Unité Mixte de Recherche 5104 CNRS - INPG - UJF
Centre Equation
2, avenue de VIGNATE
F-38610 GIERES
tel : +33 456 52 03 40
fax : +33 456 52 03 50
http://www-verimag.imag.fr

A symbolic decision procedure for cryptographic protocols with time stamps

L. Bozga, C. Ene and Y. Lakhnech

Report n^{o} 02

October 15, 2004

A symbolic decision procedure for cryptographic protocols with time stamps

L. Bozga, C. Ene and Y. Lakhnech

VERIMAG
2, av. de Vignates, 38610 Gières - FRANCE
{fistname.lastname}@imag.fr

October 15, 2004

Abstract

We present a symbolic decision procedure for time-sensitive cryptographic protocols. We consider protocols described in a process algebra-like notation that includes clocks, time-stamps and time variables. While the values of all clocks increase with rate one when time passes, time variables are simply variables that range over the time domain and can be used to remember time-stamps, i.e. time values. Our symbolic decision procedure deals with secrecy, authentication and any property that can be described as a safety property. Our approach is based on a logic representation of sets of configurations that combines a decidable logic with time constraints.

Keywords: Security, Cryptographic Protocols, Time stamps. Decidability, Verification

Reviewers: X

Notes:

How to cite this report:

@techreport { verimag-TR-2004-02,
title = { A symbolic decision procedure for cryptographic protocols with time stamps},
authors = { L. Bozga, C. Ene and Y. Lakhnech},
institution = { Verimag Technical Report },
number = {02},
year = { 2004},
note = { }
}

1 Introduction

Some cryptographic protocols rely upon timestamps that recipients use to verify timeliness of the message and recognize and reject replays of messages communicated in the past. Timestamps are also used in conjunction with short term keys. The presence of timestamps makes the specification and verification of cryptographic protocols a challenging problem. Indeed, most of the existing verification methods and decidability results for cryptographic protocols consider time-independent protocols [29, 26, 3, 5, 23, 17, 12, 13]. Because of the subtleties and complexity of the verification of time-dependent protocols, theorem provers have been used to verify such protocols.

In this paper, we present a model for time-dependent cryptographic protocols and a corresponding decidability result. Although, the model we present only deals with bounded protocols, that is, when a fixed number of sessions are considered, our model clearly identifies the main ingredients to be included in a general model. It is well-known that the verification problem of unbounded cryptographic protocols is undecidable in the untimed case, and hence, it is so for the timed case. Besides general models for distributed systems that can be used to model security protocols such as Timed CSP and MSR (multiset rewriting over first-order atomic formulae), we do not know about a model for timed cryptographic protocols.

To model timed cryptographic protocols, we include in our model clocks, time variables and timestamps. Clocks are variables that range over the time domain and advance with the same rate as time. Each agent has its own set of clocks that he can reset. That is clocks can be used to measure the time that elapses between two events, for instance, sending a message and receiving the corresponding response. Also, we allow a global clock that is never reset and that can be read and tested by all participants. Time variables correspond to timestamps in received messages. Such values can be stored and used together with clocks to put conditions on the acceptance of a message.

A second contribution of this paper is a complete and sound symbolic verification algorithm for timed cryptographic protocols. We consider a rich class of reachability properties that allow to specify confidentiality and authentication. In fact, we introduce a logic that allows to describe secrecy, equalities between terms and control points. Then, given a bounded protocol Π and two formulae in this logic Φ and Ψ, the reachability problem we consider is whether there is a run of Π that starts in a configuration that satisfies Φ and reaches a configuration that satisfies Ψ.

We devise a symbolic algorithm that given a property described by a formula Ψ in this logic and given a bounded protocol computes the set of configurations that reaches Ψ. This algorithm uses symbolic constraints (logic formulae) to describe sets of configurations. The logic we introduce combines constraints on the knowledge of the intruder with time constraints on clock values and time variables. To show effectiveness of our verification method we show:

1. that for each action of our model we can express the predecessor configurations of a set of configurations as a formula. We consider input, output and time actions.
2. Then, we show decidability of the satisfiability problem for our logic.

Related work Our model is clearly inspired by timed automata and our verification method influenced by the work on symbolic verification of timed automata and temporal logics for real-time systems (e.g.[1, 19, 2, 6]).

The results of this paper provide an algorithm for checking security properties (confidentiality and authentication) of timed 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.

Handling time constraints, unbounded message size symbolically and automatically is the distinguishing feature of our verification method. Most of the work on timed cryptographic protocols uses theorem-provers or finite-state model-checking [4, 10, 16, 21]. While the first needs human help, the second relies on typing assumptions and assumption on the time window to bound the search space.

2 Preliminaries

Let $\mathcal{X}$ be a countable set of variables and let $\mathcal{F}^i$ be a countable set of function symbols of arity i, for every $i \in \mathbb{N}$. Let $\mathcal{F} = \bigcup_{i\in\mathbb{N}} \mathcal{F}^i$. The set of *terms over $\mathcal{X}$ and $\mathcal{F}$*, is denoted by $\mathcal{T}(\mathcal{X},\mathcal{F})$. We denote by $\leq$ the *subterm* relation on $\mathcal{T}(\mathcal{X},\mathcal{F})$. As usual, function symbols of arity 0 are called constant symbols. *Ground terms* are terms with no variables. We denote by $\mathcal{T}(\mathcal{F})$ the set of ground terms over $\mathcal{F}$. For any $t_1, t_2 \in \mathcal{T}(\mathcal{X},\mathcal{F})$, 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 $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{X},\mathcal{F})$ and term $t \in \mathcal{T}(\mathcal{X},\mathcal{F})$, we denote by $dom(\sigma)$ the domain of σ and 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 $\Gamma(\{x\})$. Given two substitutions σ and ρ with disjoint domains, $\sigma \oplus \rho$ is the substitution equal to σ on $dom(\sigma)$, equal to ρ on $dom(\rho)$, and undefined elsewhere.

A tree tr is a function from a non-empty finite subset $dom(tr)$ of ω^* to $\mathcal{X} \cup \mathcal{F}$ such that $tr(u) \in \mathcal{F}^n$ iff $u \cdot j \in dom(tr)$, for every $j \in \{1, \cdots, n\}$ and $u \cdot j \notin dom(tr)$ for every $j > n$, also $tr(u) \in \mathcal{X}$ implies $u \cdot j \notin dom(tr)$ for every $j \in \mathbb{N}$.

Henceforth, we tacitly identify the term t with $Tr(t)$. The elements of $dom(t)$ are called *positions* in t. The set of positions is denoted by $\mathcal{P}os$. 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 and w_2 is a prefix of w_1, 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 adopted in this paper. We assume Dolev-Yao's intruder model except that, since we are dealing with timed protocols, we add rules that:

1. allow the derivation of any time stamp and
2. allow the derivation of any short-term key k after some delay Δ_k since derivation of any message encrypted with k.

In addition to the usual terms considered in Dolev-Yao model, we add:

1. Clocks, i.e. variables that range over the underlying time model. We denote the set of clocks by $\mathcal{C}$.
2. Timestamps, that is values in the time domain.
3. Time variables, that is variables that range over the time domain. We denote by $\mathcal{Y}$ the set of time variables.

It is important to understand the difference between these three disjoint sets of variables: a time stamp is just a constant; clocks and time variables are variables. The difference is that the value of a clock advances with rate one with time while the value of a time variable does not. A time variable is simply a variable that ranges over the time domain.We fix the time domain to be the set of non-negative real numbers. Our results, however, hold also when we consider the natural numbers instead.

We consider two disjoint non-empty sets of keys: $\mathcal{KS}$ the set of short keys and $\mathcal{K}$ the set of any other keys. Moreover, we have the following sets of constant symbols: $\mathcal{P}$ for principal names and $\mathcal{N}$ for nonces. Let $\mathcal{X}$ denote the set of variables that range over terms. Let $\mathcal{A} = \mathcal{P} \cup \mathcal{N} \cup \mathcal{KS} \cup \mathcal{K} \cup \mathbb{R}_{\geq 0}$ and $\mathcal{F} = \mathcal{A} \cup \{\mathbf{encr}, \mathbf{pair}\}$. We consider terms build from constant symbols in $\mathcal{A}$, clocks in $\mathcal{C}$ and time variables in $\mathcal{Y}$ using the function symbols in $\mathcal{F}$. As usual, we write (m_1, m_2) for $\mathbf{pair}(m_1, m_2)$ and $\{m\}_k$ instead of $\mathbf{encr}(m, k)$. A *Clock-free term* is a term in which no clock appears; time variables and time stamps may appear in a clock-free term. We denote the set of clock-free terms by $\mathcal{T}(\mathcal{X} \cup \mathcal{Y}, \mathcal{F})$. *Messages* are ground terms in $\mathcal{T}(\mathcal{X} \cup \mathcal{Y}, \mathcal{F})$, we denote by $\mathcal{M} = \mathcal{T}(\mathcal{F})$ the set of messages. For conciseness, we write $\mathcal{T}$ instead of $\mathcal{T}(\mathcal{X} \cup \mathcal{Y}, \mathcal{F})$ and $\mathcal{T}_c$ instead of $\mathcal{T}(\mathcal{X} \cup \mathcal{Y} \cup \mathcal{C}, \mathcal{F})$.

For the time being, we will use the usual model of Dolev and Yao [15] augmented with the axiom:

(Time stamp) If $r \in \mathbb{R}_{\geq 0}$ then $E \vdash r$.

The axiom (Time stamp) represents the fact that the intruder can guess every possible time-stamp, i.e. time value.

For self-containedness, we briefly recall the derivation rules of the Doles-Yao model:

$$\frac{m \in E}{E \vdash m} \text{ (axiom)} \quad \frac{E \vdash (m_1, m_2)}{E \vdash m_i}, i = 1, 2 \quad \frac{E \vdash \{m\}_k,\ E \vdash k^{-1}}{E \vdash m} \quad \text{(decomposition rules)}$$

$$\frac{E \vdash m_1, E \vdash m_2}{E \vdash (m_1, m_2)} \quad \frac{E \vdash m,\ E \vdash k}{E \vdash \{m\}_k} \quad \text{(composition rules)}$$

As usual, we write $E \vdash m$, when m is derivable from E using the augmented Dolev-Yao model. A derivation of a message that does not use decomposition rules is denoted by $E \vdash_c m$. For a term t, we use the notation $E \vdash t$ to denote that there exists a substitution $\sigma : \mathcal{X} \to \mathcal{M}$ such that $E \vdash t\sigma$. For a set of messages M, we use the notation $E \vdash M$ to denote $E \vdash m$ for each m in M and $E \not\vdash M$ to denote $E \not\vdash m$ for each m in M.

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 $t(q) =$ **encr**; otherwise it is called *critical*. That is, encryption key positions are non-critical.

3.1 Process model

Timed cryptographic protocols are build from timed actions. Here, we consider two types of actions: message input and message output. A time constraint is associated to an action and describes when the action is possible.

Definition 3.1 (time constraints) *Time constraints are defined by:*

$$g ::= \top \mid \sum_{i=1}^{n} a_i c_i + \sum_{j=1}^{m} b_j T_j \bowtie d \mid g_1 \wedge g_2 \mid g_1 \vee g_2$$

where $m, n \in \mathbb{N}$, $c_i \in \mathcal{C}$ are clocks, $T_j \in \mathcal{Y}$ are time variables , $a_i, b_j \in \mathbb{Z}$, $d \in \mathbb{Z}$, and $\bowtie \in \{<, \leq\}$. The set of time constraints is denoted by $\mathcal{TC}$.

A time constraint is interpreted with respect to a valuation ν defined over a finite set of clocks $\{c_1, \ldots, c_n\}$ that associates values in the time domain to clocks, and a substitution σ that assigns ground clock-free terms to variables. The interpretation of a time constraint is given by:

- $[\![\top]\!]_{\nu,\sigma} = 1$, for any ν and σ.
- $[\![\sum_{i=1}^{n} a_i c_i + \sum_{j=1}^{m} b_j T_j \bowtie d]\!]_{\nu,\sigma} = 1$ iff $\sum_{i=1}^{n} a_i \nu(c_i) + \sum_{j=1}^{m} b_j \sigma(T_j) \bowtie d$;
- $[\![g_1 \wedge g_2]\!]_{\nu,\sigma} = 1$ iff $[\![g_1]\!]_{\nu,\sigma} = [\![g_2]\!]_{\nu,\sigma} = 1$;
- $[\![g_1 \vee g_2]\!]_{\nu,\sigma} = 1$ iff $[\![g_1]\!]_{\nu,\sigma} = 1$ or $[\![g_2]\!]_{\nu,\sigma} = 1$

Then (ν, σ) is said to be a *model* for a time constraint g, if $[\![g]\!]_{\nu,\sigma} = 1$.

Given a time constraint g and a set $\mathcal{R}$ of clocks,we denote by $g[\mathcal{R}]$ the time constraint obtained by substituting 0 for all clocks in $\mathcal{R}$. We also use the notation $g + d$ to denote the time constraint obtained from g by substituting each clock c in g by $c + d$.

Definition 3.2 (actions and protocols) *We consider input and output actions:*

- **An input** *action is of the form* $l \stackrel{g,\mathcal{R},?t(\tilde{x})}{\longrightarrow} l'$*, where*
  - $g \in \mathcal{TC}$ *is a time constraint called the guard,*
  - $t \in \mathcal{T}$ *is a clock free term and* $\tilde{x} \subseteq \mathcal{X} \cup \mathcal{Y}$ *is the set of variables instantiated by the input action.*

– *$\mathcal{R} \subseteq \mathcal{C}$ is a subset of clocks*
– *l, l' are labels.*

- **An output** *action is of the form $l \stackrel{g,\mathcal{R},!t_c}{\longrightarrow} l'$ where g, l, l' and $\mathcal{R}$ are as above and $t_c \in \mathcal{T}_c$ is a clock dependent term.*

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 $\alpha_j^i = \ell_j^i \stackrel{\beta_j^i}{\to} \ell_{j+1}^i$ 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 interleavings of a fixed set of sessions put in parallel $\Pi = \sum_{i=1}^{n} \ell_0^i \beta_0^i \cdots \ell_{n_i}^i \beta_{n_i}^i \ell_{n_i+1}^i$, where l_i^j are obtained combining the labels of each session into a single label and the variables of the protocol actions are indexed by the session identifier in order to distinct the same variable in different sessions.

For simplicity, we assume that each variable x occurs exactly once bind in an input action, that is in $\tilde{x}$ in action $?t(\tilde{x})$, and moreover, this occurence precedes any other occurence of x in any action. It is not difficult to see that this restriction can be easily handled.

Let $\mathcal{R} \subseteq \mathcal{C}$ be a subset of clocks, $\delta \in \mathbb{R}_{\geq 0}$ a constant, $\nu : \mathcal{C} \longrightarrow \mathbb{R}_{\geq 0}$ a valuation for clocks, and let $t_c \in \mathcal{T}_c$ be a clock dependent term. We denote by $\nu[\mathcal{R}]$ the valuation obtained from ν by resetting all clocks in $\mathcal{R}$, i.e. $\nu[\mathcal{R}](c) = 0$ for any $c \in \mathcal{R}$ and $\nu[\mathcal{R}](c) = \nu(c)$ for any $c \notin \mathcal{R}$; $\nu + \delta$ denotes the valuation which advances all clocks by the same delay δ, i.e. $(\nu + \delta)(c) = \nu(c) + \delta$; and $t_c[\nu]$ is the term obtained from t_c by replacing all occurrences of c by the value of $\nu(c)$.

Definition 3.3 (operational semantics) *A configuration of a protocol run is given by a tuple (σ, E, ν, ℓ) consisting of a substitution σ, a set of messages E, a valuation of clocks ν and a control point ℓ. The operational semantics is defined as a labelled transitional system over the set of configurations $\mathcal{C}onf$. The transition relation*

$$(\sigma, E, \nu, \ell) \stackrel{\alpha}{\longrightarrow} (\sigma', E', \nu', \ell')$$

is defined as follows:

- **output:** $\alpha = \ell \stackrel{g,\mathcal{R},!t}{\longrightarrow} \ell'$. *Then, we have*

  $$(\sigma, E, \nu, \ell) \stackrel{\alpha}{\longrightarrow} (\sigma, E', \nu', \ell')$$

  if $[\![g]\!]_{\nu,\sigma} = 1$, $E' = E \cup \{t(\sigma \oplus \nu[\mathcal{R}])\}$ and $\nu' = \nu[\mathcal{R}]$.

  That is, sending the message t (provided that guard g is satisfied by the actual configuration) amounts to reset clocks in $\mathcal{R}$ and adding t evaluated with respect to the substitution σ and the valuation of clocks $\nu[\mathcal{R}]$, to the knowledge of the intruder

- **input:** $\alpha = \ell \stackrel{g,\mathcal{R},?t(\tilde{x})}{\longrightarrow} \ell'$.

  Then, we have

  $$(\sigma, E, \nu, \ell) \stackrel{\alpha}{\longrightarrow} (\sigma', E, \nu', \ell')$$

  if there is $\rho \in \Gamma(\tilde{x})$ with $E \vdash t(\sigma \oplus \rho)$, $[\![g]\!]_{\nu,\sigma\oplus\rho} = 1$, $\sigma' = \sigma \oplus \rho$, and $\nu' = \nu[\mathcal{R}]$.

  That is, $?t$ corresponds to receiving any message, known to the intruder, that matches with $?t\sigma$ by a substitution ρ, such that g is satisfied by the pair ν, $\sigma \oplus \rho$; in addition, this action resets clocks in $\mathcal{R}$.

- **time passing:** $(\sigma, E, \nu, \ell) \stackrel{\delta}{\longrightarrow} (\sigma, E, \nu + \delta, \ell)$, *for any $\delta \in \mathbb{R}_{\geq 0}$. This action represents the passage of δ time units; passage of an arbitrary time is denoted by $\stackrel{\tau}{\longrightarrow} = \bigcup_{\delta \in \mathbb{R}_{\geq 0}} \stackrel{\delta}{\longrightarrow}$.*

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, a valuation ν_0 and a control point $\ell_0 \in \{\ell_0^1, \cdots, \ell_{n_i}^n\}$.

Example 3.1 *The Denning-Sacco shared key protocol [9], a protocol for distribution of a shared symmetric key by a trusted server and mutual authentication. Here, the timestamps are used to ensure the freshness of the shared key. Using the usual notation for cryptographic protocols, the protocol is described as follows:*

$$\begin{array}{lr} A \rightarrow S: & A, B \\ S \rightarrow A: & \{B, Kab, T, \{Kab, A, T\}_{Kbs}\}_{Kas} \\ A \rightarrow B: & \{Kab, A, T\}_{Kbs} \end{array}$$

The keys Kas and Kbs are shared keys between the participant A respectively B and the server S. The goal of the Denning-Sacco shared key protocol is to allow two principals A and B to obtain a secret symmetric key from a trusted server S.

The next table shows how we describe the protocol. The constant parameters δ_1, δ_2 represent network delays for A respectively B. We use a special clock now which is a global clock that is never reset and has an arbitrary initial value. For convenience of notation, we write $\ell\ \alpha\ \ell'$ instead of $\ell \xrightarrow{\alpha} \ell'$ and we omit the guard when it is the constant $\top$ and the set of clocks to be reset when it is empty.

A:

$0\ !(A, B)\ 1$
$1\ now - T_1 < \delta_1,$
$\quad ?\{B, x, T_1, y\}_{smk(A,S)}\ 2$
$2\ !y\ 3$

S:

$0\ ?(z, v)\ 1$
$1\ !\{v, K, now, \{K, z, now\}_{smk(v,S)}\}_{smk(z,S)}\ 2$

$B:$

$0\ now - T_2 < \delta_2,\ ?\{u, p, T_2\}_{smk(B,S)},\ 1$

Each participant of the protocol may be seen as a sequential process. First, the participant A sends his identity A and the identity of B to the server. Then, A receives back the message $\{B, x, T_1, y\}_{smk(A,S)}$. If T_1, is "timely", i.e. the difference between the current time and the value of T_1 is less than the constant parameter δ_1 then A accepts x as session key and forwards the message y to B. On the other side, B, when receives the message $\{u, p, T_2\}_{smk(B,S)}$, it checks if T_2 is "timely" and, if so, it accepts p as session key. The server S, every time when it receives a pair of two participants (z, v) it generates a new session key K and sends it together with its current time now in a message of the form $\{v, K, now, \{K, z, now\}_{smk(v,S)}\}_{smk(z,S)}$ to the first participant of the pair z.

3.2 Short term keys

Intuitively, to each short term key $k \in \mathcal{KS}$ we associate a constant Δ_k and a clock $c(k)$ that can be activated when the intruder deduces a message of the form $\{x\}_k$. Then, when the value of the clock $c(k)$ reaches Δ_k, the key k becomes deducible by the intruder. That is, a short term key k is "cracked" Δ_k time units after a message $\{x\}_k$ becomes known.

To take into account short term keys, instead of directly extending the intruder's model, we consider parallel composition of the protocol to be verified with the following process

$$\bigoplus_{k \in \mathcal{KS}} \alpha_0^k; \alpha_1^k, \text{ where}$$

$\alpha_0^k \equiv \ell_0 \xrightarrow{\{c(k)\}, ?\{x\}_k} \ell_1$ and $\alpha_1^k \equiv \ell_1 \xrightarrow{c(k) \geq \Delta_k, !k} \ell_2$, where $\bigoplus$ is used to denote parallel composition. That is, the action α_0^k resets the clock $c(k)$, when it is possible to perform the input $?\{x\}_k$, i.e., when the intruder can deduce a message of the form $\{x\}_k$, and the action α_1^k reveals the key k when the value of $c(k)$ exceeds Δ_k.

4 The TSPL logic

In this section, we introduce the constraints/formulae we use to describe security properties. The logic we introduce allows to describe secrecy, authentication and any safety property.

Henceforth, let $K \subseteq \mathcal{K}$ be a fixed but arbitrary set of keys, such that $\emptyset \neq K \neq \mathcal{K}$. This set of keys can be understood as the set of "good" keys, whose inverses are not known by the intruder.

A major problem we face for developing a complete inference system for cryptographic protocols is 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.

4.1 Term transducers and the main modality of the logic

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 a 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 it will become clear later, a run of a CP provides the intruder with new term transducer she (he) can apply to learn new terms.

The main modality of the logic we use can be defined as follows:

Definition 4.1 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
- $m = \textbf{pair}(m_1, m_2)$, $m \neq s$ and both $m_1\langle w\rangle s$ and $m_2\langle w\rangle s$ are true, or
- $m = \textbf{encr}(m_1, k)$, $m \neq s$, $k \notin K$ and $m_1\langle w\rangle s$ is true, or
- $m = \textbf{encr}(m_1, k)$, $m \neq s$, $k \in K$ and $w = \epsilon$, or
- $m = \textbf{encr}(m_1, k)$, $w = (b, r).w_1$, $m \neq s$, $k \in K$, and $m \neq b$ or $m|_r\langle w_1\rangle s$ is true.

In other words, s is w-protected in m means s can not be obtained from m by means of decomposition or w-transducer use.

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 4.1 *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.

4.1.1 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, for any message m in the set, all messages along at least one path of the tree representing the message m. The same idea is used in e.g. [24, 29, 14].

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 4.2 (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 4.2 *Consider the message $m = (\{A, N\}_k, B)$. Then $wc(m)$ consists of the following sets:*

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

We can prove the following:

Lemma 4.1 *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 4.3
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 4.4 *Let $E = \{\{s\}_k, k\}$ be a set of messages and let $b = \{\{s\}_k\}_k$. We have $E\langle(b, 11)\rangle s$. But we have both $E \vdash b$ and $\neg b\langle(b, 11)\rangle s$.*

This example shows that $E\langle w\rangle S$ is not stable under intruder composition rules. To remedy to this we only consider *well-formed* formulae. To define well-formedness, we need the following:

Let t be a term and p a critical position in t. Then, we denote by $NP(t, p)$ recursively on the structure of t as follows:

- if t is a constant or a variable, or $p = \epsilon$, then $NP(t, p)$ is undefined.
- if $t = (t_1, t_2)$ and $p = 1 \cdot p'$ then $NP(t, p) = 1 \cdot NP(t_1, p')$. Similarly, when $p = 2 \cdot p'$.
- if $t = \{t'\}_k$ and $k \in K$ and $p \neq \epsilon$ then $NP(t, p) = \epsilon$.
- if $t = \{t'\}_k$ and $k \notin K$ and $p \neq \epsilon$ then $NP(t, p) = 1 \cdot NP(t', 1^{-1}p)$.

Example 4.5 *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 $\mathrm{NP}(t, p) = 1$, which corresponds to the key k_2; $\mathrm{NP}(t, p')$ is, however, undefined.*

Definition 4.3 *Let (b, p) be a term transducer. Then* the next term transducer in b from above that dominates p *(denoted by $\mathrm{NT}(b, p)$) is defined as follows:*

$$\mathrm{NT}(b, p) = \begin{cases} (b|_{1q}, (1q)^{-1}p) & \textit{if } \mathrm{NP}(b|_1, 1^{-1}p) = q \\ undefined & \textit{otherwise} \end{cases}$$

We illustrate this definition by the following example.

Example 4.6 *Let b be the term $\{(\{N\}_{k'}, A)\}_k$ with $k, k' \in K$. Then, we have* $\mathrm{NT}(b, 111) = (\{N\}_{k'}, 1)$*. On the other hand,* $\mathrm{NT}(b, 12)$ *and* $\mathrm{NT}(b, 11)$ *is not defined.*

We have now everything we need to express the conditions that guarantee stability under the intruder's derivations. i.e, the well-formedness condition:

Definition 4.4 *$(w_i, S_i)_{i \in I}$ is called* well-formed*, if the following conditions are satisfied for every $i \in I$:*

1. *S_i is closed,*
2. *if $w_i = (b, r).w$, then the following conditions are satisfied:*
   a. *there exists $j \in I$ such that $w_j = w$ and $S_i \subseteq S_j$,*
   b. *if there exists a term transducer $(b_1, r_1) = \mathrm{NT}(b, r)$, then there exists $k \in I$ such that either $b \in S_k$ or $w_k = (b_1, r_1).w$ and $S_i \subseteq S_k$.*

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

Proposition 4.2 *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: See Appendix B.2. ■

4.2 TSPL: A Logic for Security Properties

In order to express general secrecy properties that involve variables, we introduce a new set of function symbols $\mathcal{B}$. Extended terms are build as before except that now we allow function symbols in $\mathcal{B}$ to occur applied to variables, which we denote by $x.f$. We denote by $\mathcal{BX}$ the set $\{x.f \mid f \in \mathcal{B}, x \in \mathcal{X}\}$. Given a substitution σ that associates a message m to x, it will associate a set in $wc(m)$ to $x.f$.

The syntax of TSPL is defined in Table 4.2, where X is a fixed second-order variable that ranges over sets of messages and f is a meta-variable that ranges over $\mathcal{B}$. x is a meta-variable that ranges over the set $\mathcal{X}$ of first-order variables. First-order variables range over messages; t is a meta-variable over terms. Moreover, S is a finite set of extended 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 $\mathcal{C}onf_0 = \{(\sigma, E, \nu, l) \mid (\sigma, E, \nu, l) \in \mathcal{C}onf, \mathcal{K} \setminus K^{-1} \subseteq E\}$.

$\Psi ::=$	$\top \mid \sum_{i=1}^{n} a_i c_i + \sum_{j=1}^{m} b_j T_j \bowtie \delta \mid \Psi_1 \wedge \Psi_2 \mid \Psi_1 \vee \Psi_2$	time constraints
$\Phi ::=$	$X\langle w\rangle S \mid x\langle w\rangle S \mid t\langle\epsilon\rangle x.f \mid x = t \mid pc = \ell \mid$	
	$\exists x \Phi \mid \forall f \Phi \mid \top \mid \Phi \wedge \Phi \mid \neg\Phi$	term formulae
$\Gamma ::=$	$\Phi \mid \Psi \mid \Gamma \vee \Gamma \mid \Gamma \wedge \Gamma$	TSPL formulae

Table 1: The set of formulae TSPL

Notice that omitting the negation for time constraints is not essential as any negation of a time constraint can be put in a positive form.

Definition 4.5 (semantics) *The interpretation of a formula is given by the set of its models, i.e., the set of configurations from $\mathcal{C}onf_0$ that satisfy the formula. The semantics of* TSPL *is defined by the following clauses:*

- $[\![\Psi]\!] = \{(\sigma, E, \nu, \ell) \mid [\![\Psi]\!]_{\nu,\sigma} = 1\}$
- $[\![X\langle w\rangle S]\!] = \{(\sigma, E, \nu, \ell) \mid E\langle w\sigma\rangle S\sigma\}$
- $[\![x\langle w\rangle S]\!] = \{(\sigma, E, \nu, \ell) \mid \{x\sigma\}\langle w\sigma\rangle S\sigma\}$
- $[\![x = t]\!] = \{(\sigma, E, \nu, \ell) \mid x\sigma = t\sigma\}$
- $[\![pc = \ell]\!] = \{(\sigma, E, \nu, \ell') \mid \ell' = \ell\}$
- $[\![t\langle\epsilon\rangle x.f]\!] = \{(\sigma, E, \nu, \ell) \mid \{t\sigma\}\langle\epsilon\rangle(x.f)\sigma\}$
- $[\![\forall f\Phi]\!] = \bigcap_{f_0 \in \mathcal{B}} \{(\sigma, E, \nu, \ell) \mid (\sigma \oplus [f \mapsto f_0], E, \nu, \ell) \in [\![\Phi]\!]\}$
- $[\![\exists\Phi]\!] = \bigcup_{x_0 \in \mathcal{M}} \{(\sigma, E, \nu, \ell) \mid (\sigma \oplus [x \mapsto x_0], E, \nu, \ell) \in [\![\Phi]\!]\}$
- $[\![\neg\varphi]\!] = \mathcal{C}onf_0 \setminus [\![\varphi]\!]$
- $[\![\top]\!] = \mathcal{C}onf_0$
- $[\![\Gamma_1 \wedge \Gamma_2]\!] = [\![\Gamma_1]\!] \cap [\![\Gamma_2]\!]$
- $[\![\Gamma_1 \vee \Gamma_2]\!] = [\![\Gamma_1]\!] \cup [\![\Gamma_2]\!]$.

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

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

The semantics of the newly introduced formulae is:

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

We can prove that any formula of the form $t\langle w\rangle S$ is definable in TSPL, i.e. that there is a TSPL formula $\mathcal{J}(t, w, s)$ such that $t\langle w\rangle s \equiv \mathcal{J}(t, w, s)$ (see Appendix B.3).

Notations We use the notations $\perp$ for $\neg\top$, $(\sigma, E, \nu, \ell) \models \varphi$ for $(\sigma, E, \nu, \ell) \in [\![\varphi]\!]$, $t\langle \not w\rangle S$ for $\neg t\langle w\rangle S$, $X\langle \not w\rangle S$ for $\neg X\langle w\rangle S$. Also, given a term s, 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 TSPL logic using set inclusion).

Well-formed formulae. We extend now the notion of closure of sets of messages to sets of extended terms. The definition is similar except that we have to consider two new cases: 1.) the case of a term t of the form $x.f$: $\mathrm{wc}_t(t) = t \odot \{K^{-1}\}$ and 2.) the case of a variable x: $\mathrm{wc}_t(x) = x.f \odot \{K^{-1}\}$, where f is a fresh function symbol.

The Definition 4.4, that defines when $(w_i, S_i)_{i\in I}$ is well-formed for S_i sets of messages, is now easily extended to sets of extended terms. As now we are dealing with formulae, we have to define when a formula is well-formed in the same sense.

Definition 4.6 *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 a parallel to Proposition 4.1 and given by the following corollary, which is a direct consequence of Definitions 4.4 and 4.6 and Proposition 4.1.

Corollary 4.1 *Let Φ be a well-formed formula such that $\Phi \Rightarrow X\langle w\rangle S$ and let $(\sigma, E, \nu, l) \in [\![\Phi]\!]$. If m is a message such that $E \vdash m$, then $m\langle w\sigma\rangle S\sigma$.*

Now, the property of Corollary 4.1 turns out to be crucial for developing a complete symbolic method 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 $\mathcal{H}(X\langle b.w\rangle S)$, such that $\mathcal{H}(X\langle b.w\rangle S) \Rightarrow X\langle b.w\rangle S$. The intuition follows from the Definition 4.4: in the case that the term transducer $b = (t, p)$ contains an inner term transducer b_1, either t cannot be built or it doesn't help; moreover, the formula $\mathcal{H}(X\langle b.w\rangle S)$ is closed with respect to suffixes of w.

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

Proposition 4.3 *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 4.4 and 4.6. ■

Given a term t, let $F(t)$ denote the formula $\forall \overrightarrow{f} \bigwedge_{S' \in wc_t(t)} X\langle \not{f}\rangle S'$ where $\overrightarrow{f}$ is the set of all fresh variables $f \in \mathcal{BX}$ that occur in $wc_t(t)$. The intuitive explanation of the usefulness of $F(t)$ is the following: being in a state (σ, E, ν, 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 $(\sigma, E, \nu, l) \in [\![F(t\rho)]\!]$. To give an idee of how secrecy and authentication can be expressed in TSPL we present an example in Appendix A.

5 Computing Predecessors

We are interested in proving reachability properties of bounded timed cryptographic protocols. Given a property φ and an action α, $pre(\alpha, \mathcal{C})$ denotes the smallest set of configurations such that by executing α may lead to a configuration that satisfies φ. That is,

Definition 5.1 (predecessors) *The predecessor of a set of configurations $\mathcal{C} \subseteq \mathcal{C}onf$ with respect to an action α, denoted $pre(\alpha, \mathcal{C})$ is the set of configurations s, such that there is at least one possible execution of α that leads from s to a configuration in $\mathcal{C}$. More precisely*

$$pre(\alpha, \mathcal{C}) ::= \left\{(\sigma, E, \nu, l) \mid \exists(\sigma', E', \nu', l') \in \mathcal{C} \text{ s. t. } (\sigma, E, \nu, l) \xrightarrow{\alpha} (\sigma', E', \nu', l')\right\}.$$

Given a formula Φ, we use $pre(\alpha, \Phi)$ instead of $pre(\alpha, [\![\Phi]\!])$ to denote the predecessor of a formula $\Phi \in$ TSPL.

The purpose of this section is to show that $pre(\alpha, \Phi)$ is effectively expressible in TSPL, when Φ is a positive boolean combination of time constraints and term formulae of the form:

$$x = t \mid pc = \ell \mid \top \mid \bot \mid x \neq t \mid pc \neq \ell \mid X\langle \not{w}\rangle S \mid x\langle \not{w}\rangle S \mid t\langle \not{e}\rangle x.f \mid \forall f \Phi \mid \exists x \Phi.$$

First, it is easy to see that $pre(\alpha, \Phi) = \Phi$, if α is a time passing action and Φ is a term formula. Also, for any action $\alpha = l \xrightarrow{g,\mathcal{R},!t} l'$, respectively $\alpha = l \xrightarrow{g,\mathcal{R},?t} l'$, and any time constraint Ψ, we have $pre(\alpha, \Psi) = g \wedge pc = \ell \wedge \Psi[R] \wedge F(t)$.

Notice that pre distributes with respect to disjunction (finite or infinite). Moreover, it distributes over conjunction, finite and infinite, for discrete action (input or output). The main reason is that the only non-deterministic discrete action is input which gives raise to external non-determinism.

5.1 Time passing and time constraints

In this section, we show that the predecessor of $[\![\Psi]\!]$, where Ψ is a time constraint, can be described by a TSPL formula. We consider the action $\xrightarrow{\tau}$, i.e. time passing. The case of input and output actions is described above.

We need first to define three kinds of normal forms for time constraints. Let Ψ be the atomic time constraint $\sum_{i=1}^{n} a_i c_i + \sum_{j=1}^{m} b_j T_j \bowtie d$. We denote by $\mathscr{C}(\Psi)$ the sum of the coefficients of clocks, i.e. $\sum_{i=1}^{n} a_i$. Then, an atomic time constraint $\Psi \equiv \sum_{i=1}^{n} a_i c_i + \sum_{j=1}^{m} b_j T_j \bowtie d$ is in *positive normal form* (PNF for short), if $\mathscr{C}(\Psi) > 0$; it is in *negative normal form* (NNF for short), if $\mathscr{C}(\Psi) < 0$; and finally, it is in *0-normal form*, if $\mathscr{C}(\Psi) = 0$.

Clearly any time constraint can be put in the form of a disjunction of conjunctions of the form $\Psi_1 \wedge \Psi_2 \wedge \Psi_3$, where Ψ_1 is a conjunction of formulae in PNF, Ψ_2 is a conjunction of formulae in NNF and Ψ_3 is a conjunction of formulae in 0-NF. For the rest of this section, we write $\psi \in \Psi_i$ to state that ψ is a conjunct of Ψ_i, i.e., we view conjunctions of formulae as sets of formulae.

Thus, let us consider a time constraint of the form $\Psi_1 \wedge \Psi_2 \wedge \Psi_3$ as above. Then, $pre(\xrightarrow{\tau}, \Psi_1 \wedge \Psi_2 \wedge \Psi_3)$ can be described by the formula $\exists\delta \geq 0 \cdot \Psi_1 + \delta \wedge \Psi_2 + \delta \wedge \Psi_3 + \delta$. We have then to show that we can eliminate the quantification on δ while obtaining a time constraint.

First, notice that $\Psi_3 + \delta$ is logically equivalent to Ψ_3, since it is in 0-NF. Therefore, we can rewrite the formula to the equivalent formula $\exists\delta \geq 0 \cdot (\Psi_1 + \delta \wedge \Psi_2 + \delta) \wedge \Psi_3$ and focus on discussing how to transform $\exists\delta \geq 0 \cdot (\Psi_1 + \delta \wedge \Psi_2 + \delta)$ into an equivalent time constraint. Let us explain the main idea by considering the simple case where Ψ_1 and Ψ_2 are atomic time constraints.

The simple case Consider a PNF constraint $\Psi_1 \equiv \sum_{i=1}^{n} a_i c_i + \sum_{j=1}^{m} b_j T_j \bowtie_1 d$ and a NNF one $\Psi_2 \equiv \sum_{i=1}^{n} a'_i c_i + \sum_{j=1}^{m} b'_j T_j \bowtie_2 d'$. Then, we have:

$$\begin{array}{l} \Psi_1 + \delta \equiv \sum_{i=1}^{n} a_i c_i + \sum_{j=1}^{m} b_j T_j + \delta \sum_{i=1}^{n} a_i \bowtie_1 d \\ \Psi_2 + \delta \equiv \sum_{i=1}^{n} a'_i c_i + \sum_{j=1}^{m} b'_j T_j + \delta \sum_{i=1}^{n} a'_i \bowtie_2 d' \end{array}$$

By multiplying with $\mathscr{C}(\Psi_1)$ and $|\mathscr{C}(\Psi_2)|$ we have:

$$\begin{array}{l} \Psi_1 + \delta \equiv \sum_{i=1}^{n} |\mathscr{C}(\Psi_2)| a_i c_i + \sum_{j=1}^{m} |\mathscr{C}(\Psi_2)| b_j T_j + \delta |\mathscr{C}(\Psi_2)| \sum_{i=1}^{n} a_i \bowtie_1 |\mathscr{C}(\Psi_2)| d \\ \Psi_2 + \delta \equiv \sum_{i=1}^{n} \mathscr{C}(\Psi_1) a'_i c_i + \mathscr{C}(\Psi_1) \sum_{j=1}^{m} b'_j T_j + \delta \mathscr{C}(\Psi_1) \sum_{i=1}^{n} a'_i \bowtie_2 \mathscr{C}(\Psi_1) d' \end{array}$$

Adding the right-hands of the equivalences yields the time constraint:

$$\sum_{i=1}^{n} a''_i c_i + \sum_{j=1}^{m} b''_j T_j \bowtie' |\mathscr{C}(\Psi_2)| d + \mathscr{C}(\Psi_1) d'$$

with $a''_i = |\mathscr{C}(\Psi_2)| a_i + \mathscr{C}(\Psi_1) a'_i$, $b''_j = |\mathscr{C}(\Psi_2)| b_i + \mathscr{C}(\Psi_1) b'_i$ and if $\bowtie_1 \equiv \bowtie_2$ then $\bowtie' \equiv \bowtie_1$ else $\bowtie' \equiv <$.

Let us denote this formula by $\Delta(\Psi_1, \Psi_2)$. Notice that $\Delta(\Psi_1, \Psi_2)$ is independent of δ. One can prove that $\exists\delta \geq 0 \cdot (\Psi_1 + \delta \wedge \Psi_2 + \delta)$ is equivalent to the time constraint $\Psi_1 \wedge \Delta(\Psi_1, \Psi_2)$. The conjunct Ψ_1 has to be kept as we are interesting in the predecessors, thus the upper bound on the clocks must be satisfied as time only increases.

The general case Let us now return to the general case, where Ψ_1 and Ψ_2 are arbitrary conjunctions of formulae in PNF, respectively, NNF. To handle this case we generalize Δ to sets (conjunctions of formulae as follows):

- $\Delta(\emptyset, \psi) = \top$.
- $\Delta(\psi, \emptyset) = \psi$.
- $\Delta(\Psi_1, \Psi_2) = \bigwedge_{\psi_1 \in \Psi_1, \psi_2 \in \Psi_2} \Delta(\psi_1, \psi_2)$

Then we can prove that $\exists\delta \geq 0 \cdot (\Psi_1 + \delta \wedge \Psi_2 + \delta)$ is equivalent to $\Delta(\Psi_1, \Psi_2)$.

Summarizing together, we can transform $\exists\delta \geq 0 \cdot \Psi_1 + \delta \wedge \Psi_2 + \delta \wedge \Psi_3 + \delta$ into the equivalent time constraint $\Delta(\Psi_1, \Psi_2) \wedge \Psi_1 \wedge \Psi_3$. Hence, if we define $\mathbf{Pre}(\xrightarrow{\tau}, \Psi_1 \wedge \Psi_2 \wedge \Psi_3) \stackrel{def}{=} \Delta(\Psi_1, \Psi_2) \wedge \Psi_1 \wedge \Psi_3$, we obtain the following result:

Proposition 5.1 *For any time constraint Ψ,*

$$pre(\xrightarrow{\tau}, [\![\Psi]\!]) = [\![\mathbf{Pre}(\xrightarrow{\tau}, \Psi)]\!].$$

5.2 Output action and atomic term formulae

Throughout this section let $\alpha = l \stackrel{g,\mathcal{R},!t}{\longrightarrow} l'$, and let $\tilde{c}$ be all the clocks that occur in t and do not occur in $\mathcal{R}$. We show that we can express $pre(\alpha, \varphi)$, for any atomic term formula φ. The core point here is how we deal with the clocks occurrences in the sent message. Since the values of clocks change with time, we have to freeze these values in the message added to the intruder knowledge; we do this by replacing in t, all occurrences of clocks that are not reset $\tilde{c}$ with fresh time variables $\tilde{T_c}$ and by introducing the constraints $\tilde{T_c} = \tilde{c}$. For more details, see *Example B.1*, presented in Appendix B.7.

Let us define $\mathbf{Pre}(\alpha, \varphi)$:

1. $\mathbf{Pre}(\alpha, \varphi) \stackrel{def}{=} g \wedge pc = \ell \wedge (\varphi \vee ((X, t[0/\mathcal{R}, \tilde{T_c}/\tilde{c}])\langle \not{w} \rangle S \wedge \tilde{T_c} = \tilde{c}))$, where $\tilde{T_c}$ are fresh time variables, if φ is a formula of the form $X\langle \not{w} \rangle S$ or $(X, x)\langle \not{w} \rangle S$.

2. $\mathbf{Pre}(\alpha, \varphi) \stackrel{def}{=} g \wedge pc = \ell \wedge \varphi$, if φ is of the form $x \neq t'$, $x = t'$, $\top$, $\bot$, $pc = \ell'$ or $t\langle \not{e} \rangle x.f$.

Then, we have the following :

Proposition 5.2 *For any output action α and atomic term formula φ,*

$$pre(\alpha, [\![\varphi]\!]) = [\![\mathbf{Pre}(\alpha, \varphi)]\!].$$

5.3 Input action and atomic term formulae

Throughout this section let $\alpha = l \stackrel{g,\mathcal{R},?t(\tilde{x})}{\longrightarrow} l'$. We show that we can express $pre(\alpha, \varphi)$, for any atomic term formula φ. To do so, we need to prove a few intermediate results.

The intuitive explanation of next lemma is the following: being in a state (σ, E, ν, 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 $(\sigma, E, \nu, l) \in [\![F(t\rho)]\!]$.

Lemma 5.1 *Let E be a set of terms, l be a label, ν be a clocks valuation and let ρ and σ be ground substitutions such that $dom(\rho) = \tilde{x}$ and $dom(\sigma) \cap \tilde{x} = \emptyset$. Then it holds $(\sigma, E, \nu, l) \in [\![F(t\rho)]\!]$ iff $E \vdash t(\sigma \oplus \rho)$.*

Proof: See Appendix B.4. ■

First, notice that the effect of an input action $?t$ depends on the messages that match with t and that are known by the intruder. Therefore, we need to characterize the set of configurations s, such that if in the next step x is instantiated by an input $?t(\tilde{x})$, the reached configuration s' satisfies $x\langle \not{w} \rangle S$.

To understand how this characterization is obtained, the best is to consider the negation of $x\langle \not{w} \rangle S$, i.e., $x\langle w \rangle S$. 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 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 5.2 *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*

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

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

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

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

Proof: See Appendix B.5. ■ Let now α be the action $\alpha = l \overset{g, \mathcal{R}, ?t(\tilde{x})}{\longrightarrow} l'$, where $\tilde{x}$ are the variables that are instantiated by this action. We then define $\mathbf{Pre}(\alpha, \varphi)$ as follows:

1. $\mathbf{Pre}(\alpha, \varphi) \overset{def}{=} g \wedge pc = \ell \wedge F(t) \wedge \varphi$, if φ is of the form $X\langle \not{w} \rangle S$, $t \langle \not{f} \rangle x.f$, $(X, y)\langle \not{w} \rangle S$, $x \neq t'$, $x = t'$, $pc = \ell'$, $\top$ or $\perp$ and $y \notin \tilde{x}$.

   The main point here is the conjunct $F(t)$ which ensures that the intruder can derive a message that maches with the input term.

2. $\mathbf{Pre}(\alpha, (X, x)\langle \not{w} \rangle S) \overset{def}{=} g \wedge pc = \ell \wedge F(t) \wedge \neg \mathcal{K}(t, x, w, S)$, if $x \in \tilde{x}$.

Proposition 5.3 *For any input action α and term formula φ,*

$$pre(\alpha(\tilde{x}), [\![\varphi]\!]) = [\![\exists \tilde{x} \cdot \mathbf{Pre}(\alpha, \varphi)]\!].$$

5.4 Collecting the results together

It is easy to see that for any formula $\varphi \in$ TSPL and any action α, $\mathbf{Pre}(\alpha, \varphi) \in$ TSPL. Then, we have the following theorem:

Theorem 5.1 *Let α be any action and φ any formula in* TSPL. *Then,*

$$pre(\alpha(\tilde{x}), [\![\varphi]\!]) = [\![\exists \tilde{x} \cdot \mathbf{Pre}(\alpha, \varphi)]\!].$$

6 Decidability of TSPL

In this section, we prove decidability of the existence of a model for existential TSPL formulae (that is, formulae of the form $\exists x_1 \ldots \exists x_m \exists T_1 \ldots \exists T_n \forall f_1 \ldots \forall f_p \phi$ with ϕ a quantifier free formula). Notice that since we showed in Section 5 that given a formula φ in TSPL and a bounded CP π, one can compute $\mathbf{Pre}(\pi, \varphi)$, decidability of the satisfiability of formulae yields a decision procedure for reachability of configurations described by TSPL formulae.

Second, we prove that the problem of deciding the existence of a model for an existential TSPL formula (shortly called TSPL-SAT) is NP-complete. Finally, we show that if we allow both existential and universal quantifiers for variables in $\mathcal{X}$, then the problem of deciding the existence of a model for a TSPL formula is undecidable.

6.1 A decidable fragment of TSPL

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. We do not consider formulae of the form $X\langle w \rangle s$ or $x\langle w \rangle s$ with s a variable; first, positive formulae appears only from initial conditions, and clearly, it does not make much sense to consider positive formulae with s a variable; second, such formulae add some technical difficulties that make harder the presentation of our results. On the other hand we shall add to TSPL two new kinds of formulae, $X, U \lhd \not\in \rhd x$ and $U \lhd \not\in \rhd x$ with $x \in \mathcal{X}$ and U a meta-variable that ranges over sets of terms, and which have the following semantics:
$[\![X, U \lhd \not\in \rhd x]\!] = \{(\sigma, E, \nu, \ell) \mid \forall A \in wc(\sigma(x))\ \{E\sigma \cup U\sigma\}\langle \not\in \rangle A\}$
$[\![U \lhd \not\in \rhd x]\!] = \{(\sigma, E, \nu, \ell) \mid \forall A \in wc(\sigma(x))\ \{U\sigma\}\langle \not\in \rangle A\}$.

Let $\Phi = \exists x_1 \ldots \exists x_m \exists T_1 \ldots \exists T_n \forall f_1 \ldots \forall f_p \phi$ where $\{x_i \mid i = 1 \ldots m\} \cup \{T_j \mid j = 1 \ldots n\} \cup \{f_k \mid k = 1 \ldots p\}$ is the set of all variables that appear in ϕ, and ϕ is a quantifier free formula builded using the connectives $\wedge$ and $\vee$ and the following literals:

$$X\langle w\rangle t \mid t\langle w\rangle t' \mid x = t \mid \top \mid X\langle \not w\rangle s \mid t\langle \not w\rangle t' \mid t\langle \not\epsilon\rangle x.f$$
$$x \neq t \mid \perp \mid \sum_{i=1}^{n} a_i c_i + \sum_{j=1}^{m} b_j T_j \bowtie d,$$

where X is a fixed second-order variable that ranges over sets of messages, x is a meta-variable that ranges over the set $\mathcal{X}$ of first-order variables, f is a meta-variable that ranges over $\mathcal{B}$, s ranges over extended terms, t, t' range over terms and w is a finite sequence of term transducers that can contain free variables.

To prove decidability for the satisfiability of TSPL formulae we follow a rule based approach (e.g., [20, 11] 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 that transfom any formula φ into a set of solved formulae, such that φ is satisfiable iff one of the formulae in solved form is satisfiable.

3. We prove soundness and completeness of these rules.

4. We also prove their termination for a given control, i.e. that normal forms are reached and that normal forms are indeed in intermediate form.

The reduction of a formula φ into a set of solved formulae is done in three phases.

1. We define a *preliminary form* and we introduce a set of rewriting rules to transform any formula in the fragment that interest us, into a preliminary form.

2. We define an *intermediate form* and we introduce a set of rewriting rules to transform any formula in preliminary form into an intermediate form.

3. For each formula in intermediate form, we show how to reduce its satisfiability to the satisfiability of a set of *saturated formulae* in intermediate form; moreover, for each saturated formula in intermediate form, we can extract a formula in solved form such that a model exists for the formula in intermediate form if and only if the extracted formula in solved form is satisfiable.

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 $[\![\varphi]\!] = \bigcup_{i=1}^{n} [\![\varphi_i]\!]$.

6.2 Solved form

A formula is called in *solved form* if it is syntactically equal to $\top$, $\perp$ or to a conjunction $\Psi \wedge \varphi$ where Ψ is a time constraint and φ is of the form:

$$\bigwedge_{i=1}^{n_1} X\langle\epsilon\rangle w_i \wedge \bigwedge_{i=1}^{n_2} X\langle\not\epsilon\rangle w_i' \wedge \bigwedge_{i=1}^{n} [\bigwedge_{j=1}^{m_i} x_i\langle\epsilon\rangle t_i^j \wedge \bigwedge_{j=1}^{l_i} x_i\langle\not\epsilon\rangle u_i^j \wedge \bigwedge_{j=1}^{o_i} x_i \neq v_i^j]$$

such that:

- For any $i = 1, \cdots, n$, $x_i \in \mathcal{X}$.

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

where by $var(t)$ we denote the set of variables that appear in the term t.

We now show how one can check whether a formula in solved form has a model. Let us first assume that the time constraint is $\top$, that is, we only have to deal with φ. We will later show how to reduce the general case to this one.

Satisfiability of φ So, let φ a conjunction as above. We consider clocks and time variables as constant symbols and 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 public 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

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

Now, let $E = \{w_1', \ldots, w_{n_2}', k\}$.

Theorem 6.1 *Let φ be a term formula in solved form syntactically different from $\top$ and $\bot$. Let σ be the substitution and E be the set of messages as defined above. Then, φ has a model iff (σ, E) 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, E) 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.

■

6.3 The general case

Let us now return to the case where Ψ is a conjunction of time constraints. It turns out that only the equalities between the variables in $\mathcal{C} \cup \mathcal{Y}$ that are implied by Ψ might rule out some of the models of φ. That is, we need only to take into account such equalities. Let us illustrate this by an example. Consider the formula $\varphi' \equiv x\langle\epsilon\rangle(A, c) \wedge x\langle\not\epsilon\rangle(A, T)$. Then, $\varphi' \wedge 0 \leq c \leq 1 \wedge 0 \leq T \leq 1$ is satisfiable; while $\varphi' \wedge c - T = 0$ is not. Indeed, in the first the time constraint does not imply any equality; while in the second case it implies $c = T$.

Therefore, we proceed as follows: We compute the strongest time constraint Ψ' of the form $\bot$ or

$$\bigwedge_{i=1}^{m} z_i = z_i'$$

where $z_i, z_i' \in \mathcal{C} \cup \mathcal{Y}$ and such that Ψ implies Ψ'. If Ψ' is $\bot$ then $\Psi \wedge \varphi$ is not satisfiable, and we are done.

Therefore, let us suppose that Ψ' is satisfiable. Then, Ψ' induces an equivalence relation on the variables in $\mathcal{C} \cup \mathcal{Y}$ as follows: $z \sim_{\Psi'} z'$ iff $z = z'$ is a consequence of Ψ'. Using this equivalence relation we can define an idempotent substitution $\sigma_{\Psi'}$ that associates to the members of an equivalence class a designated representant. Now, we apply the substitution $\sigma_{\Psi'}$ to φ and check that the obtained formula is satisfiable.

6.4 Rewriting rules

In this section, we present a set of rewriting rules that transfom any formula φ as considered in subsection 6.1, into a set of solved formulae, such that φ is satisfiable iff one the formulae in solved form is satisfiable.

For the rules of the form $\varphi \longrightarrow \psi$, where φ is an atomic formula, we tacitly assume a rule $\neg\varphi \longrightarrow \neg\psi$. Obvious rules (as distributivity of $\vee$ (respectively $\wedge$) with respect to $\wedge$ (respectively $\vee$)) are not mentioned explicitely.

Transducer elimination Rules **T** decreases the length of w in sub-formulae of the form $u\langle w\rangle s$, if $w \neq \epsilon$ and u is X or any term; it allows to reduce such formulae to the case $w = \epsilon$.

$$\boxed{u\langle (b,p).w\rangle s \quad \mapsto \quad u\langle\epsilon\rangle s \wedge (u\langle\epsilon\rangle b \vee b|_p\langle w\rangle s) \quad \text{if } u \in \mathcal{T} \cup \{X\} \qquad \textbf{(T)}}$$

Preliminary rules The following rules are useful to eliminate the universal quantifiers and variables $x.f$

$$\begin{array}{|ccccc|}\hline \forall f(\phi \wedge \psi) & \mapsto & \forall f\phi \wedge \forall f\psi & & \textbf{(P1)} \\ \forall f(\phi \vee \psi) & \mapsto & (\forall f\phi) \vee \psi \text{ if } f \notin \mathit{var}(\psi) & & \textbf{(P2)} \\ \forall f(X\langle\not\epsilon\rangle x.f \vee \bigvee_{t\in U} t\langle\not\epsilon\rangle x.f) & \mapsto & X, U \lhd \not\epsilon \rhd x & & \textbf{(P3.1)} \\ \forall f(\bigvee_{t\in U} t\langle\not\epsilon\rangle x.f) & \mapsto & U \lhd \not\epsilon \rhd x & & \textbf{(P3.2)} \\ \hline \end{array}$$

Replacement This is the usual rule for substituing a term for a variable.

$$\boxed{x = t \wedge \Phi \quad \mapsto \quad \Phi[t/x] \text{ if } x \notin var(t). \qquad \textbf{(R)}}$$

Elimination of trivial sub-formulae The following formulae eliminate trivially satisfied or unsatisfied sub-formulae (where $\equiv_s$ denotes syntactic equality).

Decomposition rules The following rules deal with equalities and formulae of the form $t\langle\epsilon\rangle s$. The first rule deals with the case where t is not atomic and transforms the formula into equalities, inequalities between terms and formulae of the form $x\langle\epsilon\rangle s$, where x is a variable. The second rule deals with the same formula but for the case where t is atomic.

Occur-check The main idea behind this rule is that $y\langle\not\epsilon\rangle t$ induces an ordering on the variables in t and y. Indeed, if x is a variables in t then, in any model of this formula, the term assigned to x is is a sub-term of the term assigned to y.

Simplification rules Rules **(Si1)** and **(Si2)** deal with formulae of the form $U \lhd \epsilon \rhd s$ and $X, U \lhd \epsilon \rhd s$ in the case that s is not a variable (such formulae can be introduced by the elimination of equalities).

Saturate rules Rules **(Sa1)** and **(Sa2)** allow us to saturate a formula in intermediate form.

$$\begin{array}{cccc}
t = t \mapsto \top & t\langle\epsilon\rangle t \mapsto \bot & \bot \wedge \Phi \mapsto \bot & \top \wedge \Phi \mapsto \Phi \\
x = t \mapsto \bot & x\langle\epsilon\rangle t \mapsto \top & x \neq t \mapsto \top & \text{if } x \in \mathcal{X} \cap \mathit{var}(t) \wedge t \not\equiv_s x \\
\hline
x = \{t\}_k \mapsto \bot & x = (t_1, t_2) \mapsto \bot & x = N \mapsto \bot & x = P \mapsto \bot \\
x = k \mapsto \bot & \multicolumn{3}{c}{\text{if } x \in \mathcal{C} \cup \mathcal{Y} \cup \mathbb{R}, N \in \mathcal{N}, P \in \mathcal{P} \text{ and } k \in \mathcal{K} \cup \mathcal{KS}}
\end{array}$$

$$\begin{array}{cclc}
t\langle\epsilon\rangle s & \mapsto & \mathcal{J}(t, \epsilon, s), \ \text{if } t \notin \mathcal{X} \cup \mathcal{C} \cup \mathcal{Y} \cup \mathbb{R}_{\geq 0} & \textbf{(D1.1)} \\
t\langle\epsilon\rangle s & \mapsto & t \neq s, \ \text{if } t \in \mathcal{C} \cup \mathcal{Y} \cup \mathbb{R}_{\geq 0} & \textbf{(D1.2)} \\
s = t & \mapsto & \mu(s, t) \text{ if } s, t \notin \mathcal{X} \cup \mathcal{C} \cup \mathcal{Y} \cup \mathbb{R}_{\geq 0} & \textbf{(D2)}
\end{array}$$

6.5 Preliminary form

A formula Φ is called in *preliminary form* if it is of the form

$$\exists x_1 \ldots \exists x_m \exists T_1 \ldots \exists T_n \phi$$

where $\{x_i \mid i = 1 \ldots m\} \cup \{T_j \mid j = 1 \ldots n\}$ is the set of all variables that appear in ϕ, and ϕ is a quantifier free formula builded using the connectives $\wedge$ and $\vee$ and the following literals:

$$X\langle\epsilon\rangle t \mid t\langle\epsilon\rangle t' \mid x = t \mid \top \mid X\langle\not\epsilon\rangle t \mid t\langle\not\epsilon\rangle t' \mid X, U \lhd \not\epsilon \rhd x \mid U \lhd \not\epsilon \rhd x$$
$$x \neq t \mid \bot \mid \sum_{i=1}^{n} a_i c_i + \sum_{j=1}^{m} b_j T_j \bowtie d,$$

where X is a fixed second-order variable that ranges over sets of messages, x is a meta-variable that ranges over the set $\mathcal{X}$ of first-order variables, and t, t' range over terms. It is easy to see that repeated application as much as possible of *Transducer elimination* and *Preliminary rules* transform any formula as considered in subsection 6.1, into an equivalent formula in preliminary form. From now on, it is obvious that as we consider satisfiability of formulae in preliminary form, we can restrict ourselves to conjunctions of literals.

6.6 Intermediate form

A formula is called in *intermediate form* if it is syntactically equal to $\top$, $\bot$ or to a conjunction $\Psi \wedge \varphi_1 \wedge \varphi_2$ where Ψ is a time constraint, φ_1 is of the form:

$$\bigwedge_{i=1}^{n_1} X\langle\epsilon\rangle w_i \wedge \bigwedge_{i=1}^{n_2} X\langle\not\epsilon\rangle w'_i \wedge \bigwedge_{i=1}^{n} [\bigwedge_{j=1}^{m_i} x_i\langle\epsilon\rangle t_i^j \wedge \bigwedge_{j=1}^{l_i} x_i\langle\not\epsilon\rangle u_i^j \wedge \bigwedge_{j=1}^{o_i} x_i \neq v_i^j]$$

and φ_2 is of the form:

$$\bigwedge_{i=1}^{n} [\bigwedge_{j=1}^{k_i} X, U_i^j \lhd \not\epsilon \rhd x_i \wedge \bigwedge_{j=1}^{h_i} V_i^j \lhd \not\epsilon \rhd x_i]$$

such that:

- For any $i = 1, \cdots, n$, $x_i \in \mathcal{X}$.

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

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 $y\langle\not\epsilon\rangle t$ with $x \in \mathit{var}(t)$". **(OC)**

If $s \notin \mathcal{X}$ then:

$$U \lhd \notin \rhd s \quad \mapsto \quad \bigwedge_{A \in wc(s)} \Big[\bigvee_{t \in A \setminus \mathcal{X}} \bigvee_{u \in U} u \langle \notin \rangle t \vee \bigvee_{t \in A \cap \mathcal{X}} U \lhd \notin \rhd t \Big] \qquad \textbf{(Si1)}$$

$$X, U \lhd \notin \rhd s \quad \mapsto \quad \bigwedge_{A \in wc(s)} \Big[\bigvee_{t \in A \setminus \mathcal{X}} (X \langle \notin \rangle t \vee \bigvee_{u \in U} u \langle \notin \rangle t) \vee \bigvee_{t \in A \cap \mathcal{X}} X, U \lhd \notin \rhd t \Big] \qquad \textbf{(Si2)}$$

$$\phi \wedge x \langle \notin \rangle t \wedge X, U \lhd \notin \rhd x \quad \mapsto \quad \phi \wedge x \langle \notin \rangle t \wedge X, U \lhd \notin \rhd x \wedge X, U \lhd \notin \rhd t \qquad \textbf{(Sa1)}$$

$$\phi \wedge x \langle \notin \rangle t \wedge U \lhd \notin \rhd x \quad \mapsto \quad \phi \wedge x \langle \notin \rangle t \wedge U \lhd \notin \rhd x \wedge U \lhd \notin \rhd t \qquad \textbf{(Sa2)}$$

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

A formula in intermediate form defined as above, is called *saturated*, if $\Psi \wedge \varphi_1 \wedge \varphi_2$ is satisfiable if and only if $\Psi \wedge \varphi_1$ is satisfiable.

From preliminary form to intermediate form

Theorem 6.2 *Application of the rules of Subsection 6.4 terminates in an intermediate form.*

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

1. Rules **D1.1** and **D1.2** decrease the number of sub-formulae of the form $t\langle \epsilon \rangle s$ or $t\langle \not\epsilon \rangle s$ but may introduce equalities and disequalities.
2. Rule **D2** decreases the number of equalities (disequalities) where the two members are not variables.
3. Rules **Sa1** and **Sa2** decrease the number of sub-formulae of the form $X, U \lhd \notin \rhd s$ and $U \lhd \notin \rhd s$ but may introduce new formulae of the form $t\langle \notin \rangle s$.
4. 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 cardinality of $\mathit{var}\,(\varphi)$.
- $f_2(\varphi)$ is the number of formulae of the form $X, U \lhd \notin \rhd s$ and $U \lhd \notin \rhd s$ with s not a variable.
- $f_3(\varphi)$ is the number of formulae of the form $t\langle \epsilon \rangle s$ with t not a variable.
- $f_4(\varphi)$ is the number of equalities (disequalities) where both members are not variables.
- $f_5(\varphi)$ is the size of φ.

Figure 1 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 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 indermediate form. This proof is easy and tedious and is left to the reader. ∎

	f_1	f_2	f_3	f_4	f_5
E	$=$	$=$	$=$	$=$	$<$
D2	$=$	$=$	$=$	$<$	
$\mathbf{D1.1} + \mathbf{D1.2}$	$=$	$=$	$<$		
$\mathbf{Sa1} + \mathbf{Sa2}$	$=$	$<$			
$\mathbf{R} + \mathbf{OC}$	$<$				

Figure 1: Variation of the ranking functions

Saturating formulae in intermediate form We prove that for any formula φ in intermediate form, we can construct a set of saturated formulae $\varphi_1; \ldots; \varphi_n$ in intermediate form, such that $[\![\varphi]\!] = \bigcup_{i=1}^{n} [\![\varphi_i]\!]$.

Theorem 6.3 *There exists a strategy to apply the rules of Subsection 6.4 that terminates in a saturated intermediate form.*

Proof: To ensure the termination, we apply the *Saturate rules* **Sa1** and **Sa2** only for the pairs $(x\langle \not\epsilon \rangle t;\ X, U \lhd \not\epsilon \rhd x)$ or $(x\langle \not\epsilon \rangle t;\ V \lhd \not\epsilon \rhd x)$ that are not marked, and after the application of such a rule, we mark the corresponding pair. On the other hand, any time we apply the *Replacement* or the *Occur-check rule*, we unmark all the pairs of constraints which were marked before. Then, the termination follows from the remark that the number of variables is finite, all the rules but Replacement or Occur-check introduce only subformulae of the formulae we already have, and no rule does not introduce any new variable. Then, to prove that any formula obtained after the termination of the above algoritm is saturated, we make an induction on the position of variables w.r.t. to the order $\leq$. Indeed let $\Psi \wedge \varphi_1 \wedge \varphi_2$ be such a formula and let σ be the substitution and E be the set of messages as defined in Theorem 6.1 corresponding to the formula $\Psi \wedge \varphi_1$. Then, (σ, E) satisfies $\Psi \wedge \varphi_1 \wedge \varphi_2$. We prove by induction on the position of the variable x w.r.t. to the order $\leq$ that for any constraints $X, U \lhd \not\epsilon \rhd x \in \varphi_2$, and $V\langle \not\epsilon \rangle x \in \varphi_2$ and for any $A \in wc(\sigma(x))$ it holds $(E \cup U)\sigma\langle \not\epsilon \rangle A$, and $V\sigma\langle \not\epsilon \rangle A$. The key of the proof, is that when the algorithm terminates, we already applied the Saturate rules to the pairs $(x\langle \notin \rangle t;\ X, U \lhd \notin \rhd x)$ and $(x\langle \notin \rangle t;\ V \lhd \notin \rhd x)$, and such rules introduce only formulae of the kind ϕ_1, or of the form $X, U \lhd \notin \rhd z$ or $V \lhd \notin \rhd z$ with $z \leq x$. ■

6.7 Complexity of the decidability of satisfiability of a TSPL-formula

In this subsection we prove that the problem of deciding the existence of a model for a TSPL formula (shortly called TSPL-SAT) is NP-complete. We define the size of a formula φ to be the size of its DAG representation. Roughly speaking, it is the cardinality of the set of its sub-formulae and sub-terms. We denote the size of φ by $|\varphi|$.

First, we prove the NP-hardness, using a polynomial reduction of 3-SAT to TSPL-SAT. Let $x_1, \ldots, x_n$ be Boolean variables, and let $f = \bigwedge_{i=1}^{m} l_i^1 \vee l_i^2 \vee l_i^3$ a formula in 3-conjunctive normal form, where $l_i^j \in \{x_1, \ldots, x_n, \neg x_1, \ldots \neg x_n\}$. It is well known that deciding the existence of a model for such a formula is NP-complete. We shall construct a TSPL-formula φ_f such that φ_f is satisfiable if and only if f is satisfiable.

Let $c, T, F \in \mathcal{A}$ be three distinct constants and let $k_1, k_2 \in K$ be two distinct keys in K. For any literal l, we denote

$$\mathbf{t}(l) = \begin{cases} \{c, x_j\}_{k_1} & \text{if } l = x_j \\ \{c, x_j\}_{k_2} & \text{if } l = \neg x_j \end{cases}$$

and for any clause $C = l^1 \vee l^2 \vee l^3$ we denote $\mathbf{t}(C) = ((\mathbf{t}(l^1), \mathbf{t}(l^2)), \mathbf{t}(l^3))$. Then, for any clause $C_i = l_i^1 \vee l_i^2 \vee l_i^3$, we consider the formula

$$\varphi_{C_i} = \neg(\mathbf{t}(C_i), \{\mathbf{t}(C_i), T\}_{k_1})\langle(\{x, T\}_{k_1}, 11).(\{y, F\}_{k_2}, 11)\rangle c.$$

and finally, for $f = \bigwedge_{i=1}^{m} C_i$, we take $\varphi_f = \bigwedge_{i=1}^{m} \varphi_{C_i}$.

Now we prove that models of f coincide with models of φ_f. More precisely, given a substitution $\sigma : \{x_1, \dots, x_n\} \to \{T, F\}$, let $\hat{\sigma}$ denote the boolean function such that $\hat{\sigma}(x_i) = \top$, if $\sigma(x_i) = T$ and $\hat{\sigma}(x_i) = \bot$, othewise. Then, we prove

$$\sigma \models \varphi_f \text{ iff } \hat{\sigma} \models f.$$

For any variable x_i, if a substitution σ satisfies $\neg\mathbf{t}(x_i)\langle(\{x,T\}_{k_1}, 11)\rangle c$, then $\sigma(x_i) = T$ and similarly, if σ satisfies $\neg\mathbf{t}(\neg x_i)\langle(\{y,F\}_{k_2}, 11)\rangle c$, then $\sigma(x_i) = F$. Moreover, the formulae $\neg\mathbf{t}(\neg x_i)\langle(\{x,T\}_{k_1}, 11)\rangle c$ and $\neg\mathbf{t}(x_i)\langle(\{y,F\}_{k_2}, 11)\rangle c$ are not satisfiable. Therefore, σ is a model of

$$\neg\mathbf{t}(C_i)\langle(\{x,T\}_{k_1}, 11)\rangle c \vee \neg\mathbf{t}(C_i)\langle\{y,F\}_{k_2}, 11)\rangle c$$

iff $\hat{\sigma}$ is a model of $C_i = l_i^1 \vee l_i^2 \vee l_i^3$. And hence,

$$\sigma \models \varphi_f \text{ iff } \hat{\sigma} \models \varphi.$$

Now, since if φ has a model, then it has a model that maps each variable x_i into $\{T, F\}$, and since φ_f is polynomial in the size of f, we have the following:

Proposition 6.1 TSPL-*SAT is NP-hard.*

Now we prove that TSPL-SAT is in NP. We only consider term formulae as the complexity of time constraints is well-konwn [28]. Moreover, as seen in Section 6.3 time constraints can be eliminated leading to a formula Ψ'. This can be done in NP-time and the size of Ψ' is polynomial in the size of Ψ. Moreover, from Ψ' we obtain an equivalent formula Ψ'' in preliminary form, such that $|\Psi''|$ is polynomially bounded by $|\Psi'|$.

Let φ be a conjunction of literals in preliminary form. Now let $|\varphi|_t$ be the cardinality of the set $St(\varphi)$ consisting of the sub-terms of φ. Clearly, we have $|\varphi|_t \le |\varphi|$. Then, we show that if φ is satisfiable then it has a model σ such that the size of $\sigma(x)$ is polynomially bounded by $|\varphi|_t$, for each variable x. To do so, we first introduce a special kind of substitutions.

Definition 6.1 *Let Tr a set of terms, and let ρ a substitution defined on the set of variables of Tr. Then ρ is called a Tr-substitution, if:*

- *for any variable $x \in dom(\rho)$, there is a term $v_x \in St(Tr)$ which is not a variable (i.e. $v_x \notin \mathcal{X}$), and such that $\rho(x) = \rho(v_x)$;*
- *ρ is idempotent.*

Then, the following result can be proved using a similar reasoning as in the Theorem 1 from [26]:

Proposition 6.2 *Let Tr a set of terms and let ρ a Tr-substitution. Then for any variable x, $|\rho(x)| \le |St(Tr)|$.*

Now, let $\{x_1, \cdots, x_n\}$ be the variables of φ. For each x_i, we introduce $|\varphi|_t$ new variables $z_1^i, \cdots, z_{|\varphi|_t}^i$ and consider the term t_{x_i} defined as follows:

$$t_{x_i} = (z_1^i, \cdots (z_{|\varphi|_t}^i, F(|\varphi|_t) + i)) \cdots).$$

Let us assume that φ is satisfiable. Then, there exists a formula ψ in solved form such that ψ is obtained from φ using the rewriting rules from Section 6.4, and such that ψ is satisfiable. Then, we can prove that the assertion

“there exists a $St(\varphi)$- substitution σ_0, such that $St(\psi) \subseteq (St(\varphi))\sigma_0$”

is an invariant for the rewriting rules of Subsection 6.4. Hence, using Proposition 6.2, we obtain that $|\psi|_t$ is polynomial in $|\varphi|_t$. Now, let (E, σ) be the model of Subsection 6.2 that satisfies ψ. Then, we can check that for any variable x, there is a term u_x in $St(\psi) \cup \{t \mid t \le t_{x_i}, i = 1, \cdots, n\}$ such that $\sigma(x) = \sigma(u_x)$, and u_x is not a variable. Then, using again Proposition 6.2 we obtain that the size of σ is polynomial in the size of $St(\psi) \cup \{t_{x_i} \mid i = 1, \cdots, n\}$. Now, since the size of $\{t \mid t \le t_{x_i}, i = 1, \cdots, n\}$ is polynomial in the size of $St(\varphi)$, and since $|E| \le |St(\psi)|$ we have the following:

Proposition 6.3 TSPL-*SAT is in NP.*

6.8 Undecidability for the entire TSPL logic

In this section we prove that the TSPL logic is undecidable, if we allow both existential and universal quantifiers. We show that Post's correspondence problem is reducible to the decision problem in our logic. The proof is inspired from [30], where it is shown the undecidability of a certain fragment in the theory of free term algebras.

Theorem 6.4 *Post's correspondence problem is reducible to the decision problem for the* TSPL *logic.*

Proof:

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 shall 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 shall denote, $g_i(x) = \mathbf{pair}(d_i, x)$. The monadic functions g_i represent the alphabet: the string $d_{i_1} \ldots d_{i_j}$ is represented by the term $g_{i_1}(\ldots(g_{i_j}(c))\ldots)$. By abus of notation, if $e = d_{i_1} \ldots d_{i_j}$, we write $e(y)$ to mean $g_{i_1}(\ldots(g_{i_j}(y))\ldots)$ 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 $1 \le i_j \le n$ 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

$\exists x, x_1, x_2 \forall y_0, \ldots, y_6$

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

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

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

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

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

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

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 subformula 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 subterms 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 $s_j = q_i s_{j+1}$.

■

7 Conclusions

In this paper, we have proved the decidability of a large class of reachability properties, including secrecy and authentication, for **timed** bounded protocols. Our model for specifying timed protocols uses clocks, time variables and time-stamps. This work can be extended in several ways:

1. our model can be naturally extended to associate time values to short term keys such that if the intruder obtains a message encrypted by a short term key then after the specified amount of time elapses the key becomes known by the intruder. Our model and verification method can be extended to handle this model.

2. our model can also be extended to handle drifting clocks. It is well-known that models with clocks with drifts in bounded intervals can be transformed into models with perfect clocks modulo an abstraction, that is, taking into account more behavior. As discussed by Gong [18] drifting clocks can add subtle attacks.

3. in [7], it is shown how we can use our logic to devise an abstract interpretation based method for unbounded protocols.

References

[1] R. Alur and D. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126, 1994. 1

[2] R. Alur, T. Feder, and T.A. Henzinger. The benefits of relaxing punctuality. In *Proceedings of the 10th ACM Symposium on Principles of Distributed Computing*, pages 139–152. ACM Press, 1991. 1

[3] 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. 1

[4] G. Bella and L. C. Paulson. Mechanizing BAN Kerberos by the inductive method. In A. J. Hu and M. Y. Vardi, editors, *Proceedings of the 10th International Conference on Computer-Aided Verification (CAV'98)*, pages 416–427, Vancouver, B.C., Canada, June 1998. Springer-Verlag LNCS 1427. 1

[5] M. Boreale. Symbolic trace analysis of cryptographic protocols. In *ICALP: Annual International Colloquium on Automata, Languages and Programming*, 2001. 1

[6] A. Bouajjani and Y. Lakhnech. Temporal logic + timed automata: expressiveness and decidability. In I. Lee and S. A. Smolka, editors, *CONCUR'95: Concurrency Theory*, volume 962 of *LNCS*, pages 531–546. Springer-Verlag, 1995. 1

[7] L. Bozga. *Automatic verification of cryptographic protocols*. PhD thesis, University Joseph Fourier (Grenoble 1), 2004. 3

[8] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, February 1990. A

[9] John A. Clark and Jeremy L. Jacob. A survey of authentication protocol literature. Version 1.0, University of York, Department of Computer Science, November 1997. 3.1

[10] Ernie Cohen. Taps: A first-order verifier for cryptographic protocols. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop (CSFW'00)*, page 144. IEEE Computer Society, 2000. 1

[11] H. Comon. Disunification: A survey. In *Computational Logic: Essays in Honor of Alan Robinson*. MIT Press, Cambridge, MA, 1991. 6.1

[12] 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. 1

[13] 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. 1

[14] V. Cortier, J. Millen, and H. Rueß. Proving secrecy is easy enough. In *IEEE Computer Security Foundations Workshop*, pages 97–110, 2001. 4.1.1

[15] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983. 3

[16] Neil Evans and Steve Schneider. Analyzing time dependent security properties in CSP using PVS. In *ESORICS*, pages 222–237, 2000. 1

[17] 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. 1

[18] Li Gong. A security risk of depending on synchronized clocks. *Operating Systems Review*, 26(1):49–53, 1992. 2

[19] T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model-checking for real-time systems. In *Seventh Annual IEEE Symposium on Logic in Computer Science*, pages 394–406. IEEE Computer Society Press, 1992. 1

[20] 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. 6.1

[21] G. Lowe. Breaking and fixing the Needham-Schroeder Public-Key protocol using FDR. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 1055 of *LNCS*, pages 147–166, 1996. 1

[22] 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. A

[23] 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. 1

[24] L. Paulson. Proving properties of security protocols by induction. In *IEEE Computer Security Foundations Workshop*, pages 70–83, 1997. 4.1.1

[25] 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. A

[26] M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *IEEE Computer Security Foundations Workshop*, 2001. 1, 6.7

[27] 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. A

[28] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons Ltd., Chichester, 1986. 6.7

[29] J. Thayer, J. Herzog, and J. Guttman. Honest Ideals on Strand Spaces. In *IEEE Computer Security Foundations Workshop*, pages 66–78, 1998. 1, 4.1.1

[30] K. N. Venkataraman. Decidability of the purely existential fragment of the theory of term algebras. *JACM*, 1987. 6.8

[31] Thomas Y. C. Woo and Simon S. Lam. Authentication for distributed systems. *Computer*, 25(1):39–52, January 1992. A

A Expressing security properties

To illustrate how TSPL can be used to express security properties, 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:

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

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.

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 us 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:

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

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

We can express semantic secrecy in our logic using the following result:

Proposition A.1 *Let t be a term. Then,*

$$[\![\bigvee_{S' \in wc_t(t)} X\langle\epsilon\rangle S']\!] = \{(\sigma, E, \nu, \ell) \mid E \not\vdash t\sigma\}.$$

There are many definitions of authentication that we can find in the literature [8, 31, 22, 27, 25]. We show here, by means of an example, how the introduced logic allows to specify the authentication properties discussed in [22].

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 us 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).

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

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).

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

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.

$$\begin{array}{lll} pc_B^{(S1)} = 3 & \Rightarrow & (y^{(S1)} = a \wedge \\ & & ((pc_A^{(S1)} \neq 0 \wedge p^{(S1)} = b \wedge z^{(S1)} = {N_a}^{(S1)}) \vee \\ & & \quad (pc_A^{(S2)} \neq 0 \wedge p^{(S2)} = b \wedge z^{(S1)} = {N_a}^{(S2)}))) \end{array}$$

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.

$$\begin{array}{lll} pc_B^{(S1)} = 3 & \Rightarrow & \\ & & [y^{(S1)} = a \wedge \\ & & ((pc_A^{(S1)} \neq 0 \wedge p^{(S1)} = b \wedge z^{(S1)} = {N_a}^{(S1)} \wedge x^{(S1)} = {N_b}^{(S1)} \wedge \\ & & \quad (pc_A^{(S2)} = 0 \vee p^{(S2)} \neq b \vee z^{(S1)} \neq {N_a}^{(S2)} \vee x^{(S2)} \neq {N_b}^{(S1)})) \vee \\ & & (pc_A^{(S2)} \neq 0 \wedge p^{(S2)} = b \wedge z^{(S1)} = {N_a}^{(S2)} \wedge x^{(S2)} = {N_b}^{(S1)} \wedge \\ & & \quad (pc_A^{(S1)} = 0 \vee p^{(S1)} \neq b \vee z^{(S1)} \neq {N_a}^{(S1)} \vee x^{(S1)} \neq {N_b}^{(S1)})\ ) \\ & &)] \end{array}$$

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.

B Proofs

B.1 Proof of Proposition 4.1

Proposition 4.1. *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. $\neg E\langle w_i, S_i\rangle_I$ 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 $\neg m'\langle w_i, S_i\rangle_I$.

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 $\neg m'\langle w_i, S_i\rangle_I$. 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 $E\langle w_i, S_i\rangle_I$.
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 $m_1\langle w_i, S_i\rangle_I$ and $k_1\langle w_i, S_i\rangle_I$. Since $\neg m\langle w_i, S_i\rangle_I$, there exists $i \in I$ such that $\neg m\langle w_i\rangle S_i$. It follows that $w_i \neq \epsilon$ and hence $w_i = (b, r).w$ and $m = b$ and $\neg b|_r\langle w_i\rangle S_i \quad (*)$.

   If $NT(b, r)$ does not exist then we have $\neg m_1\langle \epsilon\rangle S_i$, and hence, $\neg m_1\langle w_i\rangle S_i$, which contradicts the derivation-minimality of m.

   So, let $(b_1, r_1) = NT(b, r)$. From definition, we have that $b|_r = b_1|_{r_1} \quad (**)$

   Since $(w_i, S_i)_{i\in I}$ is well-formed, following the definition 4.4(2a) 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 and $m = b$, we obtain that either $m_1 \in S_j$ or $k_1 \in S_j$ and hence either $\neg m_1\langle w_j\rangle S_j$ or $\neg k_1\langle w_j\rangle S_j$, contradiction.

   Hence $w_j = (b_1, r_1).w$ and $S_i \subseteq S_j$. From $m_1\langle w_j\rangle S_j$ we obtain $b_1|_{r_1}\langle w\rangle S_j$ and using $(**)$ we obtain $b|_r\langle w\rangle S_j \quad (***)$.

   From $(*)$, $(***)$ and the definition 4.4(2a) 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 $m_1\langle w_i, S_i\rangle_I$, and then we obtain that $m\langle w_i, S_i\rangle_I$, 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 $m_1\langle w_i, S_i\rangle_I$ and $k_1^{-1}\langle w_i, S_i\rangle_I$. If we suppose that $k_1 \notin K$, then we obtain that either $\neg m_1\langle w_i, S_i\rangle_I$ or $m\langle w_i, S_i\rangle_I$, 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 $k_1^{-1}\langle w_i, S_i\rangle_I$.

■

B.2 Proof of proposition 4.2

In order to prove the Proposition 4.2 we start with proving the following proposition:

Proposition B.1 *Let m, s be 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 B.1 *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 B.1 we obtain that $E, m \vdash s$. But $m \in E$ hence we have $E \vdash s$, contradiction. ■ **Proposition 4.2.** *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: "⇒":** $E \not\vdash m \Rightarrow \exists A \in \mathrm{wc}(m)$ 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 B.1 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 B.1 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.

"⇐": $\exists A \in \mathrm{wc}(m)$ 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 4.1 we obtain that $m\langle\epsilon\rangle A$. But, from $A \in \mathrm{wc}(m)$ we have $m \in A$, contradiction. ■

B.3 Definability of $t\langle w\rangle S$ in TSPL

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

Let t be a term, s be an extended term, let w be a sequence of term transducers and let $\mathcal{J}$ be defined as follows :

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

where we denote by

$$\mathcal{G}(t, s) = \begin{cases} \neg\mu(a, s) & \text{if } s \notin \mathcal{BX} \\ t\langle\epsilon\rangle s & \text{if } s \in \mathcal{BX} \end{cases}$$

Then, $t\langle w\rangle s \equiv \mathcal{J}(t, w, s)$, i.e., both formulae are equivalent. **Proof:** We give the proof for the first case, when $s \notin \mathcal{BX}$, the other case being similar. First, notice that if $\mu(t_1, t_2) \neq \bot$, then $(\sigma, E, \nu, l) \in \mu(t_1, t_2)$ iff $t_1\sigma = t_2\sigma$, and if $\mu(t_1, t_2) = \bot$, then for any (σ, E, ν, l), 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 $(\sigma, E, \nu, l) \in \neg\mu(a, s)$ iff $s\sigma \neq a$ iff $a\langle w\sigma\rangle s\sigma$ iff $(\sigma, E, \nu, l) \in a\langle w\rangle s$.
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

   $(\sigma, E, \nu, l) \in \mathcal{J}(t, w, s)$ iff $(\sigma, E, \nu, l) \in t_1\langle w\rangle s \wedge t_2\langle w\rangle s \wedge \neg\mu(t, s)$ 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 $(\sigma, E, \nu, l) \in t\langle w\rangle s$.
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 $(\sigma, E, \nu, l) \in t\langle w\rangle s$ iff $t\sigma\langle\epsilon\rangle s\sigma$ iff $t\sigma \neq s\sigma$ iff $(\sigma, E, \nu, l) \in \neg\mu(t, s)$ iff $(\sigma, E, \nu, l) \in \mathcal{J}(t, w, s)$.
6. If $t = \{t_1\}_k \wedge k \in K \wedge w = (b, r).w_1$, then $\mathcal{J}(t, w, s) = \neg\mu(t, s) \wedge (\neg\mu(b, t) \vee \mathcal{J}(b|_r, w_1, s))$. By induction hypothesis, we have that $b|_r\langle w_1\rangle s \equiv \mathcal{J}(b|_r, w_1, s)$.

   We obtain $(\sigma, E, \nu, l) \in t\langle w\rangle s$ 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 $t\sigma \neq s\sigma \wedge (b\sigma = t\sigma \Rightarrow (b|_r)\sigma\langle w_1\sigma\rangle s\sigma)$ iff $(\sigma, E, \nu, l) \in \neg\mu(t, s) \wedge (\mu(b, t) \Rightarrow b|_r\langle w_1\rangle s)$ iff $(\sigma, E, \nu, l) \in \neg\mu(t, s) \wedge (\mu(b, t) \Rightarrow \mathcal{J}(b|_r, w_1, s))$ iff $(\sigma, E, \nu, l) \in \mathcal{J}(t, w, s)$.

■

B.4 Proof of Lemma 5.1

Lemma 5.1. *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 $dom(\sigma) \cap \tilde{x} = \emptyset$. Then it holds $(\sigma, E, \nu, l) \in [\![F(t\rho)]\!]$ iff $E\sigma \vdash t(\sigma \oplus \rho)$.* **Proof:** Since $dom(\sigma) \cap \tilde{x} = \emptyset$ and using the Definition 4.5, we have

$(\sigma, E, \nu, l) \notin [\![F(t\rho)]\!]$ iff $(\sigma, E, \nu, l) \in \exists\overrightarrow{f} \bigcup_{S' \in wc(t)} [\![X\langle\epsilon\rangle S'\rho]\!]$ iff
$\exists\overrightarrow{f}\exists S' \in wc(t)\ s.t.\ (\sigma, E, \nu, l) \in [\![X\langle\epsilon\rangle S'\rho]\!]$ iff
$\exists\overrightarrow{f}\exists S' \in wc(t)\ s.t.\ E\sigma\langle\epsilon\rangle(S'\rho)\sigma$ iff
$\exists\overrightarrow{f}\exists S' \in wc(t)\ s.t.\ E\sigma\langle\epsilon\rangle S'(\sigma \oplus \rho)$ iff
$\exists S' \in wc(t(\sigma \oplus \rho))\ s.t.\ E\sigma\langle\epsilon\rangle S'$ iff (using Proposition 4.2)
$E\sigma \nvdash t(\sigma \oplus \rho)$. ■

B.5 Proof of Lemma 5.2

Lemma 5.2. *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*

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

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

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

Then $[\![\Phi]\!] = [\![\rho(\mathcal{K}(t, x, w, S))]\!]$. **Proof: "⇐":** Let suppose that $(\sigma, E, \nu, l) \in [\![\rho(\mathcal{K}(t, x, w, S))]\!]$. Since $(\sigma, E, \nu, l) \in [\![X\langle w\rho\rangle S\rho]\!]$, 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 $(\sigma, E, \nu, l) \in [\![\mathcal{H}(X\langle(t|_p, p^{-1}p_x).w\rangle S)]\!]$ 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 4.1, 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 4.1 we obtain $\rho(x)\langle w(\sigma \oplus \rho)\rangle S(\sigma \oplus \rho)$.

"⇒": We have that $(\sigma, E, \nu, l') \in [\![\rho(x)\langle w\rho\rangle S\rho \wedge X\langle w\rho\rangle S\rho]\!]$. Let suppose that $\forall p_x \in P_{x,t} \exists NT(t, p_x)$. We have to prove for any $p_x \in P_{x,t}$ that:

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

From $(\sigma, E, \nu, l') \in [\![\rho(x)\langle w\rho\rangle S\rho \wedge X\langle w\rho\rangle S\rho]\!]$ 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 $(\sigma, E, \nu, l) \in [\![\mathcal{H}((X\langle(t|_p, p^{-1}p_x).w\rangle S)]\!]$. First 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 4.1 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 4.3.

The case $\exists p_x \in P_{x,t} \not\exists NT(t, p_x)$ is similar. ■

B.6 Proof of Proposition 5.2

To prove the Proposition 5.2, we need an auxiliary Lemma.

Lemma B.1 *Let $\rho \in \Gamma(\tilde{x})$, and let φ an atomic term formula. Then*

$$\rho(\mathbf{Pre}(l \xrightarrow{g,\mathcal{R},?t(\tilde{x})} l', \varphi)) \equiv \rho(g) \wedge (pc = l) \wedge F(t\rho) \wedge \rho(\varphi)$$

Proof: By an analysis by cases on φ. The only non-trivial case is when $\varphi = (X, x)\langle w\rangle S$ and $x \in \tilde{x}$, and the assertion follows as a direct consequence of Lemma 5.1 and Lemma 5.2. ■ **Proposition 5.3.** *For any input action α and term formula φ,*

$$pre(\alpha(\tilde{x}), [\![\varphi]\!]) = [\![\exists\tilde{x} \cdot \mathbf{Pre}(\alpha, \varphi)]\!].$$

Proof: In the sequel l'' is a label, E is a set of terms and σ is a ground substitution such that $\tilde{x} \cap (dom(\sigma) \cup var(E)) = \emptyset$.

$(\sigma, E, \nu, l'') \in pre(l \xrightarrow{g,\mathcal{R},?t(\tilde{x})} l', [\![\varphi]\!])$ iff
$\exists\rho \in \Gamma(\tilde{x})$ s.t. $l'' = l \wedge E\sigma \vdash t(\sigma \oplus \rho) \wedge [\![g]\!]_{\nu,\sigma\oplus\rho} = 1 \wedge (\sigma \oplus \rho, E, \nu, l'') \in [\![\varphi]\!]$ iff
$\exists\rho \in \Gamma(\tilde{x})$ s.t.$(\sigma, E, \nu, l'') \in [\![pc = l \wedge \rho(g)]\!] \wedge (\sigma, E, \nu, l'') \in [\![F(t\rho)]\!] \wedge (\sigma, E, \nu, l'') \in [\![\rho(\varphi)]\!]$ iff
$\exists\rho \in \Gamma(\tilde{x})$ s.t.$(\sigma, E, \nu, l'') \in [\![\rho(g) \wedge (pc = l) \wedge F(t\rho) \wedge \rho(\varphi)]\!]$ iff
$\exists\rho \in \Gamma(\tilde{x})$ s.t.$(\sigma, E, \nu, l'') \in [\![\rho(\mathbf{Pre}(l \xrightarrow{g,\mathcal{R},?t(\tilde{x})} l', \varphi))]\!]$ iff
$(\sigma, E, \nu, l'') \in [\![\exists\tilde{x} \cdot \mathbf{Pre}(l \xrightarrow{g,\mathcal{R},?t(\tilde{x})} l', \varphi)]\!]$ ■

B.7 Computation of predecessors for a sequence of actions

In this subsection, we give an example that shows how we compute the set of predecessors with respect to a simple protocol.

Example B.1
Let $\alpha_0 = l_0 \xrightarrow{\top,\emptyset,!\{c\}_k} l_1$, $\alpha_1 = l_1 \xrightarrow{\top,\{d\},!\{c\}_k} l_2$, $\alpha_2 = l_2 \xrightarrow{g_2,\emptyset,?\{T\}_k} l_3$, $\alpha_3 = l_3 \xrightarrow{\top,\emptyset,!s} l_4$ where c and d are clocks, k is a symmetric key (intended to remain secret for the intruder), s is a message (the secret) and $g_2 \equiv d = 1 \wedge -c + T < -1$. Let $\Phi \stackrel{def}{=} X\langle\not\epsilon\rangle s$ be the formula that represents the "bad configurations" (where secret s is known to the intruder). Now let $\Pi_1 = \alpha_1\alpha_2\alpha_3$ and $\Pi_0 = \alpha_0\alpha_1\alpha_2\alpha_3$ are two protocols. We show that Π_1 is secure w.r.t. to formula Φ, while the same assertion does not hold for Π_2. For sake of simplicity, we work modulo $\equiv$.

Then $\mathbf{Pre}(\xrightarrow{\tau}, X\langle\not\epsilon\rangle s) = X\langle\not\epsilon\rangle s$.

Pre$(\alpha_3, X\langle\not\in\rangle s) = \top \wedge pc = l_3 \wedge (X, s)\langle\not\in\rangle s \equiv pc = l_3$
Pre$(\xrightarrow{\tau}, pc = l_3) = pc = l_3$
Pre$(\alpha_2, pc = l_3) = \Phi_1$ where

$$\Phi_1 \equiv d = 1 \wedge -c + T < -1 \wedge pc = l_2 \wedge X\langle\not\in\rangle\{\{T\}_k, T\} \wedge X\langle\not\in\rangle\{\{T\}_k, k\}$$

Pre$(\xrightarrow{\tau}, \Phi_1) = \Phi_2$ where

$$\Phi_2 \equiv (d \leq 1) \wedge d - c + T < 0 \wedge pc = l_2 \wedge X\langle\not\in\rangle\{\{T\}_k, T\} \wedge X\langle\not\in\rangle\{\{T\}_k, k\}$$

Pre$(\alpha_1, \Phi_2) = \Phi_3$ where

$$\begin{array}{rcl}\Phi_3 & \equiv & (0 \leq 1) \wedge 0 - c + T < 0 \wedge pc = l_1 \wedge (X, \{T_c\}_k)\langle\not\in\rangle\{\{T\}_k, T\} \wedge \\ & & (X, \{T_c\}_k)\langle\not\in\rangle\{\{T\}_k, k\} \wedge T_c = c \\ & \equiv & -c + T < 0 \wedge pc = l_1 \wedge T_c = c \wedge T < T_c \wedge ((X\langle\not\in\rangle\{\{T\}_k, T\} \wedge \\ & & X\langle\not\in\rangle\{\{T\}_k, k\}) \vee T_c = T) \\ & \equiv & -c + T < 0 \wedge pc = l_1 \wedge T_c = c \wedge T < T_c \wedge \\ & & (X\langle\not\in\rangle\{\{T\}_k, T\} \wedge X\langle\not\in\rangle\{\{T\}_k, k\})\end{array}$$

Pre$(\xrightarrow{\tau}, \Phi_3) = \Phi_4$ where

$$\Phi_4 \equiv pc = l_1 \wedge c \leq T_c \wedge T < T_c \wedge X\langle\not\in\rangle\{\{T\}_k, T\} \wedge X\langle\not\in\rangle\{\{T\}_k, k\}$$

Pre$(\alpha_0, \Phi_4) = \Phi_5$ where

$$\begin{array}{rcl}\Phi_5 & \equiv & pc = l_0 \wedge c \leq T_c \wedge T < T_c \wedge \\ & & (X, \{T'_c\}_k)\langle\not\in\rangle\{\{T\}_k, T\} \wedge (X, \{T'_c\}_k)\langle\not\in\rangle\{\{T\}_k, k\} \wedge T'_c = c \\ & \equiv & pc = l_0 \wedge c \leq T_c \wedge T < T_c \wedge T'_c = c \wedge T'_c < T_c \wedge ((X\langle\not\in\rangle\{\{T\}_k, T\} \wedge \\ & & X\langle\not\in\rangle\{\{T\}_k, k\}) \vee T'_c = T)\end{array}$$

Pre$(\xrightarrow{\tau}, \Phi_5) = \Phi_6$ where

$$\begin{array}{rcl}\Phi_6 & \equiv & pc = l_0 \wedge c \leq T_c \wedge c \leq T'_c \wedge T < T_c \wedge T'_c < T_c \wedge \\ & & ((X\langle\not\in\rangle\{\{T\}_k, T\} \wedge X\langle\not\in\rangle\{\{T\}_k, k\}) \vee T'_c = T)\end{array}$$

Hence, we obtain

$$pre(\Pi_1, X\langle\not\in\rangle s) \equiv pc = l_1 \wedge c \leq T_c \wedge T < T_c \wedge X\langle\not\in\rangle\{\{T\}_k, T\} \wedge X\langle\not\in\rangle\{\{T\}_k, k\}$$

and

$$\begin{array}{rcl}pre(\Pi_0, X\langle\not\in\rangle s) & \equiv & pc = l_0 \wedge c \leq T_c \wedge c \leq T'_c \wedge T < T_c \wedge T'_c < T_c \wedge \\ & & ((X\langle\not\in\rangle\{\{T\}_k, T\} \wedge X\langle\not\in\rangle\{\{T\}_k, k\}) \vee T'_c = T)\end{array}$$

Since we supposed that k is a secret symmetric key (i.e.$X\langle\not\in\rangle k$), if there is no any message of the form $\{T''\}_k$ known initially to the intruder, the protocol Π_1 is secure with respect to the secrecy of s. On the contrary, protocol Π_0 is unsecure. If we pick T_c, and T'_c such that $T < T_c \wedge T'_c < T_c \wedge T'_c = T$, then we obtain an attack, that corresponds to the fact that the first message sent by our participant can be replayed successfully by the intruder (it satisfies the time constraints), while the same is not true for the second sent message.