)$, $m \neq s$ and both $m_1\langle w\rangle s$ and $m_2\langle w\rangle s$ are true, or*
- *$m = \mathbf{encr}(m_1, k)$, $m \neq s$, $k \notin K$ and $m_1\langle w\rangle s$ is true, or*
- *$m = \mathbf{encr}(m_1, k)$, $m \neq s$, $k \in K$ and $w = \epsilon$, or*
- *$m = \mathbf{encr}(m_1, k)$, $w = (b, r).w_1$, $m \neq s$, $k \in K$, and $m \neq b$ or $m|_r\langle w_1\rangle s$ is true.*

This definition is easily generalized to sets of messages: Let M and S be sets of messages, w a sequence of term transducers and K a set of keys. We say that the secrets S are w - protected in M denoted by $M\langle w\rangle S$, if it holds $\bigwedge_{m\in M, s\in S} m\langle w\rangle s$.

Example 1. Let $m = (\{A, \{N\}_{k_1}\}_{k_2}, A)$ and $K = \{k_1, k_2\}$. Then, $m\langle\epsilon\rangle N$ is true since $\{A, \{N\}_{k_1}\}_{k_2}\langle\epsilon\rangle N$ and $A\langle\epsilon\rangle N$ are true.

Let now $w = (\{A, \{N\}_{k_1}\}_{k_2}, 12).(\{N\}_{k_1}, 1)$. Then, $m\langle w\rangle N$ is false since applying the term transducer $(\{A, \{N\}_{k_1}\}_{k_2}, 12)$ yields $\{N\}_{k_1}$ on which an application of $(\{N\}_{k_1}, 1)$ yields N.

Closure of sets of secrets In this section, we define when a set of messages is closed. Closed sets of secrets enjoy the property that they are not derivable by composition. Intuitively, a set of messages is closed if it contains all messages along every path of the tree representing a message in the set.

Let M be a set of sets of messages and let m be a message. We use the notation: $m \odot M = \{M_i \cup \{m\} | M_i \in M\}$.

We define when a set of messages is closed. The closure of a set S ensures that the intruder cannot derive a message in S by composition rules.

Definition 5 (closure).

$$wc(m) = m \odot \begin{cases} wc(m1) \cup wc(m2) & \textit{if } m = (m1, m2) \\ wc(m') \cup wc(k) & \textit{if } m = \{m'\}_k \\ \{K^{-1}\} & \textit{if } m \textit{ is atomic} \end{cases}$$

where $K^{-1} = \{k^{-1} \mid k \in K\}$. A set M of messages is called closed, if for any $m \in M$ there exists $M' \in wc(m)$ such that $M' \subseteq M$.

Example 2. Consider the message $m = (\{A, N\}_k, B)$. Then wc(m) consists of the following sets:

$K^{-1} \cup \{(\{A, N\}_k, B), \{A, N\}_k, (A, N), A\}$ $\quad$ $K^{-1} \cup \{(\{A, N\}_k, B), \{A, N\}_k, k\}$
$K^{-1} \cup \{(\{A, N\}_k, B), \{A, N\}_k, (A, N), N\}$ $\quad$ $K^{-1} \cup \{(\{A, N\}_k, B), B\}$.

We can prove the following:

Lemma 1. *Let S be a closed set of messages. And let E be a set of messages such that $S \cap E = \emptyset$. Then, $E \not\vdash_c S$. In other words, if S is closed then no message in S can be derived uniquely by the composition rules.*

We use the notation $E\langle w_i, S_i\rangle_I$ for $\bigwedge_{i\in I} E\langle w_i\rangle S_i$. Our purpose now is to define conditions on w_i and S_i such that for any set E of messages, if $E\langle w_i, S_i\rangle_I$ then $m\langle w_i, S_i\rangle_I$, for any message m derivable from E. In other words, such conditions ensure that $E\langle w_i, S_i\rangle_I$ is stable under the derivations rules defining the intruder. Remember that closure guarantees stability only under composition rules.

Example 3. Let $E = \{s_1, s_2\}$ be a set of messages. Then we have $E\langle w\rangle(s_1, s_2)$. But we have both $E \vdash (s_1, s_2)$ and $\neg(s_1, s_2)\langle w\rangle(s_1, s_2)$.

This example shows that we need to consider only closed sets of secrets. But this is not sufficient, as showed by the following example.

Example 4. Let $E = \{\{\{s\}_{k_1}, k_2\}$ be a set of messages. We have $E\langle(\{\{s\}_{k_1}\}_{k_2}, 11)\rangle s$. But we have both $E \vdash \{\{s\}_{k_1}\}_{k_2}$ and $\neg\{\{s\}_{k_1}\}_{k_2}\langle(\{\{s\}_{k_1}\}_{k_2}, 11)\rangle s$.

Hence, we need to deal also with the interior term transducers. To do so, let (b, p) be a term transducer. Then, we denote by $\mathrm{lpt}(b, p)$ the next term transducer in b from above that dominates $b_{|p}$, if it exists. A formal definition is given in Appendix B.2 but let us give an example.

Example 5. Let b be the term $\{(\{N\}_{k'}, A)\}_k$ with $k, k' \in K$. Then, $\mathrm{lpt}(b, 111) = (\{N\}_{k'}, 1)$. But $\mathrm{lpt}(b, 12)$ does not exist neither $\mathrm{lpt}(b, 11)$ does.

We have now everything we need to express the conditions that guarantee stability under the intruder's derivations:

Definition 6. $(w_i, S_i)_{i \in I}$ *is called* well-formed*, if the following conditions are satisfied for every $i \in I$:*

– *S_i is closed,*
– *if $w_i = (b, r).w$ and if there exists a term transducer $(b_1, r_1) = lpt(b, r)$, then there exists $j \in I$ such that one of the following is true:*
  • *$b \in S_j$*
  • *$w_j = (b_1, r_1).w$ and $S_i \subseteq S_j$.*

The main property of $E\langle w_i, S_i\rangle_I$ is that it is stable under the intruder's deduction rules. Indeed, we have:

Proposition 1. *Let E be a set of messages such that $E\langle w_i, S_i\rangle_I$ and let $(w_i, S_i)_{i\in I}$ be well-formed. Moreover, let m be a message with $E \vdash m$. Then, $m\langle w_i, S_i\rangle_I$.*

Proof. See Appendix B.3.

The modality $E\langle w\rangle S$ has another interesting property with respect to intruder's derivations:

Proposition 2. *Let m be a message and E a set of messages such that $\mathcal{K} \setminus K^{-1} \subseteq E$. Then, $E \not\vdash m$ iff there exists a set of messages $A \in wc(m)$ s.t. $E\langle\epsilon\rangle A$.*

4.2 SPL: A Logic for Security Properties

The syntax of SPL is defined in Table 2, where X is a fixed second-order variable that ranges over sets of messages and x is a meta-variable that ranges over the set $\mathcal{V}$ of first-order variables. First-order variables range over messages; t is a meta-variable over terms. Moreover, S is a finite set of terms and w is a finite sequence of term transducers that can contain free variables. The formulae are interpreted over a restricted set of configurations $\mathcal{C}onf_0 = \{(\sigma, E, l) | (\sigma, E, l) \in \mathcal{C}onf, \mathcal{K} \setminus K^{-1} \subseteq E\}$.

Definition 7 (semantics). *The semantics of* SPL *is defined as in Definition 3 except that we also have the following clauses:*

– $[\![X\langle w\rangle S]\!] = \{(\sigma, E, \ell) \mid E\sigma\langle w\sigma\rangle S\sigma\}$

$$\varphi, \psi ::= X\langle w\rangle S \mid x\langle w\rangle S \mid x = t \mid pc = \ell \mid \top \mid \bot \mid \varphi \wedge \psi \mid \neg\varphi \mid \forall x \varphi$$

Table 2. The set of formulae SPL

– $[\![x\langle w\rangle S]\!] = \big\{(\sigma, E, \ell) \mid \{\sigma(x)\}\langle w\sigma\rangle S\sigma\big\}$

For convenience of notations, we extend the set of formulae SPL as follows:

$$\text{SPL}_{+} \ni \varphi, \psi ::= \ldots \mid (X, x)\langle w\rangle S \mid t\langle w\rangle S$$

The semantics of the newly introduced formulae is:

$$[\![t\langle w\rangle S]\!] = \big\{(\sigma, E, \ell) \mid t\sigma\langle w\sigma\rangle S\sigma\big\}$$
$$[\![(X, x)\langle w\rangle S]\!] = [\![X\langle w\rangle S]\!] \cap [\![x\langle w\rangle S]\!]$$

We prove that any formulae of the form $t\langle w\rangle S$ is definable in SPL.

Proposition 3. *Let s, t be terms, let w be a sequence of term transducers and let $\mathcal{J}(t, w, s)$ be defined as follows:*

$$\mathcal{J}(t,w,s) = \begin{cases} x\langle w\rangle s & \textit{if } t = x \in \mathcal{V} \\ \neg\mu(a, s) & \textit{if } t = a \in \mathcal{A} \\ \mathcal{J}(t_1, w, s) \wedge \mathcal{J}(t_2, w, s) \wedge \neg\mu(t, s) & \textit{if } t = (t_1, t_2) \\ \mathcal{J}(t_1, w, s) \wedge \neg\mu(t, s) & \textit{if } t = \{t_1\}_k \wedge k \notin K \\ \neg\mu(t, s) & \textit{if } t = \{t_1\}_k \wedge k \in K \wedge w = \epsilon \\ ((\mathcal{J}(b|_r, w_1, s) \wedge \mu(b, t)) \vee \neg\mu(b, t)) & \\ \wedge\neg\mu(t, s) & \textit{if } t = \{t_1\}_k \wedge k \in K \wedge w = (b, r).w_1 \end{cases}$$

Then, $t\langle w\rangle s \equiv \mathcal{J}(t, w, s)$, i.e., both formulae are equivalent.

Proof. See Appendix B.5

From now one, we will tacitly identify $t\langle w\rangle S$ and $\mathcal{J}(t, w, s)$. We also use the notations $(\sigma, E, \ell) \models \varphi$ for $(\sigma, E, \ell) \in [\![\varphi]\!]$, $t\langle \not w\rangle S$ for $\neg t\langle w\rangle S$, and $X\langle \not w\rangle S$ for $\neg X\langle w\rangle S$. Also, given s a term, we write $X\langle w\rangle s$ instead of $X\langle w\rangle\{s\}$ and $t\langle w\rangle s$ instead of $t\langle w\rangle\{s\}$. We identify formulae modulo the usual properties of boolean connectives such as associativity and commutativity of $\wedge$, $\vee$, distributivity etc... and use $\Rightarrow$ as the classical logical implication (it can be easily defined in SPL logic using set inclusion).

The predicate $Secret(t)$ can be expressed in SPL, and hence, the specification language of Section 3.4 can be embedded into SPL.

Given a term t, let $F(t)$ denote the formula $\bigvee_{S' \in wc(t)} X\langle\epsilon\rangle S'$. Then we have:

Proposition 4. *Let t be a term. Then, $[\![Secret(t)]\!] = [\![F(t)]\!]$.*

Proof. See Appendix B.6.

Well-formed formulae. In Definition 6, we introduced when $(w_i, S_i)_{i\in I}$ is well-formed. As now we are dealing with formulae, we have to define when a formula is well-formed in the same sense.

Definition 8. *A formula Φ is well-formed, if for any sequence of term transducers w and closed set of terms S, whenever $\Phi \Rightarrow X\langle w\rangle S$, there exist $(w_i, S_i)_{i\in I}$ well-formed, such that $\Phi \Rightarrow \bigwedge_{i\in I} X\langle w_i\rangle S_i$ and $(w, S) \in (w_i, S_i)_{i\in I}$.*

The main property satisfied by well-formed formulae is an a parallel to Proposition 1 and given by the following corollary, which is a direct consequence of Definitions 6 and 8.

Corollary 1. *Let Φ be a well-formed formula such that $\Phi \Rightarrow X\langle w\rangle S$ and let $(\sigma, E, l) \in [\![\Phi]\!]$. If m is a message such that $E\sigma \vdash m$, then $m\langle w\sigma\rangle S\sigma$.*

Now, the property of Corollary 1 turns out to be crucial for developing a complete weakest precondition calculus and well-formedness has to be preserved. Therefore, we introduce the function $\mathcal{H}$. It takes as arguments a formula $X\langle b.w\rangle S$ and computes the weakest (the largest w.r.t. set inclusion) well-formed formula (see Definition 6) $\mathcal{H}(X\langle b.w\rangle S)$, such that $\mathcal{H}(X\langle b.w\rangle S) \Rightarrow X\langle b.w\rangle S$:

$$\mathcal{H}(X\langle b.w\rangle S) = \begin{cases} X\langle b.w\rangle S & \text{if } \mathrm{lpt}(b) \text{ is undefined} \\ X\langle b.w\rangle S \wedge & \\ \left(\mathcal{H}(X\langle b_1.w\rangle S) \vee \bigvee_{S' \in wc(t)} X\langle \epsilon\rangle S'\right) & \text{if } b = (t,p) \wedge b_1 = \mathrm{lpt}(b) \end{cases}$$

Proposition 5. *Let Φ be a well-formed formula. Let $b.w$ be a sequence of term transducers and S a closed set of terms such that $\Phi \Rightarrow X\langle b.w\rangle S$. Then $\Phi \Rightarrow \mathcal{H}(X\langle b.w\rangle S)$.*

Proof. A direct consequence of Definitions 6 and 8.

5 Weakest precondition calculus

We are interested in proving partial correctness of bounded cryptographic protocols w.r.t. pre- and post-condition given by universally quantified SPLformulae. Thus, using the usual notation, we are interested in proving validity of Hoare triples $\{\varphi\}\pi\{\psi\}$. As our formalisation of bounded CP consists of the actions, sequential composition and non-deterministic choice, the Hoare logic contains axioms for the actions and the usual inference rules for composition and choice, and the Consequence rule. The rules are standard. Therefore, we focus on the axioms for the actions. That is, for each action we show that we can express the weakest liberal precondition in SPL.

Let us now precisely define the fragment of SPL for which we develop a complete Hoare Logic. As shown in Section 3.4 most security propeties (authentication and secrecy at least) can be expressed by such formulae. We denote this fragment by $\mathrm{SPL}_\forall$.

$$\varphi, \psi ::= X\langle w\rangle S \mid (X,x)\langle w\rangle S \mid x = t \mid pc = \ell \mid x \neq t \mid \top \mid \bot \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \forall \tilde{x}\varphi$$

Table 3. The set of formulas $\mathrm{SPL}_\forall$

Definition 9 (weakest precondition). *The weakest precondition of a set of configurations $\mathcal{C} \subseteq \mathcal{C}onf$ with respect to an action α, denoted $wlp(\alpha, \mathcal{C})$ is defined to be the set of configurations s, such that whenever action α is allowed in s, it leads to a configuration in $\mathcal{C}$. More formally*

$$wlp(\alpha, \mathcal{C}) ::= \left\{(\sigma, E, l) \mid (\sigma, E, l) \xrightarrow{\alpha} (\sigma', E', l') \;\Rightarrow\; (\sigma', E', l') \in \mathcal{C}\right\}.$$

Given a formula φ, we use $wlp(\alpha, \varphi)$ instead of $wlp(\alpha, [\![\varphi]\!])$ to denote the weakest precondition of a formula $\varphi \in$ SPL.

Let t be a term and p a valid position in t. Then, we denote by $\mathrm{lpp}(t, p)$ the position of the first term transducer in t from above that dominates p if it exists. A formal definition is given in Appendix B.2 but let us give an example.

Example 6. Consider the term $t = (\{A, \{N\}_{k_1}\}_{k_2}, N)$, where $k_1, k_2 \in K$. Let $p = 1121$ and $p' = 2$. Thus, $t_{|p} = t_{|p'} = N$. Then, we have $\mathrm{lpp}(t, p) = 1$, which corresponds to the key k_2; $\mathrm{lpp}(t, p')$ is, however, undefined.

We remind from section 4, that given a term t, $F(t) ::= \bigvee_{S' \in wc(t)} X\langle\epsilon\rangle S'$. The intuitive explanation of next lemma is the following: being in a state (σ, E, l), in order to be able to make an input $t(\tilde{x})$, such that $\tilde{x}$ are instantiated by ρ, it must be that $(\sigma, E, l) \notin [\![F(t\rho)]\!]$.

Lemma 2. *Let E be a set of terms, l be a label and let ρ and σ be ground substitutions such that $dom(\rho) = \tilde{x}$ and $(dom(\sigma) \cup var(E)) \cap \tilde{x} = \emptyset$. Then it holds $(\sigma, E, l) \in [\![F(t\rho)]\!]$ iff $E\sigma \nvdash t(\sigma \oplus \rho)$.*

Proof. See Appendix B.7.

Let t be a term, w a sequence of term transducers and S a set of terms. We denote by $\mathcal{G}(t, w, S)$ the formula obtained from $\bigwedge_{s \in S} \mathcal{J}(t, w, s)$ as follows:

- First, use distributivity of $\wedge$ and $\vee$ to push "inside" $\bigwedge_{s \in S}$ as much as possible.
- Then, replace any occurence of $\bigwedge_{s \in S} x\langle w\rangle s$ by $(X, x)\langle w\rangle S$.

It is easy to prove by induction on the structure of t that $\mathcal{G}(t, w, S) \in \mathrm{SPL}_\forall$, and similar to Proposition 3, we can prove that $X\langle w\rangle S \wedge \mathcal{G}(t, w, S) \equiv X\langle w\rangle S \wedge t\langle w\rangle S$.

Lemma 3 gives the weakest condition that has to be satisfied in a configuration s, such that if in the next step x is instantiated by an input $?t(\tilde{x})$, the reached configuration s' satisfies $x\langle w\rangle S$. The key idea can be explained by considering the sequence of actions $?t(\tilde{x}); !x$. That is, if a secret s that appears in x has to be protected then it has to appear in x under an encryption. Thus, before executing $?t(\tilde{x}); !x$, it should be the case that even if we provide the intruder with the term transducer that takes as input $t(\tilde{x})$ and yields x, it is not possible to derive s.

Lemma 3. *Let t be a term, S a set of terms, w a sequence of term transducers, x a variable and $P_{x,t}$ the set of critical positions of x in t. Let*

$$\mathcal{K}(t, x, w, S) = X\langle w\rangle S \wedge \bigwedge_{p = lpp(t, p_x), p_x \in P_{x,t}} \mathcal{H}(X\langle (t|_p, p^{-1}p_x).w\rangle S).$$

Let E be a set of terms, l and l' labels, and ρ, σ ground substitutions such that $dom(\rho) = \tilde{x}$, $x \in \tilde{x}$, $(dom(\sigma) \cup var(E)) \cap \tilde{x} = \emptyset$. Let Φ a well-formed formula such that whenever $E\sigma \vdash t(\sigma \oplus \rho)$, it holds

$$(\sigma \oplus \rho, E, l') \in [\![(X, x)\langle w\rangle S]\!] \text{ iff } (\sigma, E, l) \in [\![\Phi]\!]$$

Then $[\![\Phi]\!] = [\![\rho(\mathcal{K}(t, x, w, S))]\!]$.

Proof. See Appendix B.8.

Now we are ready to introduce the weakest preconditions for all formulae in $\mathrm{SPL}_\forall$. Remark that in the case of input, $F(t)$ is used for partial correctness: if an input $?t(\tilde{x})$ is not allowed in a configuration s (i.e. it holds $s \in [\![F(t)]\!]$), then for any φ, we have that $s \in wlp(?t(\tilde{x}), \varphi)$.

Definition 10 (definition of $\hat{wlp}$). *The function $\hat{wlp}$, which gives the weakest preconditions for $SPL_\forall$, is defined below:*

1. $\hat{wlp}(l \overset{!t}{\to} l', \varphi) \overset{def}{=} pc = \ell \Rightarrow \varphi \wedge \mathcal{G}(t, w, S)$ *if* $\varphi \in \{X\langle w\rangle S, (X, x)\langle w\rangle S\}$
2. $\hat{wlp}(l \overset{!t}{\to} l', \varphi) \overset{def}{=} pc = \ell \Rightarrow \varphi$ *if* $\varphi \in \{x \neq t', x = t', \top, \bot\}$
3. $\hat{wlp}(l \overset{?t(\tilde{x})}{\to} l', (X, x)\langle w\rangle S) \overset{def}{=} pc = \ell \Rightarrow (F(t) \vee \mathcal{K}(t, x, w, S))$ *if* $x \in \tilde{x}$
4. $\hat{wlp}(l \overset{?t(\tilde{x})}{\to} l', \varphi) \overset{def}{=} pc = \ell \Rightarrow (F(t) \vee \varphi)$ *if* $\varphi \in \{X\langle w\rangle S, (X, y)\langle w\rangle S, x \neq t', x = t', \top, \bot\}$ *and* $y \notin \tilde{x}$
5. $\hat{wlp}(l \overset{x:=t}{\to} l', \varphi) \overset{def}{=} pc = \ell \Rightarrow \varphi[t\sigma/x]$ *if* $\varphi \in \{X\langle w\rangle S, (X, x)\langle w\rangle S, x \neq t', x = t', \top, \bot\}$
6. $\hat{wlp}(l \overset{x=t}{\to} l', \varphi) \overset{def}{=} pc = \ell \Rightarrow (\sigma(x) = t\sigma \Rightarrow \varphi)$ *if* $\varphi \in \{X\langle w\rangle S, (X, x)\langle w\rangle S, x \neq t', x = t', \top, \bot\}$
7. $\hat{wlp}(l \overset{\beta}{\to} l', pc = l'') \overset{def}{=} pc = \ell \Rightarrow \ell' = \ell''$
8. $\hat{wlp}(\alpha, \varphi \vee \psi) \overset{def}{=} \hat{wlp}(\alpha, \varphi) \vee \hat{wlp}(\alpha, \psi)$
9. $\hat{wlp}(\alpha, \varphi \wedge \psi) \overset{def}{=} \hat{wlp}(\alpha, \varphi) \wedge \hat{wlp}(\alpha, \psi)$
10. $\hat{wlp}(\alpha, \forall\tilde{x}\varphi) \overset{def}{=} \forall\tilde{x}\hat{wlp}(\alpha, \varphi)$ *if* $var(\alpha) \cap \tilde{x} = \emptyset$

It is easy to see that for any formula $\varphi \in \mathrm{SPL}_\forall$ and any action α, $\hat{wlp}(\alpha, \varphi) \in \mathrm{SPL}_\forall$. Then, we define the formula $\mathrm{WLP}(\alpha, \varphi)$ as follows: $\mathrm{WLP}(\alpha, \varphi) = \hat{wlp}(\alpha, \varphi)$, if $\alpha \neq l \overset{?t(\tilde{x})}{\to} l'$ and $\mathrm{WLP}(l \overset{?t(\tilde{x})}{\to} l', \varphi) = \forall\tilde{x} \cdot \hat{wlp}(l \overset{?t(\tilde{x})}{\to} l', \varphi)$.

Then, we have the following theorem:

Theorem 1. *The wp-calculus of Definition 10 is sound and complete. I.e., let α be any action and φ any formula in* $\mathrm{SPL}_\forall$. *Then,*

$$wlp(\alpha, [\![\varphi]\!]) = [\![\mathrm{WLP}(\alpha, \varphi)]\!].$$

Proof. See Appendix B.9.

Hence, following the usual completeness proof for Hoare logic, we can prove:

Corollary 2. *The Hoare logic consisting of the inference rules for composition, choice and consequence and the axiom schema $\{\mathrm{WLP}(\alpha, \varphi)\}\alpha\{\varphi\}$, for each action, is sound and complete.*

We show in the next section how this logic provides a decision procedure for bonded protocols.

6 Decidability of SPL

In this section, we study the decidability of the existence of a model (the satisfiability problem) of an SPL formula. We prove decidability of this problem for existential formulae (i.e., formulae in Σ_0) and undecidability in the general case. Notice that since we showed in Section 5 that given a formula φ in $\mathrm{SPL}_\forall$ and a bounded CP π, one can compute $\textsc{wlp}(\pi, \varphi)$, decidability of the satisfiability of existential formulae yields a decision procedure. Indeed, assume that we are given an existential formula ψ and φ in $\mathrm{SPL}_\forall$, assume also that we are given a bounded CP π then $\{\psi\}\pi\{\varphi\}$ is true iff $\psi \wedge \neg\textsc{wlp}(\pi, \varphi)$ is not satisfiable. Notice also that undecidability of SPL entails the non-existence of a complete and effective Hoare logic for bounded CP and SPL.

To prove decidability for existential formulae we follow a rule based approach (e.g., [13, 6] for two nice surveys) i.e.:

1. We introduce a set of formulae in *solved form*. For these formulae it is easy to decide whether a model exists.
2. We introduce a set of rewriting rules to transform any formula in the existential fragment into a solved form.
3. We prove soundness of these rules.
4. We also prove their completeness, i.e, termination for a given control that normal forms are indeed in solved form.

We will encounter two sorts of rewriting rules:

– Deterministic rules are of the form $\varphi \to \varphi'$. They transform a given problem into a single problem. A deterministic rule is sound, if $[\![\varphi]\!] = [\![\varphi']\!]$.
– Non-deterministic rules of the form $\varphi \to \varphi_1, \cdots, \varphi_n$. They transform a given problem into a set of problems. A non-deterministic rule is sound, if $[\![\varphi]\!] = \bigcup_{i=1}^{n} [\![\varphi_i]\!]$.

In this section, we do not consider formulae of the from $pc = \ell$. It will be clear that adding these formulae does not add any technical difficulty; it is only cumbersome to consider them here.

Thus, given a formulae φ with $x_1, \cdots, x_n$ as free variables, a model of φ is pair (σ, E) consisting of a ground substitution σ over $x_1, \cdots, x_n$ and a set E of messages.

6.1 Decidability of Σ_0 formulae

Let ψ be a formula in SPL of the form $\exists x_1, \cdots, x_n \cdot \varphi$, where φ is a *conjunction* of literals, i.e.,

$$X\langle w\rangle S \mid x\langle w\rangle S \mid x = t \mid \top \mid X\langle \not{w}\rangle S \mid x\langle \not{w}\rangle S \mid x \neq t \mid \bot$$

with $x_1, \cdots, x_n$ as first-order free variables.

Notice that the satisfiability of any formula $\exists x_1, \cdots, x_n \cdot \varphi$, where φ is quantifier-free can be reduced to a finite set of satisfiability problems of formulae in the form above.

Solved form A formula is called in solved form if is syntactically equal to $\top$, $\bot$ or $\exists x_1, \cdots, x_n \cdot \varphi$ and φ is of the form:

$$\bigwedge_{i=1}^{n} [\bigwedge_{j=1}^{m_i} x_i \langle \epsilon \rangle t_i^j \wedge \bigwedge_{j=1}^{l_i} x_i \langle \not\epsilon \rangle u_i^j \wedge \bigwedge_{j=1}^{o_i} x_i \neq v_i^j]$$

such that:

- For any $i = 1, \cdots, n$, $x_i \notin var(t_i^j)$, $x_i \notin var(u_i^j)$, and $x_i \notin var(v_i^j)$.
- There is an ordering $x_{i_1}, \cdots, x_{i_n}$ of $x_1, \cdots, x_n$ such that $\bigcup_{k=1}^{l_{i_k}} var(u_{i_k}^k) \cap \{x_{i_{k+1}}, \cdots, x_{i_n}\} = \emptyset$.

We now show how one can "easily" check whether a formula in solved form has a model. We only consider the third type of solved formulae. So, let φ a conjunction as above. We define a particular substitution σ such that φ has a model iff it is satisfied by σ. To do so, let $k \in K$ be a fixed key. Let $F(n)$, for $n \geq 1$, denote n concatenations of k, i.e., $F(1) = k$ and $F(n+1) = \mathbf{pair}(k, F(n))$. Let now N be a natural number strictly bigger than the size of the formula φ. We then define the substitution σ recursively as follows:

- If $n = 1$, i.e., there is only one variable then $\sigma(x_{i_1}) = (u_{i_1}^1, (\cdots, (u_{i_1}^{l_{i_1}}, \{F(N+i_1)\}_k) \cdots))$. In case $l_{i_1} = 0$ this term is understood as $\{F(N + i_1)\}_k$.
- If $n > 1$ then replace x_{i_1} by $\sigma(x_{i_1})$ in φ. This yields a new formula φ' and the ordering $x_{i_2}, \cdots, x_{i_n}$, and by recursion, a substitution σ'. Then, let

$$\sigma = [x_{i_1} \mapsto (u_{i_1}^1, (\cdots, (u_{i_1}^{l_{i_1}}, \{F(N + i_1))\}_k) \cdots))] \oplus \sigma'.$$

Theorem 2. *Let φ be a formula in solved form syntactically different from $\top$ and $\bot$. Let σ be the substitution as defined above. Then, φ has a model iff σ satisfies φ.*

Proof. We only give a sketchy idea of the main argument why the Theorem holds. The interesting implication to prove is the following: If σ does not satisfy φ then φ has no model.

Now, since σ has been defined such that $\sigma(x)$ is not a sub-term of φ, for any $x = x_1, \cdots, x_n$, we have the two crucial properties: 1.) If $u\sigma \langle \not\epsilon \rangle t\sigma$ then $u \langle \not\epsilon \rangle t$ and 2.) If $u\sigma \neq t\sigma$ then $u \neq t$. On the other hand, we can prove if σ is not a model of φ then $u_i^j \sigma \langle \not\epsilon \rangle t_i^q \sigma$, for some $i \in \{1, \cdots, n\}$, $j \in \{1, \cdots, l_i\}$ and $q \in \{1, \cdots, m_i\}$. Therefore, $u_i^j \langle \not\epsilon \rangle t_i^q$, and hence, φ has no model. $\square$

Theorem 3. *Application of the rules of Table 6.1 terminates in a solved form.*

Proof. See Appendix B.10.

If we allow both existential and universal quantifiers, then the decision problem becomes undecidable. Indeed, we can show that Post's correspondence problem is reducible to the decision problem in our logic.

Theorem 4. *Post's correspondence problem is reducible to the decision problem for the SPL logic.*

Proof. See Appendix B.11.

$x = x \mapsto \top$	$x\langle w\rangle x \mapsto \bot$	$\bot \wedge \Phi \mapsto \bot$	$x \neq x \mapsto \bot$	$x\langle \not w\rangle x \mapsto \top$	$\top \wedge \Phi \mapsto \Phi$
$x = t \mapsto \bot$	if $x \in \mathcal{V}ar(t) \wedge x \not\equiv_s t$		$w\langle \epsilon\rangle t \mapsto \top$	if $x \in \mathcal{V}ar(t) \wedge x \not\equiv_s t$	
$x \neq t \mapsto \top$	if $x \in \mathcal{V}ar(t) \wedge x \not\equiv_s t$		$w\langle \not\epsilon\rangle t \mapsto \bot$	if $x \in \mathcal{V}ar(t) \wedge x \not\equiv_s t$	

Table 4. Eliminate trivial sub-formulae

$$x = t \wedge \Phi \quad \mapsto \quad \Phi[t/x] \quad \text{if } x \notin \mathcal{V}ar(t)$$

Table 5. Replacement

$t\langle w\rangle s$	$\mapsto$	$\mathcal{J}(t, w, s)$, if $t \notin \mathcal{X}$	**D1**
$x\langle (b, p).w\rangle s$	$\mapsto$	$x\langle \epsilon\rangle s \wedge x\langle \epsilon\rangle b,\ x\langle \epsilon\rangle s \wedge b\vert_p\langle w\rangle s$	**D2**
$s = t$	$\mapsto$	$\mu(s, t)$ if $s, t \notin \mathcal{X}$	**D3**

Table 6. Decompose

$$\bigwedge_{i\in I} X\langle w_i\rangle s_i \wedge \bigwedge_{j\in J} X\langle \not w_j\rangle s'_j \mapsto \bigwedge_{j\in J} \Big[\bigwedge_{i\in I} z_j\langle w_i\rangle s_i \wedge z_j\langle \not w_j\rangle s'_j \Big]$$

where z_j with $j \in J$ are new variables

Table 7. Elimination X

$$\varphi \mapsto \varphi[y/x]$$

if x and y are syntactically different and $x \leq y$ and $y \leq x$, where $\leq$ is the reflexive transitive closure of $<$ with "$x < y$ iff there there is a sub-formula of φ of the form $y\langle \not w\rangle t$ with $x \in var(t)$".

Table 8. Occur-check

Table 9. Rules for transformations into a solved form

7 Conclusions

We showed that it is possible to have a complete and effective Hoare Logic for bounded cryptographic protocols and an expressive assertion language. This assertion language allows to specify secrecy as well as authentication and other properties. As a consequence of this result, we have a decision procedure for bounded cryptographic protocols and a large class of security properties allowing an infinite set of messages initially known by the intruder. The latter point might seem minor but is not. Indeed, if we are interested in composing protocols we have to take into account that we have no bound on how many sessions have taken place before, and hence, we should allow infinite sets of messages. Thus, in this paper, besides developing (to our knowledge) for the first time a result concerning the existence of an effective and complete Hoare Logic for CP, we significantly extend existing decidability results in two directions: 1.) larger class of properties and 2.) more general initial conditions. We also believe that this paper presents a general framework for a uniform presentation of different decidability results for bounded CP with weaker cryptographic hypothesis, e.g., considering equational theories. In the appendix, we develop this point of view for Cipher Block Chaining and in the full paper for the xor-theory.

The method presented in this paper is a basis for analsying unbounded protocols using approximations as those used in [4], where widening is used to guarantee termination. The interesting results of [2] can be used to restrict the use of the widening operator, and hence, obtain more precise analysis.

References

1. R. M. Amadio and D. Lugiez. On the reachability problem in cryptographic protocols. In *International Conference on Concurrency Theory*, volume 1877 of *LNCS*, pages 380–394, 2000.
2. Bruno Blanchet and Andreas Podelski. Verification of cryptographic protocols: Tagging enforces termination. In Andrew D. Gordon, editor, *FoSSaCS'03: Foundations of Software Science and Computation Structures*, volume 2620 of *LNCS*, pages 136–152. Springer, 2003.
3. M. Boreale. Symbolic trace analysis of cryptographic protocols. In *ICALP: Annual International Colloquium on Automata, Languages and Programming*, 2001.
4. L. Bozga, Y. Lakhnech, and M. Périn. Abstract interpretation for secrecy using patterns. In *TACAS'03*, volume 2619 of *LNCS*, 2003.
5. M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, February 1990.
6. H. Comon. Disunification: A survey. In *Computational Logic: Essays in Honor of Alan Robinson*. MIT Press, Cambridge, MA, 1991.
7. H. Comon and V. Shmatikov. Is it possible to decide whether a cryptographic protocol is secure or not? *Journal of Telecommunications and Information Technology*, 2002.
8. H. Comon-Lundh and V. Cortier. New decidability results for fragments of first-order logic and application to cryptographic protocols. In *14th Int. Conf. Rewriting Techniques and Applications (RTA'2003)*, volume 2706 of *LNCS*, 2003.
9. P. Cousot. Methods and Logics for Proving Programs. In *Handbook of Theoretical Computer Science, Volume B: Formal Methods and Semantics*, pages 841–994. Elsevier Science Publishers B. V., 1990.
10. D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.
11. F.J.T. Fábrega, J.C. Herzog, and J.D. Guttman. Strand Spaces: Why is a Security Protocol Correct ? In *IEEE Conference on Security and Privacy*, pages 160–171, 1998.
12. M. Fiore and M. Abadi. Computing symbolic models for verifying cryptographic protocols. In *14th IEEE Computer Security Foundations Workshop (CSFW '01)*, pages 160–173, Washington - Brussels - Tokyo, June 2001. IEEE.
13. J.-P. Jouannaud and C. Kirchner. Solving equations in abstract algebras: A rule-based survey of unification. In Jean-Louis Lassez and Gordon Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson*. MIT-Press, 1991.
14. G. Lowe. A hierarchy of authentication specifications. In *10th IEEE Computer Security Foundations Workshop (CSFW '97)*, pages 31–44, Washington - Brussels - Tokyo, June 1997. IEEE.
15. C. Meadows. Formal methods for cryptographic protocol analysis: Emerging issues and trends. *IEEE Journal on Selected Areas in Communication*, 21(1):44–54, January 2003.

16. J. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *ACM Conference on Computer and Communications Security*, pages 166–175, 2001.
17. A. W. Roscoe. Intensional specification of security protocols. In *9th IEEE Computer Security Foundations Workshop (CSFW '96)*, pages 28–38, Washington - Brussels - Tokyo, June 1996. IEEE.
18. M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *IEEE Computer Security Foundations Workshop*, 2001.
19. S. Schneider. Verifying authentication protocols with CSP. In *10th IEEE Computer Security Foundations Workshop (CSFW '97)*, pages 3–17, Washington - Brussels - Tokyo, June 1997. IEEE.
20. J. Thayer, J. Herzog, and J. Guttman. Honest Ideals on Strand Spaces. In *IEEE Computer Security Foundations Workshop*, pages 66–78, 1998.
21. K. N. Venkataraman. Decidability of the purely existential fragment of the theory of term algebras. *JACM*, 1987.
22. Thomas Y. C. Woo and Simon S. Lam. Authentication for distributed systems. *Computer*, 25(1):39–52, January 1992.

A Authentication properties

Non-injective agreement on N_a of the initiator is guaranteed to the responder b in session $S1$: if b completes a run of the protocol in session $S1$, as responder, with one participant, let say x, then $y^{(S1)} = x$ and the participant x has previously been running the protocol, as initiator, with b and they have the same value for N_a.

$$pc_B^{(S1)} = 3 \Rightarrow (y^{(S1)} = a \wedge$$

$$((pc_A^{(S1)} \neq 0 \wedge p^{(S1)} = b \wedge z^{(S1)} = {N_a}^{(S1)}) \vee$$

$$(pc_A^{(S2)} \neq 0 \wedge p^{(S2)} = b \wedge z^{(S1)} = {N_a}^{(S2)})))$$

Agreement on N_a *and* N_b of the initiator is guaranteed to the responder b in session $S1$: if b completes a run of the protocol in session $S1$, as responder, with one participant, let say x, then $y^{(S1)} = x$ and the participant x has previously been running the protocol, as initiator, with b and they have the same value for N_a and N_b. Moreover each such run of b corresponds to a unique run of a.

$$pc_B^{(S1)} = 3 \Rightarrow (y^{(S1)} = a \wedge$$

$$((pc_A^{(S1)} \neq 0 \wedge p^{(S1)} = b \wedge z^{(S1)} = {N_a}^{(S1)} \wedge x^{(S1)} = {N_b}^{(S1)} \wedge$$

$$(pc_A^{(S2)} = 0 \vee p^{(S2)} \neq b \vee z^{(S1)} \neq {N_a}^{(S2)} \vee x^{(S2)} \neq {N_b}^{(S1)})) \vee$$

$$(pc_A^{(S2)} \neq 0 \wedge p^{(S2)} = b \wedge z^{(S1)} = {N_a}^{(S2)} \wedge x^{(S2)} = {N_b}^{(S1)} \wedge$$

$$(pc_A^{(S1)} = 0 \vee p^{(S1)} \neq b \vee z^{(S1)} \neq {N_a}^{(S1)} \vee x^{(S1)} \neq {N_b}^{(S1)}))))$$

B Proofs

B.1 Definition of $\mathbf{lpp}(t, p)$

Definition 11 (protecting positions). *Let t be any term. The* **least p-protecting position in** t *(denoted by $lpp(t,p)$) is a position q in t such that the following conditions are satisfied:*

1. *$q \prec p$ and there exists $k \in K$ such that $t|_q \sim \{x\}_k$. Intuitively this means that the intruder has to know the inverse key of k in order to get to $t_{|p}$ by decomposing t.*
2. *For all position $q' \prec q$ and for all $k \in K$ we have $t|_{q'} \not\sim \{x\}_k$, i.e, is the least position in t satisfying the first condition.*

This definition is graphically explained in Fig. 11

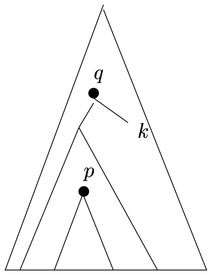

Fig. 1. Least p-protecting position in a term t

B.2 Definition of $\mathbf{lpt}(b,p)$

Definition 12 ( ). *Let* (b,p) *be a term transducer. Then* the next term transducer in b from above that dominates p *(denoted by lpt*(b,p)*) is defined as follows:*

$$lpt(b,p) = \begin{cases} (b|_{1q}, 1q^{-1}p) \ \textit{if} \ \exists lpp(b|_1, 1^{-1}p) = q \\ undefined \quad \textit{otherwise} \end{cases}$$

B.3 Proof of Proposition 1

Proposition 1. *Let* E *be a set of messages such that* $E\langle w_i, S_i\rangle_I$ *and let* $(w_i, S_i)_{i\in I}$ *be well-formed. Moreover, let* m *be a message with* $E \vdash m$*. Then,* $m\langle w_i, S_i\rangle_I$*.*

Proof. Before tackling the proof, we introduce the following definition: We say that m is a *derivation-minimal counter-example*, if the following conditions are satisfied:

1. $E \vdash m$,
2. $\neg E\langle (w_i, S_i)_{i\in I}\rangle_K$ and
3. there is a derivation for $E \vdash m$ which does not contain any strict sub-derivation $E \vdash m'$ of a message m' with $\neg m'\langle (w_i, S_i)_{i\in I}\rangle_K$.

Assume that the assertion does not hold. Then, there exists a derivation-minimal counter-example m. The existence of m can be proved as follows. Take a derivation of $E \vdash m$ and let N_0 be its size. If m is not a derivation-minimal counter-example then there must exist a sub-derivation $E \vdash m'$ with $\neg m'\langle (w_i, S_i)_{i\in I}\rangle_K$. Clearly, the size N_1 of the derivation tree of m' is strictly smaller than N_0. Repeated application of the same argument must lead to a derivation-minimal counter-example as there are no strictly decreasing chains in $\mathbb{N}$.

Thus, let us come back to our derivation-minimal counter-example m. We derive a contradiction by case analysis on the last derivation step in $E \vdash m$.

1. $m \in E$. This, contradicts the assumption $E\langle (w_i, S_i)_{i\in I}\rangle_K$.
2. Case of encryption with a key from K. Thus, $m = \{m_1\}_{k_1}$, $E \vdash m_1$ and $E \vdash k_1$ with $k_1 \in K$. Since m is a derivation-minimal counter-example, we have $m_1\langle (w_i, S_i)_{i\in I}\rangle_K$ and $k_1\langle (w_i, S_i)_{i\in I}\rangle_K$. Since $\neg m\langle (w_i, S_i)_{i\in I}\rangle_K$, there

exists $i \in I$ such that $\neg m\langle(w_i, S_i)\rangle_K$. It follows that $w_i \neq \epsilon$ and hence $w_i = (b, r).w$ and $m = b$ and $\neg b|_r\langle(w_i, S_i)\rangle_K$ $(*)$.
If $\mathrm{lpt}(b, r)$ does not exist then we have $\neg m_1\langle(\epsilon, S_i)\rangle_K$, and hence, $\neg m_1\langle(w_i, S_i)\rangle_K$, which contradicts the derivation-minimality of m.
So, let $(b_1, r_1) = \mathrm{lpt}(b, r)$. From definition, we have that $b|_r = b_1|_{r_1}$ $(**)$
Since $(w_i, S_i)_{i \in I}$ is well-formed, there exists $j \in I$ such that either $b \in S_j$ or $w_j = (b_1, r_1).w$ and $S_i \subseteq S_j$.
If we suppose that $b \in S_j$, since S_j is closed, we obtain that either $m_1 \in S_j$ or $k_1 \in S_j$ and hence either $\neg m_1\langle(w_j, S_j)\rangle_K$ or $\neg k_1\langle(w_j, S_j)\rangle_K$, contradiction.
Hence $w_j = (b_1, r_1).w$ and $S_i \subseteq S_j$. From $m_1\langle(w_j, S_j)\rangle_K$, we obtain $b_1|_{r_1}\langle(w, S_j)\rangle_K$ $(***)$.
From $(*)$, $(**)$ and $(***)$ we obtain a contradiction.
3. Case of encryption with a key which is not in K. Thus, $m = \{m_1\}_{k_1}$, $E \vdash m_1$ and $E \vdash k_1$ with $k_1 \notin K$. Since m is a derivation-minimal counter-example, we have $m_1\langle(w_i, S_i)_{i \in I}\rangle_K$, and then we obtain that $m\langle(w_i, S_i)_{i \in I}\rangle_K$, contradiction.
4. Case of pairing. Similar to the previous case.
5. Case of projection. This also contradicts the derivation-minimality assumption.
6. Case of decryption. Thus, $m_1 = \{m\}_{k_1}$, $E \vdash m_1$ and $E \vdash k_1^{-1}$. Since m is a derivation-minimal counter-example, we have $m_1\langle(w_i, S_i)_{i \in I}\rangle_K$ and $k_1^{-1}\langle(w_i, S_i)_{i \in I}\rangle_K$. If we suppose that $k_1 \notin K$, then we obtain that either $\neg m_1\langle(w_i, S_i)_{i \in I}\rangle_K$ or $m\langle(w_i, S_i)_{i \in I}\rangle_K$, contradiction.
If $k_1 \in K$, since for all $i \in I$, S_i are closed, we obtain that $k_1^{-1} \in S_i$, contradiction with $k_1^{-1}\langle(w_i, S_i)_{i \in I}\rangle_K$.

□

B.4 Proof of proposition 2

In order to proof the Proposition 2 we start with proving the following proposition:

Proposition 6. *Let m, s are two messages and E be a set of messages such that $\mathcal{K} \setminus K^{-1} \subseteq E$. If $\neg m\langle\epsilon\rangle s$ then $E, m \vdash s$.*

Proof. By induction on the structure of m.

1. Case m atomic. Since $\neg m\langle\epsilon\rangle s$ we have $m = s$,so that $E, m \vdash s$.
2. Case of pair $m = (m_1, m_2)$. By definition, from $\neg m\langle\epsilon\rangle s$ we have either $m = s$ or $\neg m_1\langle\epsilon\rangle s \vee \neg m_2\langle\epsilon\rangle s$. If $m = s$ we have $m \vdash s$ else using the induction we have $E, m_1 \vdash s \vee E, m_2 \vdash s$ and we can conclude that $E, m \vdash s$.
3. Case of encrypted message with a key which is not in K, $m = \{m_1\}_{k_1}, k_1 \notin K$. By definition, we have either $m = s$ or $\neg m_1\langle\epsilon\rangle s$. if $m = s$ we have $E, m \vdash s$ else, using the induction we have $E, m_1 \vdash s$. From the hypothesis we know $\mathcal{K} \setminus K^{-1} \subseteq E$ and we are in the case where $k_1 \notin K$. Therefore, $k_1^{-1} \in E$ and consequently $E, \{m_1\}_{k_1} \vdash m_1$. Hence, we obtain $E, m \vdash s$.

4. Case of encrypted message with a key of K, $m = \{m_1\}_{k_1}, k_1 \in K$. By definition we have $\{m_1\}_{k_1}\langle\epsilon\rangle s$ is true for $k_1 \in K$ hence, we are not in the hypothesis of our proposition.

Corollary 3. *Let s be a message and E be a set of messages such that $\mathcal{K}\setminus K^{-1} \subseteq E$. If $E \not\vdash s$ then $E\langle\epsilon\rangle s$.*

Proof. If we suppose that $\neg E\langle\epsilon\rangle s$ we have there is $m \in E$ such that $\neg m\langle\epsilon\rangle s$ and using Proposition 6 we obtain that $E, m \vdash s$. But $m \in E$ hence we have $E \vdash s$, contradiction.

PROPOSITION 2 *Let m be a message and E a set of messages such that $\mathcal{K} \setminus K^{-1} \subseteq E$. Then, $E \not\vdash m$ iff there exists a set of messages $A \in wc(m)$ s.t. $E\langle\epsilon\rangle A$.*

Proof. "$\Rightarrow$": $E \not\vdash m \Rightarrow \exists A \in \mathrm{wc}(m)$ s.t. $E\langle\epsilon\rangle A$

By induction on the structure of m.

1. Case m atomic. Then, $A = \mathrm{wc}(m) = \{\{m\}\}$ and using the Corollary 3 we obtain that $E\langle\epsilon\rangle A$.
2. Case of pair $m = (m_1, m_2)$. From $E \not\vdash m$ we have $E \not\vdash m_1$ or $E \not\vdash m_2$. Using the induction hypothesis we have $\exists A_1 \in \mathrm{wc}(m_1)$ such that $E\langle\epsilon\rangle A_1$ or $\exists A_2 \in \mathrm{wc}(m_2)$ such that $E\langle\epsilon\rangle A_2$. From $E \not\vdash m$ and Corollary 3 we obtain $E\langle\epsilon\rangle m$. Hence, for $A = \{m\} \cup A_1$ or $A = \{m\} \cup A_2$ we have $A \in \mathrm{wc}(m)$ and $E\langle\epsilon\rangle A$.
3. Case of encrypted message with a key K, $m = \{m_1\}_{k_1}$. Similar to the previous case.

"$\Leftarrow$" $\exists A \in \mathrm{wc}(m)$ s.t. $E\langle\epsilon\rangle A \Rightarrow E \not\vdash m$

Let suppose that $E \vdash m$. From hypothesis we have $E\langle\epsilon\rangle A$ and from $A \in \mathrm{wc}(m)$ we have (ϵ, A) well-formed. Hence, using Proposition 1 we obtain that $m\langle\epsilon\rangle A$. But, from $A \in \mathrm{wc}(m)$ we have $m \in A$, contradiction.

B.5 Definability of $t\langle w\rangle S$ in SPL

We prove that any formulas of the form $t\langle w\rangle S$ is definable in SPL.

Proposition 3. *Let s, t be terms, let w be a sequence of term transducers and let $\mathcal{J}(t, w, s)$ be defined as follows :*

$$\mathcal{J}(t,w,s) = \begin{cases} x\langle w\rangle s & \textit{if } t = x \in \mathcal{V} \\ \neg\mu(a,s) & \textit{if } t = a \in \mathcal{A} \\ \mathcal{J}(t_1,w,s) \wedge \mathcal{J}(t_2,w,s) \wedge \neg\mu(t,s) & \textit{if } t = (t_1,t_2) \\ \mathcal{J}(t_1,w,s) \wedge \neg\mu(t,s) & \textit{if } t = \{t_1\}_k \wedge k \notin K \\ \neg\mu(t,s) & \textit{if } t = \{t_1\}_k \wedge k \in K \wedge w = \epsilon \\ ((\mathcal{J}(b|_r,w_1,s) \wedge \mu(b,t)) \vee \neg\mu(b,t)) \wedge \neg\mu(t,s) & \textit{if } t = \{t_1\}_k \wedge k \in K \wedge w = (b,r).w_1 \end{cases}$$

Then, $t\langle w\rangle s \equiv \mathcal{J}(t, w, s)$, i.e., both formulas are equivalent.

Proof. First, notice that if $\mu(t_1, t_2) \neq \bot$, then $(\sigma, E) \in \mu(t_1, t_2)$ iff $t_1\sigma = t_2\sigma$, and if $\mu(t_1, t_2) = \bot$, then for any (σ, E), it holds $t_1\sigma \neq t_2\sigma$. Now we prove by induction on $depth(t) + |w|$ that $t\langle w\rangle s \equiv \mathcal{J}(t, w, s)$.

1. If $t = x \in X$, then $\mathcal{J}(t, w, s) = x\langle w\rangle s = t\langle w\rangle s$.
2. If $t = a \in \mathcal{A}$, then $\mathcal{J}(t, w, s) = \neg\mu(a, s)$. Then we have $(\sigma, E) \in \neg\mu(a, s)$ iff $s\sigma \neq a$ iff $a\langle w\sigma\rangle s\sigma$ iff $(\sigma, E) \in a\langle w\rangle s$.
3. If $t = (t_1, t_2)$, then $\mathcal{J}(t, w, s) = \mathcal{J}(t_1, w, s) \wedge \mathcal{J}(t_2, w, s) \wedge \neg\mu(t, s)$. By induction hypothesis, we have $\mathcal{J}(t_1, w, s) \equiv t_1\langle w\rangle s$ and $\mathcal{J}(t_1, w, s) \equiv t_1\langle w\rangle s$. We obtain
   $(\sigma, E) \in \mathcal{J}(t, w, s)$ iff $(\sigma, E) \in t_1\langle w\rangle s \wedge t_2\langle w\rangle s \wedge \neg\mu(t, s)$ iff $t_1\sigma\langle w\sigma\rangle s\sigma \wedge t_2\sigma\langle w\sigma\rangle s\sigma \wedge t\sigma \neq s\sigma$ iff $t\sigma\langle w\sigma\rangle s\sigma$ iff $(\sigma, E) \in t\langle w\rangle s$.
4. The case $t = \{t_1\}_k \wedge k \notin K$ is similar to the previous one.
5. If $t = \{t_1\}_k \wedge k \in K \wedge w = \epsilon$, then we have $(\sigma, E) \in t\langle w\rangle s$ iff $t\sigma\langle\epsilon\rangle s\sigma$ iff $t\sigma \neq s\sigma$ iff $(\sigma, E) \in \neg\mu(t, s)$ iff $(\sigma, E) \in \mathcal{J}(t, w, s)$.
6. If $t = \{t_1\}_k \wedge k \in K \wedge w = (b, r).w_1$, then $\mathcal{J}(t, w, s) = \neg\mu(t, s) \wedge (\neg\mu(b, t) \vee (\mu(b, t) \wedge \mathcal{J}(b|_r, w_1, s)))$. By induction hypothesis, we have that $b|_r\langle w_1\rangle s \equiv \mathcal{J}(b|_r, w_1, s)$. We obtain
   $(\sigma, E) \in t\langle w\rangle s$ iff $t\sigma\langle\epsilon\rangle s\sigma \wedge (b\sigma = t\sigma \Rightarrow (b|_r)\sigma\langle w_1\sigma\rangle s\sigma)$ iff $t\sigma \neq s\sigma \wedge (b\sigma \neq t\sigma \vee (b\sigma = t\sigma \wedge (b|_r)\sigma\langle w_1\sigma\rangle s\sigma))$ iff $(\sigma, E) \in \neg\mu(t, s) \wedge (\neg\mu(b, t) \vee (\mu(b, t) \wedge b|_r\langle w_1\rangle s))$ iff $(\sigma, E) \in \neg\mu(t, s) \wedge (\neg\mu(b, t) \vee (\mu(b, t) \wedge \mathcal{J}(b|_r, w_1, s)))$ iff $(\sigma, E) \in \mathcal{J}(t, w, s)$.

□

B.6 Proof of Proposition 4

Proposition 4. *Let t be a message. Then, $[\![Secret(t)]\!] = [\![F(t)]\!]$.*

Proof. Using the Definitions 7 and 3, we have
$(\sigma, E, l) \in [\![F(t)]\!]$ iff $(\sigma, E, l) \in \bigcup_{S' \in wc(t)} [\![X\langle\epsilon\rangle S']\!]$ iff
$\exists S' \in wc(t)$ *s.t.* $(\sigma, E, l) \in [\![X\langle\epsilon\rangle S']\!]$ iff
$\exists S' \in wc(t)$ *s.t.* $E\sigma\langle\epsilon\rangle S'\sigma$ iff (using Proposition 4)
$E\sigma \not\vdash t\sigma$ iff $(\sigma, E, l) \in [\![Secret(t)]\!]$.
□

B.7 Proof of Lemma 2

Lemma 2. *Let E be a set of terms, l be a label and let ρ and σ be ground substitutions such that $dom(\rho) = \tilde{x}$ and $(dom(\sigma) \cup var(E)) \cap \tilde{x} = \emptyset$. Then it holds $(\sigma, E, l) \in [\![F(t\rho)]\!]$ iff $E\sigma \not\vdash t(\sigma \oplus \rho)$.*

Proof. Since $(dom(\sigma) \cup var(E)) \cap \tilde{x} = \emptyset$ and using the Definition 7, we have
$(\sigma, E, l) \in [\![F(t\rho)]\!]$ iff $(\sigma, E, l) \in \bigcup_{S' \in wc(t\rho)} [\![X\langle\epsilon\rangle S']\!]$ iff
$\exists S' \in wc(t\rho)$ *s.t.* $(\sigma, E, l) \in [\![X\langle\epsilon\rangle S']\!]$ iff
$\exists S' \in wc(t\rho)$ *s.t.* $E\sigma\langle\epsilon\rangle S'\sigma$ iff
$\exists S' \in wc((t\rho)\sigma)$ *s.t.* $E\sigma\langle\epsilon\rangle S'$ iff
$\exists S' \in wc(t(\rho \oplus \sigma))$ *s.t.* $E\sigma\langle\epsilon\rangle S'$ iff (using Proposition 2)
$E\sigma \not\vdash t(\sigma \oplus \rho)$.
□

B.8 Proof of Lemma 3

Lemma 3. *Let t be a term, S a set of terms, w a sequence of term transducers, x a variable and $P_{x,t}$ the set of critical positions of x in t. Let*

$$\mathcal{K}(t,x,w,S) = X\langle w\rangle S \wedge \bigwedge_{p=lpp(t,p_x),p_x\in P_{x,t}} \mathcal{H}(X\langle (t|_p, p^{-1}p_x).w\rangle S).$$

Let E be a set of terms, l and l' labels and ρ, σ ground substitutions such that $dom(\rho)=\tilde{x}$, $x\in\tilde{x}$, $(dom(\sigma)\cup var(E))\cap\tilde{x}=\emptyset$. Let Φ a well-formed formula such that whenever $E\sigma\vdash\ t(\sigma\oplus\rho)$, it holds

$$(\sigma\oplus\rho,E,l')\in[\![(X,x)\langle w\rangle S]\!] \text{ iff } (\sigma,E,l)\in[\![\Phi]\!]$$

Then $[\![\Phi]\!]=[\![\rho(\mathcal{K}(t,x,w,S))]\!]$.

Proof. "$\Leftarrow$"

Let suppose that $(\sigma,E,l)\in[\![\rho(\mathcal{K}(t,x,w,S))]\!]$. Since $(\sigma,E,l)\in[\![X\langle w\rho\rangle S\rho]\!]$, it follows that $E\sigma\langle w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$.

It remains to prove that $\rho(x)\langle w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$.

We have that $E\sigma\vdash\ t(\sigma\oplus\rho)$.

From $(\sigma,E,l)\in[\![\mathcal{H}(X\langle (t|_p,p^{-1}p_x).w\rangle S)]\!]$ it follows that $E\sigma\langle (t(\sigma\oplus\rho)|_p,p^{-1}p_x).w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$. By construction, the formula $\mathcal{H}(X\langle (t|_p,p^{-1}p_x).w\rangle S)$ is well-formed. Using Corollary 1, we obtain $t(\sigma\oplus\rho)\langle (t(\sigma\oplus\rho)|_p,p^{-1}p_x).w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$, and from Definition 4 we obtain $\rho(x)\langle w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$.

"$\Rightarrow$"

We have that $(\sigma,E,l')\in[\![\rho(x)\langle w\rho\rangle S\rho\wedge X\langle w\rho\rangle S\rho]\!]$. Let suppose that $\forall p_x\in P_{x,t}\exists \mathrm{lpt}(t,p_x)$.

We have to prove that $(\sigma,E,l)\in[\![X\langle w\rho\rangle S\rho]\!]$ and $(\sigma,E,l)\in[\![\mathcal{H}(X\langle (t|_p,p^{-1}p_x).w\rangle S)]\!]$.

From $(\sigma,E,l')\in[\![\rho(x)\langle w\rho\rangle S\rho\wedge X\langle w\rho\rangle S\rho]\!]$ we obtain that $E\sigma\langle w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$ and $\rho(x)\langle w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$.

It remains to prove that $(\sigma,E,l)\in[\![\mathcal{H}((X\langle (t|_p,p^{-1}p_x).w\rangle S)]\!]$. Firstly we prove that $E\sigma\langle (t(\sigma\oplus\rho)|_p,p^{-1}p_x).w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$.

If we suppose that $\neg E\sigma\langle (t(\sigma\oplus\rho)|_p,p^{-1}p_x).w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$ it means that $\exists m\in E$ such that $\neg m\sigma\langle (t(\sigma\oplus\rho)|_p,p^{-1}p_x).w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$, and using the Definition 4 we obtain that either $\neg m\sigma\langle w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$ or $\neg\rho(x)\langle w(\sigma\oplus\rho)\rangle S(\sigma\oplus\rho)$, contradiction.

Now the assertion follows from the Proposition 5.

The case $\exists p_x\in P_{x,t}\ \nexists \mathrm{lpt}(t,p_x)$ is similar.

□

B.9 Proof of Theorem 1

To prove the main Theorem, we need an auxiliary Lemma.

Lemma 4. *Let $\rho\in\Gamma(\tilde{x})$. Then $\rho(\hat{wlp}(l\overset{?t(\tilde{x})}{\rightarrow}l',\varphi))\equiv(pc=l)\Rightarrow(F(t\rho)\vee\rho(\varphi))$*

Proof. By induction on the structure of φ. In the sequel, we use implicitly that σ is a ground substitution such that $dom(\sigma) \cap dom(\rho) = \emptyset$.

- $\varphi = X\langle w\rangle S$.
  Then
  $\rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi)) = \rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', X\langle w\rangle S)) = \rho((pc = l) \Rightarrow (F(t) \vee X\langle w\rangle S)) = (pc = l) \Rightarrow (F(t\rho) \vee \rho(X\langle w\rangle S)) = (pc = l) \Rightarrow (F(t\rho) \vee \rho(\varphi))$.
- $\varphi = (X, x)\langle w\rangle S$ and $x \in \tilde{x}$.
  Obvious using Lemma 2 and Lemma 3.
- $\varphi = (X, x)\langle w\rangle S$ and $x \notin \tilde{x}$.
  Obvious.
- $\varphi = x \neq t'$
  Obvious.
- $\varphi = x = t'$
  Obvious.
- $\varphi = \top$
  Obvious (using that for any formulas ψ, it holds $\psi \vee \top \equiv \top$).
- $\varphi = \bot$
  Obvious (using that for any formulas ψ, it holds $\psi \vee \bot \equiv \psi$).
- $\varphi = \varphi_1 \vee \varphi_2$
  Then using the induction we obtain
  $\rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi)) = \rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi_1 \vee \varphi_2)) \equiv \rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi_1)) \vee \rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi_2)) \equiv$
  $((pc = l) \Rightarrow (F(t\rho) \vee \rho(\varphi_1))) \vee ((pc = l) \Rightarrow (F(t\rho) \vee \rho(\varphi_2))) \equiv (pc = l) \Rightarrow ((F(t\rho) \vee \rho(\varphi_1)) \vee (F(t\rho) \vee \rho(\varphi_2))) \equiv (pc = l) \Rightarrow (F(t\rho) \vee (\rho(\varphi_1) \vee \rho(\varphi_2))) \equiv$
  $(pc = l) \Rightarrow (F(t\rho) \vee \rho(\varphi))$
- $\varphi = \varphi_1 \wedge \varphi_2$
  Then using the induction we obtain
  $\rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi)) = \rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi_1 \wedge \varphi_2)) \equiv \rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi_1)) \wedge \rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi_2)) \equiv$
  $((pc = l) \Rightarrow (F(t\rho) \vee \rho(\varphi_1))) \wedge ((pc = l) \Rightarrow (F(t\rho) \vee \rho(\varphi_2))) \equiv (pc = l) \Rightarrow ((F(t\rho) \vee \rho(\varphi_1)) \wedge (F(t\rho) \vee \rho(\varphi_2))) \equiv (pc = l) \Rightarrow (F(t\rho) \vee (\rho(\varphi_1) \wedge \rho(\varphi_2))) \equiv$
  $(pc = l) \Rightarrow (F(t\rho) \vee \rho(\varphi))$
- $\varphi = \forall \tilde{y} \varphi_1$.
  We can suppose that $var(t) \cap \tilde{y} = \emptyset$. Then, using that ρ and ρ_1 are ground substitutions such that $dom(\rho) \cap dom(\rho_1) = \emptyset$, we obtain
  $[\![\rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi))]\!] = [\![\rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \forall\tilde{y}\varphi_1)]\!] = [\![\rho(\forall\tilde{y}(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi_1)))]\!] =$
  $\rho(\bigcap_{\rho_1 \ \in \Gamma(\tilde{y})} [\![\rho_1(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi_1))]\!]) = \bigcap_{\rho_1 \ \in \Gamma(\tilde{y})} [\![\rho(\rho_1(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi_1)))]\!] =$
  $\bigcap_{\rho_1 \ \in \Gamma(\tilde{y})} [\![\rho_1(\rho(w\hat{l}p(l \overset{?t(\tilde{x})}{\rightarrow} l', \varphi_1)))]\!] =$
  $\bigcap_{\rho_1 \ \in \Gamma(\tilde{y})} [\![\rho_1((pc = l) \Rightarrow (F(t\rho) \vee \rho(\varphi_1)))]\!] =$
  $\bigcap_{\rho_1 \ \in \Gamma(\tilde{y})} ([\![\rho_1(pc \neq l)]\!] \cup ([\![\rho_1(F(t\rho))]\!] \cup [\![\rho_1(\rho(\varphi_1))]\!])) =$
  $\bigcap_{\rho_1 \ \in \Gamma(\tilde{y})} ([\![pc \neq l]\!] \cup ([\![F(t\rho)]\!] \cup [\![\rho_1(\rho(\varphi_1))]\!])) =$
  $[\![pc \neq l]\!] \cup ([\![F(t\rho)]\!] \cup \bigcap_{\rho_1 \ \in \Gamma(\tilde{y})} [\![\rho(\rho_1(\varphi_1))]\!]) =$

$[\![pc \neq l]\!] \cup ([\![F(t\rho)]\!] \cup \rho(\bigcap_{\rho_1 \in \Gamma(\tilde{y})} [\![\rho_1(\varphi_1)]\!])) = [\![pc \neq l]\!] \cup ([\![F(t\rho)]\!] \cup \rho([\![\forall \tilde{y} \varphi_1]\!])) = [\![pc \neq l]\!] \cup ([\![F(t\rho)]\!] \cup \rho([\![\varphi]\!])) = [\![pc \neq l]\!] \cup ([\![F(t\rho)]\!] \cup [\![\rho(\varphi)]\!]) = [\![(pc = l) \Rightarrow (F(t\rho) \vee \rho(\varphi))]\!]$

and hence we obtain that $\rho(\hat{wlp}(l \stackrel{?t(\tilde{x})}{\rightarrow} l', \varphi)) \equiv (pc = l) \Rightarrow (F(t\rho) \vee \rho(\varphi))$.

□

Theorem 1. *The wp-calculus of Definition 10 is sound and complete. I.e., let α be any action and φ any formula in* SPL$_\forall$. *Then,*

$$wlp(\alpha, [\![\varphi]\!]) = [\![\mathrm{WLP}(\alpha, \varphi)]\!].$$

Proof. The proof is analysis by cases, depending on action α.

1. The equivalence $wlp(l \stackrel{\beta}{\rightarrow} l', [\![\varphi]\!]) = [\![\mathrm{WLP}(l \stackrel{\beta}{\rightarrow} l', \varphi)]\!]$ is obvious, for the case $\beta \in \{x := t, x = t\}$.
2. The equivalence $wlp(l \stackrel{!t}{\rightarrow} l', [\![\varphi]\!]) = [\![\mathrm{WLP}(l \stackrel{!t}{\rightarrow} l', \varphi)]\!]$ is an immediate consequence of the equivalence $X\langle w\rangle S \wedge \mathcal{G}(t, w, S) \equiv X\langle w\rangle S \wedge t\langle w\rangle S$.
3. Since $\mathrm{WLP}(l \stackrel{?t(\tilde{x})}{\rightarrow} l', \varphi) = \forall \tilde{x} \cdot \hat{wlp}(l \stackrel{?t(\tilde{x})}{\rightarrow} l', \varphi)$ we have to prove that

$$wlp(l \stackrel{?t(\tilde{x})}{\rightarrow} l', [\![\varphi]\!]) = [\![\forall \tilde{x} \hat{wlp}(l \stackrel{?t(\tilde{x})}{\rightarrow} l', \varphi)]\!],$$

In the sequel l'' is a label, E is a set of terms and σ is a ground substitution such that $\tilde{x} \cap (dom(\sigma) \cup var(E)) = \emptyset$.
$(\sigma, E, l'') \in wlp(l \stackrel{?t(\tilde{x})}{\rightarrow} l', [\![\varphi]\!])$ iff
$\forall \rho \in \Gamma(\tilde{x})$ whenever $l'' = l$ it holds $E\sigma \vdash t(\sigma \oplus \rho) \Rightarrow (\sigma \oplus \rho, E, l'') \in [\![\varphi]\!]$ iff
$\forall \rho \in \Gamma(\tilde{x})$ whenever $l'' = l$ it holds $E\sigma \not\vdash t(\sigma \oplus \rho) \vee (\sigma \oplus \rho, E, l'') \in [\![\varphi]\!]$ iff
$\forall \rho \in \Gamma(\tilde{x})$ whenever $(\sigma, E, l'') \in [\![pc = l]\!]$ it holds $(\sigma, E, l'') \in [\![F(t\rho)]\!]$ or $(\sigma, E, l'') \in [\![\rho(\varphi)]\!]$ iff
$(\sigma, E, l'') \in \bigcap_{\rho \in \Gamma(\tilde{x})} [\![(pc = l) \Rightarrow (F(t\rho) \vee \rho(\varphi))]\!]$ iff
$(\sigma, E, l'') \in \bigcap_{\rho \in \Gamma(\tilde{x})} [\![\rho(\hat{wlp}(l \stackrel{?t(\tilde{x})}{\rightarrow} l', \varphi)]\!]$ iff
$(\sigma, E, l'') \in [\![\forall \tilde{x}(\hat{wlp}(l \stackrel{?t(\tilde{x})}{\rightarrow} l', \varphi)]\!]$.

□

B.10 Decidability of the existential fragment of SPL

Theorem 5. *Application of the rules of Table 6.1 terminates in a solved form.*

Proof. let us first briefly mention how each rule contributes in reaching a normal form:

1. Rule **Eliminate** X can be applied only once.

2. Rule **D1** decreases the number of sub-formulae of the form $t\langle w\rangle s$ but may introduce equalities and disequalities as well as new formulae of the form $x\langle w\rangle s$.
3. Rule **D2** decreases the length of w in sub-formulae of the form $x\langle w\rangle s$ but introduces new formulae of the form $t\langle w\rangle s$.
4. Rule **D3** decreases the number of equalities (disequalities) where the two members are not variables.
5. Rules **Occur-check** and **Replacement** eliminate a variable.

Now, to prove termination we need to introduce interpretation functions which are intended to decrease by applications of the rules:

- $f_1(\varphi)$ is the number of occurrences of X (in φ).
- $f_2(\varphi)$ is the cardinality of $var(\varphi)$.
- $f_3(\varphi)$ is the multi-set of pairs $(|w|, |t|)$, where $t\langle w\rangle s$ is a sub-formula. For the variation of this function, we take the multi-set extension of the lexicographic ordering on $\mathbb{N}^2$. This is a well-ordering.
- $f_4(\varphi)$ is the number of equations (disequations) where both members are not variables.
- $f_5(\varphi)$ is the size of φ.

Figure 2 summarizes the variation of each function by the transformation rules: Thus, if we define $F(\varphi) = (f_1(\varphi), \cdots, f_5(\varphi))$ then $F(\varphi)$ decreases with respect

	f_1	f_2	f_3	f_4	f_5
T	$=$	$\leq$	$=$	$=$	$<$
D_3	$=$	$=$	$=$	$<$	
$D_1 + D_2$	$=$	$=$	$<$		
R	$=$	$<$			
OC	$=$	$<$			
EX	$<$				

Fig. 2. Variation of the ranking functions

to lexicographic ordering by each rule. Hence, termination of the rules.

It remains now to show that if no rule can be applied then the obtained formula is in solved form. This proof is easy and tedious and is not given in this abstract. □

B.11 Proof of undecidability for the entire SPL logic

Theorem 4. *Post's correspondence problem is reducible to the decision problem for the SPL logic.*

Proof. The proof is inspired from [21], where it is shown the undecidability of a certain fragment in the theory of free term algebras.

Let $P = \{(p_i, q_i) | i = 1, \ldots, n\}$ be an instance of Post's correspondence problem, where $p_i, q_i \in D^*$, with $D = \{d_1, \ldots, d_k\}$. We use $d_1, \ldots, d_k$ as constants, and also let c be another particular constant.

We denote $f(x_1, x_2, x_3) = \mathbf{pair}(\mathbf{pair}(x_1, x_2), x_3)$ and for $i = 1, \ldots k$, we denote, $g_i(x) = \mathbf{pair}(d_i, x)$. The monadic functions g_i represent the alphabet as follows: the string $d_{i_1} \ldots d_{i_j}$ is represented by the term $g_{i_1}(\ldots(g_{i_j}(c))\ldots)$. The use of the function f will be clear later.

Suppose that P has a solution $i_1, \ldots i_m$, that is $m > 0$ and $1 \leq i_j \leq n$ for each j and $p_{i_1} \ldots p_{i_m} = q_{i_1} \ldots q_{i_m}$. For each $j = 1, \ldots, m+1$, let $r_j = p_{i_j} \ldots p_{i_m}$ and $s_j = q_{i_j} \ldots q_{i_m}$. Thus $r_1 = s_1$ and $r_{m+1} = s_{m+1} = \epsilon$. Then the formula Φ_P given below is satisfiable, with the following value for x:

$$x = f(r_1, s_1, f(r_2, s_2, f(\ldots f(r_{m+1}, s_{m+1}, c)\ldots))).$$

Conversely, if the formula is satisfiable, then from the value of x a solution to P can be recovered.

The formula Φ_P is
$\exists x, x_1, x_2 \forall y_0, \ldots, y_6$

$$[Is_{fgc}(x) \wedge \bigwedge_{i=0}^{6} x \langle \not\epsilon \rangle y_i] \tag{1}$$

$$\wedge [x = f(x_1, x_1, x_2) \wedge x_1 \neq c] \tag{2}$$

$\wedge [y_0 \neq f(y_1, y_2, y_3) \vee$

$$[y_1 \neq f(y_4, y_5, y_6) \wedge y_2 \neq f(y_4, y_5, y_6) \wedge \bigwedge_{i=1}^{k} y_3 \neq g_i(y_4)]] \tag{3}$$

$$\wedge [y_1 \neq f(y_2, y_3, c) \vee y_2 = y_3 = c] \tag{4}$$

$$\wedge [y_0 \neq f(y_1, y_2, f(y_3, y_4, y_5)) \vee \bigvee_{i=1}^{k} [y_1 = g_i(y_3) \wedge y_2 = g_i(y_4)]] \tag{5}$$

where

$$Is_{fgc}(x) ::= [Is_{gc}(x) \vee P_3(x)] \wedge \forall y [x \langle \epsilon \rangle y \vee Is_{gc}(y) \vee P_3(y)]$$

$$P_3(x) ::= \exists x_1, x_2, x_3 [x = f(x_1, x_2, x_3)]$$

$$Is_{gc}(x) ::= M(x) \wedge \forall y [x \langle \epsilon \rangle y \vee M(y)]$$

$$M(x) ::= x = c \vee \exists y [\bigvee_{i=1}^{k} x = g_i(y)]$$

The meaning of each sub-formula is given below:

1. $Is_{gc}(t)$ means that t is either c or has the form $g_{i_1}(\ldots(g_{i_p}(c)))$.
2. $Is_{fgc}(t)$ means that t is either c or has the form $g_{i_1}(\ldots(g_{i_p}(c)))$ or the form $f(t_1, t_2, t_3)$ and for the last case, the same property holds for t_1, t_2 and t_3 too.
3. (1) $y_0, \ldots, y_6$ are sub terms of x, and x and also any subterm of x are builded using only the constant c and the "function symbols" g_i and f; moreover, any subterm that has a g_i as the outermost function symbol, has the form $g_{i_1}(\ldots(g_{i_p}(c)))$.
4. (2) This forces $r_1 = s_1$.
5. (3) For any subterm $f(y_1, y_2, y_3)$ of x, y_1 and y_2 must be c or have one of the g_i as the outermost "function symbol", and y_3 must be c or have f as the outermost symbol.
6. (4) This forces $r_{m+1} = s_{m+1} = \epsilon$.
7. (5) For each j there is an i such that $r_j = p_i r_{j+1}$ and $s_j = q_i q_{j+1}$.

$\square$

C Beyond Dolev-Yao model

In this section we sketch how we can extend our logic to deal with models more general than the Dolev-Yao model. As an example we take protocols which use Cypher Block Chaining encryptions.

The intruder capabilities presented in subsection 3.2 are extended by the following rule (for simplicity reasons we suppose that all encryptions are CBC encryptions):

– If $E \vdash \mathbf{encr}(\mathbf{pair}(m_1, m_2), k)$ then $E \vdash \mathbf{encr}(m_1, k)$.

The definition 4 can be easily adapted for CBC.

Definition 13. *Let m and s be two messages and let $w \in (\mathcal{M} \times \mathcal{P}os)^*$ be a sequence of term transducers. We keep the same notation $m\langle w\rangle s$, for the predicate "s is w-protected in m". Thus, $m\langle w\rangle s$ is the strongest predicate (smallest w.r.t. set inclusion) satisfying the following conditions:*

– *m is atomic and $m \neq s$, or*
– *$m = \mathbf{pair}(m_1, m_2)$, $m \neq s$ and both $m_1\langle w\rangle s$ and $m_2\langle w\rangle s$ are true, or*
– *$m = \mathbf{encr}(m_1, k)$, $m \neq s$, $k \notin K$ and $m_1\langle w\rangle s$ is true, and if $m_1 = \mathbf{pair}(m_2, m_3)$, then $\mathbf{encr}(m_2, k)\langle w\rangle s$ is true or*
– *$m = \mathbf{encr}(m_1, k)$, $m \neq s$, $k \in K$ and $w = \epsilon$, and if $m_1 = \mathbf{pair}(m_2, m_3)$, then $\mathbf{encr}(m_2, k)\langle \epsilon\rangle s$ is true or*
– *$m = \mathbf{encr}(m_1, k)$, $w = (b, r).w_1$, $m \neq s$, $k \in K$, and $m \neq b$ or $m|_r\langle w_1\rangle s$ is true and if $m_1 = \mathbf{pair}(m_2, m_3)$, then $\mathbf{encr}(m_2, k)\langle (b, r).w_1\rangle s$ is true.*

As in the Dolev-Yao model, this definition is easily generalized to sets of messages: Let M and S be sets of messages, w a sequence of term transducers and K a set of keys. We say that the secrets S are w - protected in M denoted by $M\langle w\rangle S$, if it holds $\bigwedge_{m\in M, s\in S} m\langle w\rangle s$.

Example 7. Let us consider $m = (\{\{N\}_{k_1}, A\}_{k_2}, B)$, $K = \{k_1, k_2\}$ and the formula $m\langle w\rangle N$.

First, let take $w = \epsilon$. We have that $\{\{N\}_{k_1}, A\}_{k_2}\langle\epsilon\rangle N$, $\{\{N\}_{k_1}\}_{k_2}\langle\epsilon\rangle N$ and $B\langle\epsilon\rangle N$ are true. Hence we obtain that the formula $m\langle\epsilon\rangle N$ is true.

Second, let take $w = (\{\{N\}_{k_1}\}_{k_2}, 11)$. Let suppose that $m\langle w\rangle N$ is true. Then it must be that $\{\{N\}_{k_1}, A\}_{k_2}\langle w\rangle N$, which means that $\{\{N\}_{k_1}\}_{k_2}\langle w\rangle N$ must be true too. So we obtain $N\langle\epsilon\rangle N$, contradiction. Hence $m\langle w\rangle N$ is false.

It is easy to prove that Propositions 1 and 2 still hold in the CBC model.

The SPL logic is extended with respect to SPL^+ as follows:

$$\text{SPL}_{CBC} \ni \varphi, \psi ::= \ldots \mid s \leq_1 t \mid s \not\leq_1 t$$

The semantics of the newly introduced formulae is:

$$[\![s \leq_1 t]\!] = \{(\sigma, E, l) \mid s\sigma \in \mathcal{P}^*(t\sigma)\}$$

$$[\![s \not\leq_1 t]\!] = \{(\sigma, E, l) \mid s\sigma \notin \mathcal{P}^*(t\sigma)\}$$

where $\mathcal{P}^*(m) = \bigcup_{n\in\mathbb{N}} \mathcal{P}^n(m)$, with $\mathcal{P}^0(m) = m$, $\mathcal{P}^{n+1}(m) = \mathcal{P}(\mathcal{P}^n(m))$ and $\mathcal{P}$ is the function defined below:

$$\mathcal{P}(m) = \begin{cases} m_1 & \text{if } m = \mathbf{pair}(m_1, m_2) \\ undefined & otherwise \end{cases}$$

Proposition 3 changes as follows:

Proposition 7. *Let s, t be terms, let w be a sequence of term transducers and let $\mathcal{J}(t, w, s)$ be defined as follows :*

$$\mathcal{J}(t,w,s) = \begin{cases} x\langle w\rangle s & \textit{if } t = x \in X \\ \neg\mu(a,s) & \textit{if } t = a \in \mathcal{A} \\ \mathcal{J}(t_1,w,s) \wedge \mathcal{J}(t_2,w,s) \wedge \neg\mu(t,s) & \textit{if } t = (t_1,t_2) \\ \mathcal{J}(t_1,w,s) \wedge \neg\mu(t,s) \wedge \forall z(\neg\mu(s,\{z\}_k) \vee z \not\leq_1 t_1) & \textit{if } t = \{t_1\}_k \wedge k \notin K \\ \neg\mu(t,s) \wedge \forall z(\neg\mu(s,\{z\}_k) \vee z \not\leq_1 t_1) & \textit{if } t = \{t_1\}_k \wedge k \in K \wedge w = \epsilon \\ ((\mathcal{J}(b|_r,w_1,s) \wedge b_1 \leq_1 t_1) \vee b_1 \not\leq_1 t_1) \wedge \neg\mu(t,s) \wedge & \\ \forall z(\neg\mu(s,\{z\}_k) \vee z \not\leq_1 t) & \textit{if } t = \{t_1\}_k \wedge k \in K \\ & \wedge w = (b,r).w_1 \wedge b = \{b_1\}_k \end{cases}$$

Then, $t\langle w\rangle s \equiv \mathcal{J}(t,w,s)$, i.e., both formulas are equivalent.

Proof. Similar to the proof of Proposition 3.

The weakest precondition calculus is easily extended: if $\varphi \in \{s \leq_1 t, s \not\leq_1 t\}$ then

$\hat{wlp}(l \xrightarrow{!t} l', \varphi) \stackrel{def}{=} pc = \ell \Rightarrow \varphi$

$\hat{wlp}(l \xrightarrow{?t(\tilde{x})} l', \varphi) \stackrel{def}{=} pc = \ell \Rightarrow (F(t) \vee \varphi)$

Theorem 1 still holds in the CBC model.

The Σ_0 fragment for the extended SPL logic still remains decidable. For lack of space, we do not give the proof of this result. We just present the definition of solved forms. A formula is called in solved form if is syntactically equal to $\top$, $\bot$ or $\exists x_1, \cdots, x_n \cdot \varphi$ and φ is of the form:

$$\bigwedge_{i=1}^{n} [\bigwedge_{j=1}^{m_i} x_i \langle \epsilon \rangle t_i^j \wedge \bigwedge_{j=1}^{l_i} x_i \langle \not\epsilon \rangle u_i^j \wedge \bigwedge_{j=1}^{o_i} x_i \neq v_i^j \wedge w_i \leq_1 x_i \wedge \bigwedge_{j=1}^{n_i} w_i^j \not\leq_1 x_i]$$

such that:

- For any $i = 1, \cdots, n$, $x_i \notin var(t_i^j)$, $x_i \notin var(u_i^j)$, $x_i \notin var(v_i^j)$, and $x_i \notin var(w_i^j)$.
- There is an ordering $x_{i_1}, \cdots, x_{i_n}$ of $x_1, \cdots, x_n$ such that $\bigcup_{k=1}^{l_{i_k}} var(u_{i_k}^k) \cap \{x_{i_{k+1}}, \cdots, x_{i_n}\} = \emptyset$.