Selecting Theories and Recursive Protocols[⋆]

Tomasz Truderung

LORIA-INRIA-Lorraine, FRANCE
Institute of Computer Science, Wrocław University, POLAND

Abstract. Many decidability results are known for non-recursive cryptographic protocols, where the protocol steps can be expressed by simple rewriting rules. Recently, a tree transducer-based model was proposed for *recursive* protocols, where the protocol steps involve some kind of recursive computations. This model has, however, some limitations: (1) rules are assumed to have linear left-hand sides (so no equality tests can be performed), (2) only finite amount of information can be conveyed from one receive-send action to the next ones. It has been proven that, in this model, relaxing these assumptions leads to undecidability.
In this paper, we propose a formalism, called *selecting theories*, which extends the standard non-recursive term rewriting model and allows participants to compare and store arbitrary messages. This formalism can model recursive protocols, where participants, in each protocol step, are able to send a number of messages unbounded w.r.t. the size of the protocol. We prove that insecurity of protocols with selecting theories is decidable in NEXPTIME.

1 Introduction

Formal verification of cryptographic protocols has been very successful in finding flaws in published cryptographic protocols (see [14, 7] for an overview). Although the general verification problem is undecidable [10, 1, 11], there are important decidable variants [9, 10, 16]. One of them is the insecurity problem of protocols analyzed w.r.t. a bounded number of sessions, in presence of the so-called Dolev-Yao intruder [16, 6, 5, 8]. In this case, one assumes that actions performed by participants during the course of the protocol execution are simple and can be described by single rewrite rules of the form $t \to s$. Such a rule is intended to specify receive-send action of a principal who after receiving a message $t\theta$, for some ground substitution θ, replies $s\theta$. However, in many protocols, participants perform more complicated, recursive computations which cannot be expressed by simple rewrite rules. Examples of protocols of this kind are Internet Key Exchange Protocol (IKE), the Recursive Authentication (RA) protocol [4], and the A-GDH.2 protocol [2]. We will call protocols that involve some kind of iterative or recursive computations *recursive protocols*.

Recently, a tree transducer-based model was proposed for recursive protocols [13, 12]. Tree transducers seem to be a natural choice in the context of recursive cryptographic protocols. The proposed model has, however, the following limitations: (1) rules are assumed to have linear left-hand sides, so no equality tests can be performed,

[⋆] Partially supported by the RNTL project PROUVE-03V360 and by SATIN Project of ACI Sécurité Informatique.

(2) only finite amount of information can be conveyed from one receive-send action to the next ones. Moreover, these assumptions cannot be relaxed without losing decidability. In some cases, these limitations can make modeling of protocols inconvenient or even impossible. For example, the RA protocol, which was chosen in [13] and [12] to illustrate the tree transducer-based protocol model, has rules with non-linear left-hand sides and had to be slightly modified. It should be mentioned that both equality tests for messages of arbitrary size and the possibility of storing arbitrary messages can be easily expressed in the standard term rewriting-based model.

The goal of this paper is to provide a model which can express some recursive computations, without limiting the possibility of compare and store messages. In fact, in many cases the expression power of tree transducers is more than sufficient, so one could ask, whether there is some restricted class of tree transducers which can be used to model protocols, preserving the ability of parties to compare and store messages. One can, however, prove that these assumption cannot be relaxed even, if we consider very weak forms of tree transducers (or any similar formalism) which allow us to model the following basic kinds of computations:

(a) *list mapping* — for an input which is an encoded list $\{[t_1, \dots t_n]\}_k$, produce an encoded list $\{[t'_1, \dots, t'_n]\}_{k'}$, where, for each $i = 1, \dots, n$, the term t'_i is the result of applying some simple rewrite rule to t_i,

(b) *mapping functional symbols* — replace functional symbols of a given term with functional symbols of the same arity, preserving the exact structure of the term (distinct occurrences of a symbol need not be replaced with the same symbol).

The model presented in this paper can express recursive protocols, where participants, in each protocol step, can send a number of messages unbounded w.r.t. the size of the protocol. Each of these messages is the result of applying some simple rewriting rule to some subterm of the messages received so far. So called *selecting theories* are used to determine which rewriting rule should be applied to which terms. Participants are able to store and compare arbitrary messages, like in the case of standard term rewriting-based approach. We assume that keys used in symmetric and public key encryption are constants. Clearly, in our model, one cannot model computations described in the items (a) and (b) above. One can, however, model actions like for instance: for a list $[t_1, \dots, t_n]$ produce and send the list $[t'_1, \dots, t'_n]$, where, for each $i = 1, \dots, n$, the term t'_i is the result of applying some simple rewrite rule to t_i. It is possible, because from the point of view of the Dolev-Yao intruder, the effect of sending $[t'_1, \dots, t'_n]$ is the same as the effect of sending terms $t'_1, \dots, t'_n$ separately. The key fact here is that the result list is not encrypted, which is the case, when protocols like IKE or RA are considered. In the paper, we show how to model the RA protocol in our framework. Because the formalism can express protocols with non-linear left-hand sides of rules, we model this protocol without changes.

We prove that insecurity of protocols with selecting theories with respect to bounded number of sessions decidable in NEXPTIME.

Structure of the paper. Section 2 contains some basic definitions. In Section 3, the model is introduced. It is also showed how to model the RA protocol in the proposed framework. Section 4 contains the proof of the main result of the paper, decidability of protocols with selecting theories.

2 Preliminaries

Let $T(\Sigma, V)$ denote the set of terms over the signature Σ and the set of variables V. A term is *ground*, if it does not contain variables. A (ground) *substitution* is a mapping from variables to (ground) terms, which, in a natural way, is extended to a mapping from term to terms. We denote the set of subterms of t by $\mathit{sub}(t)$.

For a given signature Σ, a *term*-DAG D is a labelled directed acyclic ordered graph such that, if a node v is labelled with a function symbol f of arity n, then it has n ordered immediate successors $v_1, \ldots, v_n$. In such a case we write $v =_D f(v_1, \ldots, v_n)$, and we say that v is a *parent* of v_i (for each $i = 1, \ldots, n$), and v_i is a *child* of v. We define also the notion of *descendant* in the usual way. For a term-DAG D, and a vertex $v =_D f(v_1, \ldots, v_n)$, we recursively define the *term* $t(v, D)$ *represented by* v *in* D by the equation $t(v, D) = f(t(v_1, D), \ldots, t(v_n, D))$. For $s = t(v, D)$, we will write $v \rightrightarrows_D s$, or $v \rightrightarrows s$, if D is known from the context.

Let Σ be a signature, V be a set of variables, and P be a set of unary predicate symbols. If $p \in P$, and $t \in T(\Sigma, V)$, then $p(t)$ is an *atomic formula*. An atomic formula $p(t)$ is *ground*, if t is ground. A *unary Horn theory* is a finite set of *clauses* of the form $a_0 \leftarrow a_1, \ldots, a_n$, where $a_0, \ldots, a_n$ are atomic formulas.

We will use the following notation. Let T be a unary Horn theory, let A, B be sets of ground atomic formulas. We write $A \vdash_T B$, if there exists *a proof of* B *with respect to* T *assuming* A, i.e. a sequence $a_1, \ldots, a_n$ of atomic formulas such that each element of B occurs in $a_1, \ldots, a_n$, and, for each $i = 1, \ldots, n$, we have either (i) $a_i \in A$, or (ii) there exists a clause $b_0 \leftarrow b_1, \ldots, b_m$ in T and a substitution θ such that $a_i = b_0\theta$, and each of $b_1\theta, \ldots, b_m\theta$ occurs in $a_1, \ldots, a_{i-1}$. For a set of atomic formulas A, and an atomic formula a, we write $A \vdash_T a$ for $A \vdash_T \{a\}$.

3 The Formal Model

Protocols with Selecting Theories. *Messages* are ground terms over the signature Σ consisting of constants (*atomic messages* such as principal names, nonces, keys), the unary function symbol $\mathsf{hash}(\cdot)$ (*hashing*), and the following binary function symbols: $\langle\cdot,\cdot\rangle$ (*pairing*), $\{\cdot\}_\cdot$ (*symmetric encryption*), and $\{\!|\cdot|\!\}_\cdot$ (*public key encryption*). We assume that keys used to encrypt messages are constants[1]. We assume that there is a bijection $\cdot^{-1}$ on atomic messages which maps every public (private) key k to its corresponding private (public) key k^{-1}. We assume that Σ contains the constant c_0 known to the intruder and the constant *Sec* (a secret). We will sometimes omit $\langle\cdot,\cdot\rangle$ and write, for instance, $\{t, s\}_k$ instead of $\{\langle t, s\rangle\}_k$.

Let Q and R be disjoint sets of *pop predicate symbols* and *push predicate symbols*, respectively. A *selecting theory* Φ over (Q, R) is a set of clauses of the forms

$$q_1(x_1), \ldots, q_n(x_n) \Rightarrow q(f(x_1, \ldots, x_n)), \tag{1}$$

$$q_1(t), \ldots, q_l(t), r(t) \Rightarrow r'(x) \quad \text{where } x \in \mathit{Var}(t) \tag{2}$$

$$q_1(t), \ldots, q_l(t), r(t) \Rightarrow I(s) \quad \text{where } \mathit{Var}(s) \subseteq \mathit{Var}(t), \tag{3}$$

[1] In the case of the NP-completeness result for non-recursive protocols [16], only keys used in public-key cryptography are assumed to be constants.

where $I \notin Q \cup R$ is a predicate symbol, $q, q_1, \ldots, q_n \in Q$, $r, r' \in R$, $f \in \Sigma$ is a function symbol of arity n, and $x, x_1, \ldots, x_n$ are variables. Clauses of the form (1), called *pop clauses*, have an auxiliary role: they can simulate runs of any finite tree automaton. The information about which states (predicate symbols) can be assigned to a term can be used in (2) and (3), which provides a regular look-ahead. Clauses of the form (2), called *push clauses*, transfer some information (predicate symbols) from a term to its subterms. Clauses of the form (3), called *send clauses*, select terms to be sent (the predicate symbol I means that the term is sent and thus it is known to the intruder).

Let Φ be a selecting theory over (Q, R). For a term t and $r \in R \cup \{I\}$, we define the set of *terms selected by* Φ, $[\![r(t)]\!]_\Phi = \{s \mid r(t) \vdash_\Phi I(s)\}$. A *rule over* (Q, R) has the form $t \rightarrow r(s)$, where t, s are terms and $r \in R \cup \{I\}$. The intended meaning of such a rule is that a principal, after receiving a term $t\theta$, for some ground substitution θ, sends all the terms from the set $[\![r(s\theta)]\!]_\Phi$. Note that the number of terms which are sent in one step of a protocol is not bounded by the size of the protocol, it is only bounded by the size of the message $s\theta$. Because (for any Φ) we have $[\![I(s)]\!]_\Phi = \{s\}$, each simple non-recursive rewrite rule $t \rightarrow s$ can be easily expressed in our formalism by $t \rightarrow I(s)$.

A *principal* Π *over* (Q, R) is a sequence $(t_i \rightarrow r_i(s_i))_{i=1}^n$ of rules over (Q, R) such that, for each $i = 1, \ldots, n$, we have $t_i, s_i \in T(\Sigma, V)$, for a set of variables V, and every variable in s_i occurs in $t_1, \ldots, t_i$. A *protocol over* (Q, R) is a pair (P, Φ), where P is a finite set of principals over (Q, R) and Φ is a selecting theory over (Q, R).

Example. Now, we show how to model the *Recursive Authentication* (RA) protocol [4] in our formalism. This protocol has been analyzed using theorem provers [15, 3]. In [13] and [12] a version of this protocol has been expressed in the tree transducer-based model (the original version has rules with non-linear left hand sides which cannot be expressed in this model). In the presentation of the protocol we follow [13] and [12]. Because, as it was mentioned above, non-recursive receive-send actions can be modeled in our formalism in a straightforward way, we will only describe the only recursive action of the protocol. In this action, the server S receives a sequence of requests of pairs of principals who want to obtain session keys. In response, S generates certificates containing the sessions keys. For instance, suppose that S receives

$$m = h_{K_c}(C, S, N_c, h_{K_b}(B, C, N_b, h_{K_a}(A, B, N_a, -))),$$

where N_a, N_b, N_c are nonces generated by A, B, C, respectively, K_a, K_b, K_c are long-term keys shared between S and A, B, C, and $h_k(m)$ stands for the term $\langle \mathsf{hash}(k, m), m \rangle$. The constant '$-$' marks the end of the sequence of requests. In general, messages sent to S may contain an arbitrary number of requests. In response to m, the server generates two certificates for C: $\{K_{cs}, S, N_c\}_{K_c}$ and $\{K_{bc}, B, N_c\}_{K_c}$, two certificates for B: $\{K_{bc}, C, N_b\}_{K_b}$ and $\{K_{ab}, A, N_b\}_{K_b}$, and one certificate for A: $\{K_{ab}, B, N_a\}_{K_a}$.

So, suppose that $P_0, \ldots, P_n$ are principals, $S = P_n$, and K_i is the long-term key shared by P_i and S. The recursive action of S can be described by the rule $x \rightarrow r(x)$ with the selecting theory over $(\emptyset, \{r\})$ given by the following set of clauses.

$$\begin{aligned}
r\big(h_{K_i}(P_i, P_j, x, y)\big) &\Rightarrow r(y) \\
r\big(h_{K_i}(P_i, P_j, x,\, h_{K_l}(P_l, P_i, x', y))\big) &\Rightarrow I\left(\{K_{ij}, P_j, x\}_{K_i}\right), I\left(\{K_{il}, P_l, x\}_{K_i}\right) \\
r\big(h_{K_i}(P_i, P_j, x,\, -)\big) &\Rightarrow I\left(\{K_{ij}, P_j, x\}_{K_i}\right),
\end{aligned}$$

$$\begin{array}{rll} I(x), I(y) \Rightarrow I(\langle x,y\rangle), & I(x), I(k) \Rightarrow I(\{x\}_k), & (4)\\ I(x) \Rightarrow I(\mathsf{hash}(x)) & I(x), I(k) \Rightarrow I(\{\!|x|\!\}_k), & (5)\\ I(\langle x,y\rangle) \Rightarrow I(x), & I(\{x\}_k), I(k) \Rightarrow I(x), & (6)\\ I(\langle x,y\rangle) \Rightarrow I(y), & I(\{\!|x|\!\}_k), I(k^{-1}) \Rightarrow I(x) \quad \text{(for each key } k\text{)} & (7) \end{array}$$

Fig. 1. T_I — The Intruder Theory.

where the constant K_{ij} is the key for secure communication of P_i and P_j. Note that this theory does not use a regular look-ahead, and uses only one push symbol r.

Attacks. In the Dolev-Yao model [9], the intruder have the entire control over the network. He can intercept and memorize messages, generate new messages and send them to participants with a false identity. We express the ability of the intruder to generate (derive) new messages from a given set of messages by the theory T_I in Figure 1, where the predicate symbol I is intended to describe the intruder knowledge. For a set A of messages, let $I(A) = \{I(t) \mid t \in A\}$. We will say that the intruder can *derive a message t from messages A*, if $I(A) \vdash_{T_I} I(t)$.

Now, we give a definition of an *attack for a bounded number of sessions*. In an attack, the intruder nondeterministically chooses an execution order for the protocol steps and then produces input messages for the protocol rules. These input messages have to be derived from the intruder's initial knowledge and the output messages obtained so far. The aim of the intruder is to derive the secret message *Sec*. If some number of interleaving sessions of a protocol is to be analyzed, then these sessions have to be encoded into the protocol, which is the standard approach when protocols are analyzed w.r.t. a bounded number of sessions (see, for instance [16, 6]).

Formally, given a protocol $(\{\Pi_1, \ldots, \Pi_l\}, \Phi)$, a *protocol execution scheme* is a sequence of rules $\pi = \pi_1, \ldots, \pi_n$ such that each element of π can be assigned to one of the participants $\Pi_1, \ldots, \Pi_l$, and, for each participant Π_k ($k = 1, \ldots, l$), the subsequence of the elements of π assigned to Π_k is $\Pi_k^1, \ldots, \Pi_k^m$, for some $m \leq |\Pi_k|$, where Π_k^i is the i-th rule of Π_k.[2] An *attack* is a pair (π, σ), where $\pi = (t_i \to r_i(s_i))_{i=1}^n$ is a protocol execution scheme, and σ is a ground substitution such that

$$I(c_0), I([\![r_1(s_1\sigma)]\!]_\Phi), \ldots, I([\![r_{i-1}(s_{i-1}\sigma)]\!]_\Phi) \vdash_{T_I} I(t_i\sigma), \quad \text{for all } i = 1, \ldots, n \tag{8}$$

$$I(c_0), I([\![r_1(s_1\sigma)]\!]_\Phi), \ldots, I([\![r_n(s_n\sigma)]\!]_\Phi) \vdash_{T_I} I(\mathit{Sec}). \tag{9}$$

Recall that c_0 is the only constant initially known to the intruder[3]. A protocol is *insecure*, if there exists an attack on it.

We end this section with the following, easy to prove lemma.

Lemma 1. *$A \vdash_{T_I} B$ iff there exists a proof of B with respect to T_I assuming A such that all the facts obtained by rules* (6), (7) *are before the facts obtained by rules* (4), (5).

[2] More formally, a sequence $\pi_1, \ldots, \pi_n$ of rules is a protocol execution scheme, if there is a function $f : \{1, \ldots, n\} \to \{1, \ldots, l\}$ such that, for each $k = 1, \ldots, l$, assuming that integers $i_1 < \cdots < i_m$ are all the elements of $f^{-1}(k)$, we have $\pi_{i_j} = \Pi_k^j$, for each $j = 1, \ldots, m$.

[3] If we want to consider an initial knowledge of the intruder given by a finite set $\{t_1, \ldots, t_m\}$, we can add a principal with the rule $c_0 \to I(\langle t_1, \ldots, t_m\rangle)$.

$$I(x), I(y) \Rightarrow I(\langle x, y\rangle), \qquad I(x), I(k) \Rightarrow I(\{x\}_k), \tag{10}$$

$$I(x) \Rightarrow I(\mathsf{hash}(x)) \qquad I(x), I(k) \Rightarrow I(\{\!|x|\!\}_k), \tag{11}$$

$$r_I(x) \Rightarrow I(x), \tag{12}$$

$$r_I(\langle x, y\rangle) \Rightarrow r_I(x), \qquad r_I(\{x\}_k), I(k) \Rightarrow r_I(x), \tag{13}$$

$$r_I(\langle x, y\rangle) \Rightarrow r_I(y), \qquad r_I(\{\!|x|\!\}_k), I(k^{-1}) \Rightarrow r_I(x) \quad \text{(for each key } k\text{)} \tag{14}$$

$$\varphi, \qquad \text{for each pop or push rule } \varphi \text{ of } \Phi \tag{15}$$

$$I(k_1), \dots, I(k_n), q_1(t), \dots, q_l(t), r(t) \Rightarrow p(s'), \tag{16}$$

for each send rule $q_1(t), \dots, q_l(t), r(t) \Rightarrow I(s)$ of Φ, for each $s'/_K \in \mathit{Acc}(s)$ with $K = \{k_1, \dots, k_n\}$, where $p = I$, if s' is not a variable, and $p = r_I$, otherwise.

Fig. 2. Φ_I — the theory of the protocol (P, Φ).

4 Main Result

Theorem 1. *Insecurity of protocols with selecting theories w.r.t. a bounded number of sessions is decidable in nondeterministic exponential time.*

The remainder of this section is devoted to prove Theorem 1. In Subsections 4.1 and 4.2, the existence of an attack is expressed in a way which is more appropriate for the rest of the proof. In Subsection 4.3 we introduce the key notion of ADAG. ADAGs are labelled term-DAGs which can represent attacks. We show how to minimize ADAG, so that, if an ADAG exists, then there exists an ADAG of an exponential size, which gives rise to the nondeterministic exponential time algorithm for the insecurity problem.

4.1 The Theory of a Protocol

In this section we express the existence of an attack in a more uniform way, without using expressions of the form $[\![r(s)]\!]_\Phi$. We use here the fact that both selecting theories and the intruder theory are unary horn theories. Moreover, Lemma 1 allows us to extend selecting theories in such a way that the clauses (6) and (7) of T_I are not necessary.

In the following, $\mathit{Acc}(t)$ denotes the set of elements of the form $s/_K$, where s is a subterm of t and K is a minimal set of keys sufficient to access s providing t is known. For example, if $t = \{c, \{d\}_b\}_a$, then $\mathit{Acc}(t) = \{t/_\emptyset, c/_{\{a\}}, \{d\}_b/_{\{a\}}, d/_{\{a,b\}}\}$. Formally, we define Acc by the equations $\mathit{Acc}(\langle t_1, t_2\rangle) = \{\langle t_1, t_2\rangle/_\emptyset\} \cup \mathit{Acc}(t_1) \cup \mathit{Acc}(t_2)$, $\mathit{Acc}(\{t\}_k) = \{\{t\}_k/_\emptyset\} \cup \{s/_{\{k\}\cup K} \mid s/_K \in \mathit{Acc}(t))\}$, and $\mathit{Acc}(\{\!|t|\!\}_k) = \{\{\!|t|\!\}_k/_\emptyset\} \cup \{s/_{\{k^{-1}\}\cup K} \mid s/_K \in \mathit{Acc}(t))\}$. Note that $t/_\emptyset \in \mathit{Acc}(t)$, for each term t.

Definition 1. Let (P, Φ) be a protocol over (Q, R). Let r_I be a fresh predicate symbol. *The theory Φ_I of the protocol P* consists of the rules given in Fig. 2.

Note that the theory Φ_I consists of rules of three types: (a) rules (10) and (11), called the *intruder pop rules*, (b) pop rules, (c) rules of the form $I(k_1), \dots, I(k_n), q_1(t), \dots, q_l(t), r(t) \Rightarrow r'(x)$, called *generalized push rules*, and (d) rules of the form $I(k_1),\dots,I(k_n),q_1(t),\dots,q_l(t),r(t) \Rightarrow I(s)$, called *generalized send rules*. Note also that Φ_I contains all the rules of Φ. By Lemma 1, rules (12)–(14) and (16) can simulate the intruder rules (6) and (7). Thus, one can prove the following characterization of the existence of an attack.

Lemma 2. *Let (P,Φ) be a protocol over (Q,R), let $\pi = (t_i \to r_i(s_i))_{i=1}^n$ be a protocol execution scheme for P and σ be a substitution. The pair (π,σ) is an attack iff we have*

$$I(c_0),\hat{r}_1(s_1\sigma),\dots,\hat{r}_{i-1}(s_{i-1}\sigma) \vdash_{\Phi_I} I(t_i\sigma), \quad \text{for all } i=1,\dots,n \tag{17}$$

$$I(c_0),\hat{r}_1(s_1\sigma),\dots,\hat{r}_n(s_n) \vdash_{\Phi_I} I(\mathit{Sec}), \tag{18}$$

where, for each $i=1,\dots,n$, we put $\hat{r}_i = r_I$, if $r_i = I$, and $\hat{r}_i = r_i$, otherwise.

4.2 Stage Theories

In this subsection, we express the existence of an attack using *a stage theory of a protocol*. In this theory, instead of representing the knowledge of the intruder by the predicate symbol I, the family of predicate symbols $I^{(0)},\dots,I^{(m)}$ is used to represent his knowledge at different stages of an attack.

Let (P,Φ) be a protocol over (Q,R) and $\pi = (t_i \to r_i(s_i))_{i=1}^n$ be a protocol execution scheme. Let $\mathcal{K}$ be the set containing the constant *Sec* and all the keys of P. A sequence $\boldsymbol{e} = e_1,\dots,e_m$ of elements of $\mathcal{K} \cup \{1,\dots,n\}$ is called a *stage sequence for π*, if $\boldsymbol{e}$ contains all the elements $\mathit{Sec}, 1,\dots,n$, and whenever $e_i = k$ and $e_j = l$, for $i<j$, then $k<l$. A stage sequence represents key elements of the intruder knowledge at consecutive stages of an attack. An element e_i of such a sequence either represents a new key that can be used by the intruder at the i-th stage (if e_i is a key), or, if $e_i = j$, it express progress in the protocol execution, and it means that at the i-th stage the j-th step of the protocol has been executed, so the intruder can use terms from $[\![r_j(s_j\sigma)]\!]_\Phi$.

Let $\mathcal{K}_i = \{a \in \mathcal{K} \mid a = e_j \text{ for some } j \le i\}$. The *stage theory for Φ and $\boldsymbol{e}$*, denoted by $\Phi_{\boldsymbol{e}}$, is given in Figure 3, where $p^{(i)}$, for $i=0,\dots,m$, and $p \in R \cup \{r_I, I\}$, are fresh predicate symbols. The predicate symbol $I^{(k)}$ is intended to describe the intruder knowledge at the k-th stage of an attack. The intended meaning of $r^{(k)}(t)$ is that the intruder is able to prove $r(t)$ at the k-th stage.

Lemma 3. *Let $\pi = (t_i \to r_i(s_i))_{i=1}^n$ be a protocol execution scheme and σ be a ground substitution. The pair (π,σ) is an attack iff there is a stage sequence $\boldsymbol{e} = e_1,\dots,e_m$ for π such that*

$$I^{(0)}(c_0),\psi_1,\dots,\psi_m \vdash_{\Phi_{\boldsymbol{e}}} \varphi_1,\dots,\varphi_m, \tag{23}$$

where $\varphi_1,\dots,\varphi_m$ and $\psi_1,\dots,\psi_m$ are defined as follows. If $e_i = j \in \{1,\dots,n\}$, then $\varphi_i = I^{(i-1)}(t_j\sigma)$, and $\psi_i = \hat{r}_j^{(i)}(s_j\sigma)$, where $\hat{r}$ is defined like in Lemma 2. If $e_i = a \in \mathcal{K}$, then $\varphi_i = I^{(i-1)}(a)$ and $\psi_i = I^{(0)}(c_0)$.

Proof. First, suppose that (23) holds, for some π, $\boldsymbol{e}$, and σ, and that Γ is a proof of it. Let Γ_0 denotes the subsequence of Γ containing only facts of the form $q(t)$, for $q \in Q$.

$$q_1(x_1), \dots, q_n(x_n) \Rightarrow q(f(x_1, \dots, x_n)), \tag{19}$$

for each pop rule $q_1(x_1), \dots, q_n(x_n) \Rightarrow q(f(x_1, \dots, x_n))$ of Φ_I,

$$q_1(t), \dots, q_l(t), r^{(j)}(t) \Rightarrow p^{(i)}(s), \tag{20}$$

for each (generalized) push or send rule $I(k_1), \dots, I(k_m), q_1(t), \dots, q_l(t), r(t) \Rightarrow p(s)$ of Φ_I, for $i \geq j$, and $k_1, \dots, k_m \in \mathcal{K}_i$,

$$I^{(j)}(x), I^{(k)}(y) \Rightarrow I^{(i)}(\langle x, y \rangle) \quad I^{(j)}(x) \Rightarrow I^{(i)}(\mathsf{hash}(x)) \quad \text{if } i \geq j, k \tag{21}$$

$$I^{(j)}(x) \Rightarrow I^{(i)}(\{x\}_a), \quad I^{(j)}(x) \Rightarrow I^{(i)}(\{\!|x|\!\}_a) \quad \text{if } i \geq j, \text{ and } a \in \mathcal{K}_i. \tag{22}$$

Fig. 3. $\Phi_{\boldsymbol{e}}$ — The Stage Theory for Φ and $\boldsymbol{e}$.

Let Γ_i denotes the subsequence of Γ containing only facts of the form $p^{(i)}(t)$, and let $\Gamma_{\leq i}$ be the concatenation of $\Gamma_0, \dots, \Gamma_i$. Let $\Gamma^*_{\leq i}$ be the sequence obtained from $\Gamma_{\leq i}$ by substituting each $p^{(k)}$ by p. One can show that $\Gamma^*_{\leq i-1}$ is a proof of (17), and $\Gamma^*_{\leq m}$ is a proof of (18). Hence, (π, σ) is an attack.

Now, suppose that we have an attack (π, σ). By Lemma 2, (17) and (18) hold. So, let Π_i be a proof of (17), for $i = 1, \dots, n$, and let Π_{n+1} be a proof of (18). We split each Π_k (for $k = 1, \dots, (n+1)$) into the maximal (w.r.t. its length) sequence $\Pi_k^1, \dots, \Pi_k^{m_k}$ such that the last element of Π_k^i, for $1 \leq i < m_k$, is of the form $I(a)$, for $a \in \mathcal{K}$, and this occurrence of $I(a)$ is the only one in $\Pi_1, \dots, \Pi_{k-1}, \Pi_k^1, \dots, \Pi_k^i$. We want to re-index the obtained sequence of Π_k^i, so let $\hat{\Pi}_1, \dots, \hat{\Pi}_N = \Pi_1^1, \dots, \Pi_1^{m_1}, \dots, \Pi_{n+1}^1, \dots, \Pi_{n+1}^{m_{n+1}}$.

For $i = 1, \dots, N$, let Γ_i be the sequence of facts obtained from $\hat{\Pi}_i$ by substituting each $p(t)$, for $p \in R \cup \{r_I, I\}$, by $p^{(i-1)}(t)$, and let e_i be equal to k, if $\hat{\Pi}_i = \Pi_k^{m_k}$, for some k, and, otherwise, let e_i be a, where $I(a)$ is the last element of $\hat{\Pi}_i$. One can prove that the concatenation of $\Gamma_1, \dots, \Gamma_n$ is a proof of (23). □

We say that a fact $I^{(i)}(t)$ is *stronger than* $I^{(j)}(t)$, if $i \leq j$. A proof is *normal*, if for each term t, it contains at most one fact of the form $I^{(i)}$. The following lemma is easy to prove.

Lemma 4. *It holds* (23) *iff there is a normal proof of*

$$I^{(0)}(c), \psi_1, \dots, \psi_m \vdash_{\Phi_{\boldsymbol{e}}} \varphi'_1, \dots, \varphi'_m, \tag{24}$$

where, for each $k = 1, \dots, m$, the fact φ'_k is stronger than φ_k.

4.3 ADAGs

This section is the central part of the proof of Theorem 1. We give here the definition of an ADAG and link the existence of ADAGs with the existence of attacks (Lemma 5). Next, we show that if there exists an ADAG which represents an attack on a protocols, then there exists an ADAG of exponential size. Finally, as a consequence of the above, we obtain an NEXPTIME algorithm for deciding insecurity of protocols.

We will assume that selecting theories have the following property: the push rules are *flat*, i.e. are of the form (2) with $t = f(x_1, \ldots, x_n)$, where $x_1, \ldots, x_n$ are variables. We can do it without loss of generality, because, for any selecting theory, one can easily obtain an equivalent selecting theory with this property.

Definition 2. Let D be a term-DAG over Σ with the set V of vertices, and let T be a set of terms over Σ and $\mathbb{V}$. A function $\theta : \mathit{sub}(T) \to V$ is a *D-embedding for T*, if $\theta(f(t_1, \ldots, t_n)) = v$ implies that $v =_D f(v_1, \ldots, v_n)$ and $\theta(t_i) = v_i$, for $i = 1, \ldots, n$. Embeddings θ_1 and θ_2 are *compatible*, if for each variable x which is in the domain of both θ_1 and θ_2, we have $\theta_1(x) = \theta_2(x)$.

Let $v \in V$, and $t \in T(\Sigma, \mathbb{V})$. By $\mathrm{emb}(t \mapsto v)$ we denote the unique embedding θ for $\{t\}$ such that $\theta(t) = v$ (if it exists). Let $v_1, v_2 \in V$, and $t_1, t_2 \in T(\Sigma, \mathbb{V})$. The terms (t_1, t_2) *embeds to* (v_1, v_2), if the embeddings $\mathrm{emb}(t_1 \mapsto v_1)$ and $\mathrm{emb}(t_2 \mapsto v_2)$ exist and are compatible.

Definition 3. Let Φ and Ψ be stage theories over (Q, R). The theory Ψ is an *instance of Φ*, if each clause in Ψ is an instance of a clause in Φ.

Definition 4. Let (P, Φ) be a protocol over (Q, R), let $\pi = (t_i \to r_i(s_i))_{i=1}^n$ be a protocol execution scheme, and $\boldsymbol{e} = e_1, \ldots, e_m$ be a stage sequence for π. Let $\mathcal{T}_P$ denote the set $\{t_i, s_i\}_{i=1}^n \cup \{c_0\} \cup \mathcal{K}$, and $Q_{\boldsymbol{e}}$ denote the set of predicate symbols of $\Phi_{\boldsymbol{e}}$.

A DAG *of the attack* (an ADAG for short) for $(\Phi, \pi, \boldsymbol{e})$ is a tuple $\mathcal{D} = \langle D, \alpha, \beta, \Psi, \delta \rangle$ where D is a term-DAG over Σ with the set of vertices V, $\delta : V \to 2^{Q_{\boldsymbol{e}}}$, α is a D-embedding for $\mathcal{T}_P$, a stage theory Ψ is an instance of Φ, and β is a partial function from $V \times Q_{\boldsymbol{e}}$ to $V \times \Psi_{\boldsymbol{e}}$, called a *witness function*, such that

(i) if $v = \alpha(t_j)$, then $I^{(i')} \in \delta(v)$, for some $i' < i$, where i is the integer such that $e_i = j$,

(ii) for each vertex v, the set $\delta(v)$ contains at most one element of the form $I^{(i)}$,

(iii) if $p \in \delta(v)$ then one of the following conditions holds:

(a) $v = \alpha(c_0)$ and $p = I^{(l)}$ (for some l), or $v = \alpha(s_j)$ and $p = \hat{r}_j^{(i)}$, for some i, j such that $e_i = j$,

(b) $v =_D f(v_1, \ldots, v_n)$, and $\Psi_{\boldsymbol{e}}$ contains the clause $p_1(x_1), \ldots, p_n(x_n) \Rightarrow p(f(x_1, \ldots, x_n))$, for some $p_1 \in \delta(v_1), \ldots, p_n \in \delta(v_n)$,

(c) $\beta(v, p) = (v', \varphi)$, where $\varphi = \big(p_1(t), \ldots, p_l(t) \Rightarrow p(x_i)\big)$, for $t = f(x_1, \ldots, x_j)$, is a push clause of $\Psi_{\boldsymbol{e}}$, $\{p_1, \ldots, p_l\} \subseteq \delta(v')$, and $v' =_D f(v_1, \ldots, v_n)$ with $v_i = v$, or

(d) $\beta(v, p) = (v', \varphi)$, where $\varphi = \big(p_1(t'), \ldots, p_l(t') \Rightarrow p(t)\big)$ is a send clause of $\Psi_{\boldsymbol{e}}$ (so $p = I^{(j)}$), $\{p_1, \ldots, p_l\} \subseteq \delta(v')$, and (t, t') embeds to (v, v').

Lemma 5. *If there is an attack (π, σ) on a protocol (P, Φ) then there is an* ADAG *$\langle D, \alpha, \beta, \Psi, \delta \rangle$ for $(\Phi, \pi, \boldsymbol{e})$, for some stage sequence $\boldsymbol{e}$ for π, such that $\Psi = \Phi$. If there exists an* ADAG *for $(\Phi, \pi, \boldsymbol{e})$ then there exists an attack (π, σ), for some substitution σ.*

Proof. Suppose that there is an attack (π, σ). By Lemma 3 and Lemma 4, there is a sequence $\boldsymbol{e}$ and a normal proof Γ of (24). Let D be the DAG representing all the terms of the form $t\sigma$, where $t \in \mathcal{T}_P$. For $t \in \mathcal{T}_P$, let $\alpha(t)$ be the vertex v such that $v \rightrightarrows t\sigma$. For a vertex v of D, let $\delta(v)$ be the set of the predicate symbols $p \in Q_{\boldsymbol{e}}$

such that $p(t_v)$ occurs in Γ, for $v \rightrightarrows t_v$. Further, if we have $p(t_v)$ in Γ, because $\varphi = \big(q_1(s'), \ldots, q_l(s'), p'(s') \Rightarrow p(s)\big)$ is a push or send clause of $\Phi_{\boldsymbol{e}}$, $t_v = s\sigma$, for some substitution σ, and $q_1(s'\sigma), \ldots, q_l(s'\sigma), p'(s'\sigma)$ occur in Γ before $p(t_v)$, then let $\beta(v,p) = (v', \varphi)$, where v' is the vertex of D such that $v' \rightrightarrows s'\sigma$ (such a vertex exists, because $s'\sigma$ has to be a subterm of some $s_i\sigma$). One can show that $\langle D, \alpha, \beta, \Phi, \delta\rangle$ is an ADAG.

Now, suppose that $\langle D, \alpha, \beta, \Psi, \delta\rangle$ is an ADAG for $(\Phi, \pi, \boldsymbol{e})$. Let $\sigma(x) = t$, where t is the term such that $\alpha(x) \rightrightarrows t$. We produce the following sequence of facts: First, we put all the facts of the form $q(t)$, where $v \rightrightarrows t$ and $q \in \delta(v)$, for $q \in Q$, in such a way that $q(t)$ is before $q'(t')$, if $t < t'$. Second, we put all the fact of the form $r^{(i)}(t)$, where $v \rightrightarrows t$ and $r^{(i)} \in \delta(v)$, for $r \in R \cup \{r_I\}$, in such a way that $p(t)$ is before $p'(t')$, if $t > t'$. Finally, we put all the fact of the form $I^{(i)}(t)$, where $v \rightrightarrows t$ and $I^{(i)} \in \delta(v)$, in such a way that $p(t)$ is before $p'(t')$, if $t < t'$. One can prove that this sequence is a normal proof of (24) (note that Ψ is an instance of Φ, so each clause of $\Psi_{\boldsymbol{e}}$ is an instance of a clause of $\Phi_{\boldsymbol{e}}$), which by Lemma 3 and Lemma 4, implies that there exists an attack. □

Lemma 5 is a crucial step of our construction, because it characterizes the existence of an attack by a structure which is defined by some local properties. Now, we will describe how to minimize ADAGs, roughly speaking, by merging vertices which are indistinguishable from the point of view of this local properties. We proceed in three steps given by Lemmas 6, 7, and 8 below (proofs of these lemmas are given in the separate sections). To formulate these lemmas we need the following definitions.

Let (P, Φ) be a protocol, and let $\mathcal{D} = \langle D, \alpha, \beta, \Psi, \delta\rangle$ be an ADAG for $(\Phi, \pi, \boldsymbol{e})$. A vertex v of $\mathcal{D}$ is *bounded*, if $v = \alpha(t)$, for some $t \in \mathit{sub}(\mathcal{T}_P)$. Otherwise, v is *free*. Let $\mathcal{B}(\mathcal{D})$ be the set of vertices which can be reached from bounded vertices, moving from a vertex to its child, in less than $|P| \cdot |\Phi|$ steps. Note that $\mathcal{B}(\mathcal{D})$ is exponentially bounded with respect to the size of the protocol.

A *goal* is a vertex v with $I^{(i)} \in \delta(v)$, for some i, such that the item (iii,d) of Definition 4 holds for v and $p = I^{(i)}$. Let $G(\mathcal{D})$ be the set of goals of $\mathcal{D}$. For a stage sequence $\boldsymbol{e}$, let $G_k(\mathcal{D}) = \{v \mid v \in G(\mathcal{D})$, and $I^{(i)} \in \delta(v)$ for $\boldsymbol{e}^{-1}(k) \leq i < \boldsymbol{e}^{-1}(k+1)\}$, where $\boldsymbol{e}^{-1}(0) = 0$, $\boldsymbol{e}^{-1}(n+1) = \infty$, and, for $k = 1, \ldots, n$, let $\boldsymbol{e}^{-1}(k)$ be the integer i such that $e_i = k$. Let $G_{>k}(\mathcal{D}) = \bigcup_{i>k} G_i(\mathcal{D})$.

An ADAG $\mathcal{D}$ is *simple*, if, whenever $u \notin \mathcal{B}(\mathcal{D})$ is a descendant of $v \in G_i(\mathcal{D})$, then $u \notin G_{>i}(\mathcal{D})$. Let $\hat{\Phi} = \Phi \cup \{C' \mid C'$ is an instance of a send clause $C \in \Phi$ of the form $(\ldots \Rightarrow I^{(i)}(s))$, and the depth of C' is not greater than $|P| \cdot i\}$.

Lemma 6. *Let (P, Φ) be a protocol. If $\mathcal{D} = \langle D, \alpha, \beta, \Phi, \delta\rangle$ is an* ADAG *for $(\Phi, \pi, \boldsymbol{e})$, then there exists a simple* ADAG *$\mathcal{D}' = \langle D', \alpha', \beta', \hat{\Phi}, \delta'\rangle$ for $(\Phi, \pi, \boldsymbol{e})$.*

Lemma 6 states that each ADAG can be transformed to a simple ADAG. Having a simple ADAG, we can minimize the number of its goals, which is expressed by the following lemma. It allows us to minimize the size of the whole ADAG, as is stated in Lemma 8.

Lemma 7. *Let (P, Φ) be a protocol. If $\mathcal{D} = \langle D, \alpha, \beta, \hat{\Phi}, \delta\rangle$ is a simple* ADAG *for $(\Phi, \pi, \boldsymbol{e})$, then there exists an* ADAG *$\mathcal{D}' = \langle D', \alpha', \beta', \hat{\Phi}, \delta'\rangle$ such that the set of goals of $\mathcal{D}'$ is exponentially bounded w.r.t. the size of (P, Φ).*

Lemma 8. *Let $(P, \mathit{\Phi})$ be a protocol over (Q, R). If $\mathcal{D}_0 = \langle D, \alpha, \beta, \hat{\mathit{\Phi}}, \delta \rangle$ (for some D, α, β, δ) is an* ADAG *for $(\mathit{\Phi}, \pi, \boldsymbol{e})$ with an exponentially bounded set of goals (w.r.t. the size of $(P, \mathit{\Phi})$), then there is an* ADAG *for $(\mathit{\Phi}, \pi, \boldsymbol{e})$ of an exponentially bounded size.*

Lemmas 5, 6, 7, and 8 have the following consequence.

Corollary 1. *Let $(P, \mathit{\Phi})$ be a protocol, and let π be a protocol execution scheme. There is an attack (π, σ), for some σ, iff there exists an* ADAG *for $(\mathit{\Phi}, \pi, \boldsymbol{e})$, for some $\boldsymbol{e}$, of an exponential size w.r.t. the size of the protocol.*

The Algorithm. To decide insecurity of a given protocol $(P, \mathit{\Phi})$, we guess an attack skeleton π, a stage sequence $\boldsymbol{e}$, and an ADAG for $(\mathit{\Phi}, \pi, \boldsymbol{e})$ of exponential size w.r.t. the size of the protocol. Correctness of this algorithm is given by the Corollary 1. The algorithm works in NEXPTIME, which concludes the proof of Theorem 1. □

An easy to obtain lower bound is DEXPTIME, because the problem of the emptiness of the intersection of regular tree languages, which is DEXPTIME-hard, can be easily reduced to the problem of deciding protocols with selecting theories (in the reduction, pop-clauses of selecting theories are used).

4.4 Proof of Lemma 6

We start this section with technical definitions used in this section and in the following ones. For an ADAG $\mathcal{D}$, let $S^i_{\mathcal{D}}$ denote the set of descendants of $\alpha(c_0), \alpha(t_j), \alpha(s_j)$, for $j \leq i$. For a goal u, we define sets of vertices $B^u_{\mathcal{D}}$ and $F^u_{\mathcal{D}}$ in the following way. Let $\beta(u, I^{(i)}) = (u', \varphi)$, with $\varphi = \big(q_1(t'), \ldots, q_l(t'), r(t') \Rightarrow I^{(i)}(t)\big)$, $\theta = \mathrm{emb}(t \mapsto u)$, and $\theta' = \mathrm{emb}(t' \mapsto u')$. $B^u_{\mathcal{D}} = \{\theta(s) \mid s \text{ is a subterm of } t \text{ or } t'\}$. $F^u_{\mathcal{D}} = \{\theta(x) \mid x \in \mathrm{dom}(\theta) \cap \mathrm{dom}(\theta')\}$ (note that θ and θ' are compatible, so $\theta(x) = \theta'(x)$).

We write $(v', p') \overset{\mathcal{D}}{\leadsto} (v, p)$, if $\beta(v, p) = (v', \varphi)$, for $\varphi = \big(q_1(t'), \ldots, q_l(t'), p'(t') \Rightarrow p(t)\big)$. Let $\overset{\mathcal{D}}{\leadsto}{}^*$ denotes the transitive closure of $\overset{\mathcal{D}}{\leadsto}$. If u is a goal and $I^{(i)} \in \delta(u)$, then we can write $(v, p) \overset{\mathcal{D}}{\leadsto}{}^* u$ instead of $(v, p) \overset{\mathcal{D}}{\leadsto}{}^* (u, I^{(i)})$, and $v \overset{\mathcal{D}}{\leadsto}{}^* u$, if, for some p', we have $(v, p') \overset{\mathcal{D}}{\leadsto}{}^* (u, I^{(i)})$.

In order to prove Lemma 6, we construct a sequence $\mathcal{D}_0, \ldots, \mathcal{D}_n = D'$ of ADAGs such that $\mathcal{D}_0 = \langle D, \alpha, \beta, \hat{\mathit{\Psi}}, \delta \rangle$ and, for each $\mathcal{D}_i$ $(i = 0, \ldots, n)$, we have

$(*)$ if $u \notin \mathcal{B}(\mathcal{D}_i)$ is a descendant of $v \in G_j(\mathcal{D}_i)$, for $j = 1, \ldots, i$, then $u \notin G_{>j}(\mathcal{D}_i)$, and

$(**)$ if $u \in G_{>i}(\mathcal{D}_i)$ and $\beta(u) = (u', \varphi)$, then either $\varphi \in \mathit{\Phi}$, or $F^u_{\mathcal{D}_i} \subseteq S^i_{\mathcal{D}_i}$.

It is easy to show that $\mathcal{D}_0$ is an ADAG for $(\mathit{\Phi}, \pi, \boldsymbol{e})$ and $(*)$, $(**)$ hold for $\mathcal{D}_0$. Now, assume that $(*)$ and $(**)$ hold for $\mathcal{D}_{i-1} = \langle D_{i-1}, \alpha, \beta_{i-1}, \hat{\mathit{\Phi}}, \delta_{i-1} \rangle$. We will construct $\mathcal{D}_i = \langle D_i, \alpha, \beta_i, \hat{\mathit{\Phi}}, \delta_i \rangle$. Let V_{i-1} and V_i denote the sets of vertices of D_{i-1} and D_i, respectively. Let $A = \{u \mid u \text{ is a descendant of some } u' \in G_i(\mathcal{D}_{i-1}),\ u \notin G_i(\mathcal{D}_{i-1}),\ u \notin S^i_{\mathcal{D}_{i-1}}\}$. Let X be the least set of vertices of $\mathcal{D}_{i-1}$ such that (i) if $u \in A$ and u is bounded, then $u \in X$, (ii) if $u \in X$ and $u' \in A$ is a child of u, then $u' \in X$.

The construction of D_i. Let $V_i = V_{i-1} \cup W_i$, where W_i is the set of fresh vertices of the form $\hat{v}$, for $v \in A$. Now, suppose that $v =_{D_{i-1}} f(v_1, \ldots, v_n)$. For each $i = 1, \ldots, n$, we define $h(v, i)$ as follows. If $v \notin A$, $v_i \in A$, $v \notin G_{\leq i}(\mathcal{D}_{i-1})$, and v or v_i is free, then $h(v, i) = \hat{v}_i$. Otherwise, $h(v, i) = v_i$. We put $v =_{D_i} f(h(v, 1), \ldots, h(v, n))$. For $v \in A$ with $v =_{D_{i-1}} f(v_1, \ldots, v_n)$ we put $\hat{v} =_{D_i} f(v'_1, \ldots, v'_n)$, where, for each $i = 1, \ldots, n$, $v'_i = \hat{v}_i$, if $v_i \in A$, and $v'_i = v_i$, otherwise. Note that $S^i_{\mathcal{D}_{i-1}} = S^i_{\mathcal{D}_i}$.

The construction of δ_i. For $v \in A$ we define the set $R(u) \subseteq R \cup \{r_I\}$ by the following equivalence: $r \in R(u)$ iff there exist vertices $w \notin A$ and $v \in A$ such that $h(w, k) = \hat{v}$, for some k, and $(w, r'') \overset{\mathcal{D}_{i-1}}{\rightsquigarrow}{}^1 (v, r') \overset{\mathcal{D}_{i-1}}{\rightsquigarrow}{}^{1*} (u, r)$, for some r', r''. For $v \notin A$, let $\delta_i(v) = \delta_{i-1}(v)$. For $v \in A$, we define $\delta_i(v)$ and $\delta_i(\hat{v})$ as follows: $\delta_Q = \{q \in Q \mid q \in \delta_{i-1}(v)\}$, $\delta_i(v) = \delta_Q \cup \{r \in R \mid r \in \delta_{i-1}(v), r \notin R(v)\} \cup \{I^{(j)} \mid I^{(j)} \in \delta_{i-1}(v),$ and either $j \leq i$, or $v \in X\}$, and $\delta_i(\hat{v}) = \delta_Q \cup \{r \in R \mid r \in \delta_{i-1}(v), r \in R(v)\} \cup \{I^{(j)} \mid I^{(j)} \in \delta_{i-1}(v)\}$. It is easy to check that $\delta_{i-1}(v) = \delta_i(v) \cup \delta_i(\hat{v})$.

The construction of β_i. If v is a vertex of $\mathcal{D}_{i-1}$ and $r \in R \cup \{r_I\}$, $r \in \delta_i(v)$, then let $\beta_i(v, r) = \beta_{i-1}(v, r)$. If $v \in A$ and $r \in R \cup \{r_I\}$, $r \in \delta_i(\hat{v})$, then let $\beta_i(\hat{v}, r) = (w, r')$, where $\beta_{i-1}(v, r) = (u, r')$, and $w = u$, if $u \notin A$, and $w = \hat{u}$, otherwise.

For $v \in V_i$, let us define $\check{v} \in V_{i-1}$ as follows: $\check{v} = v$, if $v \in V_{i-1}$, and $\check{v} = u$, if $v = \hat{u}$. For $v \in V_{i-1}$, and $r \in \delta_{i-1}(v)$, we define $g(v, r) \in V_i$ as follows: $g(v, r) = v$, if $r \in \delta_i(v)$, and $g(v, r) = \hat{v}$, otherwise.

Let $v \in V_i$ with $I^{(j)} \in \delta_i(v)$. We will define $\beta_i(v, I^{(j)})$. Let $(v', \varphi) = \beta_{i-1}(\check{v}, I^{(j)})$, with $\varphi = \big(q_1(t'), \ldots, q_l(t'), r(t') \Rightarrow I^{(j)}(t)\big)$. Let $w = g(v', r)$. Note that, because $r \in \delta_i(u)$, we have $r \in \delta_{i+1}(w)$. Since, by the inductive hypothesis, $(**)$ holds for $\mathcal{D}_{i-1}$, it is enough to consider two cases:

1. $B^{\check{v}}_{\mathcal{D}_{i-1}} \cap A = \emptyset$, or $F^{\check{v}}_{\mathcal{D}_{i-1}} \subseteq S^i_{\mathcal{D}_i}$. In this case, let $\beta_i(v, I^{(j)}) = (w, \varphi)$.
2. $B^{\check{v}}_{\mathcal{D}_{i-1}} \cap A \neq \emptyset$, and $\varphi \in \Phi$. In this case we proceed as follows. Let $\theta_{\check{v}} = \mathrm{emb}(t \mapsto \check{v})$. We define a substitution σ with the domain $\mathrm{dom}(\sigma) = \{x \mid x \in \mathit{Var}(t),\ \theta_{\check{v}}(x) \in A\}$ as follows. Let $x \in \mathrm{dom}(\sigma)$. Let u be an (arbitrarily chosen) vertex in $G_i(\mathcal{D}_{i-1})$ such that $\theta_{\check{v}}(x)$ is a descendant of u (such a vertex exists, because $\theta_{\check{v}}(x) \in A$). Let $\beta_i(u, I^{(i)}) = (u', \psi)$, with ψ of the form $(\ldots \Rightarrow I^{(i)}(s))$, $\theta_u = \mathrm{emb}(s \mapsto u)$. One can show that there exists a subterm s'' of s such that $\theta_{\check{v}}(x) = \theta_u(s'')$. We define $\sigma(x) = s''$. Let $\varphi' = \varphi\sigma$. One can show that $\varphi' \in \hat{\Phi}$. Finally, let $\beta_i(v, I^{(j)}) = (w, \varphi')$.

One can prove that $\mathcal{D}_i$ is an ADAG and $(**)$ holds. Now we will show that $(*)$ holds. Let $u \notin \mathcal{B}(\mathcal{D}_i)$ be a descendant of some $v \in G_j(\mathcal{D}_i)$, for some $j = 1, \ldots, i$. Note that $u \notin \mathcal{B}(\mathcal{D}_i)$ implies $u \notin X$. For $j < i$, if we suppose that $u \in G_{>j}(\mathcal{D}_i)$, then we have $\check{v} \in G_j(\mathcal{D}_{i-1})$ and $\check{u} \in G_{>j}(\mathcal{D}_{i-1})$. We also have that $\check{u}$ is a descendant of $\check{v}$, which contradicts the inductive hypothesis.

Now, assume that $j = i$. Note that, for any $v \in A$, the vertex $\hat{v}$ is not a descendant of any $v' \in G_i(\mathcal{D}_i)$. So, suppose that $u \in A$. In this case the definition of δ_i guarantees that $u \notin G_{>j}(\mathcal{D}_i)$. Second, suppose that $u \notin A$. In this case $u \in S^i_{\mathcal{D}_i}$, and because $u \notin \mathcal{B}(\mathcal{D}_i)$, u is free and u is reachable from $\alpha(t_i)$. It means that there is a path $v_1, \ldots, v_M$ in $\mathcal{D}_i$, such that $v_1 = \alpha(t_i)$, v_M is a leaf, and $u = v_k$, for some k. Because $I^{(i-1)} \in \delta_i(v_1)$, then there exists an index l such that $v_l \in G_{i-1}(\mathcal{D}_i)$ and, for each $l' = 1, \ldots, l$, $I^{(i-1)} \in \delta_i(v_{l'})$. So, if $k \leq l$, then $u \notin G_{>j}(D_i)$ ($\delta_i(v)$ contains $I^{(i-1)}$, so it cannot contain $I^{(j)}$ for any $j \neq i-1$), and if $k > l$, then by inductive hypothesis, we also have $u \notin G_{>j}(D_i)$. It concludes the proof of Lemma 6. □

One can also prove, using very similar argumentation to the one in the last paragraph of the proof above, the following fact.

Lemma 9. *If $\mathcal{D}$ is a simple* ADAG, $u \in S^i_{\mathcal{D}}$ *and* $u \notin \mathcal{B}(\mathcal{D})$*, then* $u \notin G_{\geq i}(\mathcal{D})$.

4.5 Proof of Lemma 7

We will construct a sequence $\mathcal{D}_n, \dots, \mathcal{D}_0$ of ADAGs, starting with $\mathcal{D}_n = \mathcal{D}$. We will show that $G_{\geq i}(\mathcal{D}_i)$ is exponentially bounded, which, for $i = 0$, means that the set of goals of $\mathcal{D}_0$ is exponentially bounded. All the ADAGs of this family share the same α, $\hat{\Phi}$, and the same set of vertices. So, let $\mathcal{D}_{i+1} = \langle \mathcal{D}_{i+1}, \alpha, \beta, \delta_{i+1} \rangle$. We will construct $\mathcal{D}_i = \langle \mathcal{D}_i, \alpha, \beta, \delta_i \rangle$. By induction, we assume that $G_{>i}(\mathcal{D}_{i+1})$ is exponentially bounded.

For $v_1, v_2 \in G_i(\mathcal{D}_{i+1})$, let $v_1 \sim v_2$ iff $\delta_{i+1}(v_1) = \delta_{i+1}(v_2)$. Let h be a function which for the equivalence class $[v]_\sim$ of v, gives some vertex $h([v]_\sim) \in [v]_\sim$ such that no vertex $v' \in [v]_\sim$ is a descendant of $h([v]_\sim)$. Let $H = \{v \in G_i(\mathcal{D}_{i+1}) \mid h([v]_\sim) = v\}$. Let $\mathcal{G}$ be the least subset of $G_i(\mathcal{D}_{i+1})$ such that:
(a) if $u \in G_i(\mathcal{D}_{i+1})$ is an element of $\mathcal{B}(\mathcal{D}_{i+1}) \cup H$, then $u \in \mathcal{G}$,
(b) if $u \in B^v_{\mathcal{D}_{i+1}}$, for some $v \in G_{>i}(\mathcal{D}_{i+1})$, then $u \in \mathcal{G}$,
(c) if $u \overset{\mathcal{D}_{i+1}}{\rightsquigarrow}{}^{*} u'$, for some $u' \in G_{>i}(\mathcal{D}_{i+1})$, then $u \in \mathcal{G}$,
(d) if $u \in G_i(\mathcal{D}_{i+1})$ is a descendant of some $u' \in \mathcal{G}$, then $u \in \mathcal{G}$.
Using Lemma 9 and the fact that, for $u \in G_i(\mathcal{D}_{i+1})$, we have $F^u_{\mathcal{D}_{i+1}} \subseteq S^i_{\mathcal{D}_{i+1}}$, one can show that each $u \in G_i(\mathcal{D}_{i+1})$ can have at most exponentially many descendants in $\mathcal{G}$, and hence, the size of $\mathcal{G}$ is exponentially bounded as well. Let $\bar{\mathcal{G}} = G_i(\mathcal{D}_{i+1}) \setminus \mathcal{G}$.

The construction of $\mathcal{D}_i$. We define $\delta_i(v)$ as follows. Let $\delta_Q(v) = \delta_{i+1}(v) \cap Q$, let $\delta_R(v) = \{r^{(j)} \mid r^{(j)} \in \delta_{i+1}(v),\ \text{and}\ (v, r^{(j)}) \overset{\mathcal{D}_{i+1}}{\rightsquigarrow} u,\ \text{for some}\ u \notin \bar{\mathcal{G}}\ \}$, and let $\delta_I(v) = \{I^{(j)} \mid I^{(j)} \in \delta_{i+1}(v)\}$. If $v \in \bar{\mathcal{G}}$, then let $\delta_i(v) = \delta_Q(v) \cup \delta_R(v)$. Otherwise, let $\delta_i(v) = \delta_Q(v) \cup \delta_R(v) \cup \delta_I(v)$. To define the term-DAG D_i, let $v =_{\mathcal{D}_{i+1}} f(v_1, \dots, v_k)$. For each $i = 1, \dots, k$, we define v'_i: If $I^{(j)} \in \delta(v)$, $v \notin G_j(\mathcal{D}_{i+1})$, and $v_i \in \bar{\mathcal{G}}$, then $v'_i = h([v_i]_\sim)$. Otherwise, $v'_i = v_i$. Note that because $G_i(\mathcal{D}_i) = \mathcal{G}$, the size of $G_i(\mathcal{D}_i)$ is exponentially bounded. Note also that the number of goals from $G_{>i}(\mathcal{D}_i) \cup G_{<i}(\mathcal{D}_i)$ has not been changed.

One can show that $\mathcal{D}_i$ it is an ADAG. The most difficult thing to prove is that the item (iii,d) of Definition 4 holds for each vertex $v \in G_j(\mathcal{D}_i)$ (for some j). So suppose that $v \in G_j(\mathcal{D}_i)$. Clearly, $v \in G_j(\mathcal{D}_{i+1})$. Let $(v', \varphi) = \beta(v, I^{(j)})$ with $\varphi = \big(q_1(t'), \dots, q_l(t'), r(t') \Rightarrow I^{(j)}(t)\big)$. We have (t, t') embeds to (v, v') in $\mathcal{D}_{i+1}$. If $B^v_{\mathcal{D}_{i+1}}$ does not contain any $u \in \bar{\mathcal{G}}$, then $B^v_{\mathcal{D}_{i+1}}$ and $B^v_{\mathcal{D}_i}$ have exactly the same structure and clearly (t, t') embeds to (v, v') in $\mathcal{D}_i$. So suppose that there exists $u \in B^v_{\mathcal{D}_{i+1}}$ such that $u \in \bar{\mathcal{G}}$. Because $u \notin \mathcal{G}$, we have $v \notin G_{>i}(\mathcal{D}_{i+1})$ (see (b) above). We consider two cases. In the both we get a contradiction.

1. *u is a descendant of v.* Then $v \notin G_{<i}(\mathcal{D}_{i+1})$, because $\mathcal{D}_{i+1}$ is simple and, by (a), $u \notin \mathcal{B}(\mathcal{D}_{i+1})$. So, $v \in G_i$. But in this case we cannot have $v \in \mathcal{G}$ (because u would be in $\mathcal{G}$ too; see (d)), and v cannot be in $G_i(\mathcal{D}_i)$.
2. *u is not a descendant of v.* By Lemma 9, either $u \in \mathcal{B}(\mathcal{D}_{i+1})$ and $u \in \mathcal{G}$ (see (a)), or $u \notin S^i_{\mathcal{D}_{i+1}}$ which implies $v' \notin S^i_{\mathcal{D}_{i+1}}$ and $v \notin G_{\leq i}(\mathcal{D}_{i+1})$. □

4.6 Proof of Lemma 8

For an ADAG $\mathcal{D}$ let $U(\mathcal{D})$ be the set of free vertices which are not in $B^{v}_{\mathcal{D}}$, for any goal v of $\mathcal{D}$, and let $\overline{U}(\mathcal{D})$ denote the set of vertices of $\mathcal{D}$ which are not in $U(\mathcal{D})$. One can check that the size of $\overline{U}(\mathcal{D}_0)$ is exponentially bounded.

Now, consider the following procedure. For an input ADAG $\mathcal{D} = \langle D, \alpha, \beta, \Psi, \delta \rangle$ such that some vertex $u \in U(\mathcal{D})$ has more than one parent, we construct an ADAG $\mathcal{D}' = \langle D', \alpha, \beta', \Psi, \delta' \rangle$ in the following way. Let $v_1, \ldots, v_k$ be the parents of u ($k > 1$). We construct the term-DAG D' from D by splitting u into $u_1, \ldots, u_k$ and making v_i the only parent of u_i. If $u' \neq u$, then we put $\delta'(u') = \delta(u)$ and $\beta'(u') = \beta(u')$. We put $\delta'(u_i) = \{p \in \delta(u) \mid \text{either } p \in Q,\ p \text{ is of the form } I^{(j)}, \text{ or } (p', v_i) \overset{\mathcal{D}}{\rightsquigarrow} (p, u),\ \text{for some } p'\}$. For $r \in R \cup \{r_I\}$, $r \in \delta'(u_i)$, we put also $\beta'(u_i, r) = \beta(u, r)$. One can verify, that $\mathcal{D}'$ is in fact an ADAG. Note also that $\overline{U}(\mathcal{D}') = \overline{U}(\mathcal{D})$.

Starting with $\mathcal{D}_0$, we can repeat this procedure until we obtain an ADAG $\mathcal{D}_1 = \langle D_1, \alpha, \beta_1, \hat{\Phi}, \delta_1 \rangle$ for $(\Phi, \pi, \boldsymbol{e})$ such that each $v \in U(\mathcal{D}_1)$ has at most one parent and, because $\overline{U}(\mathcal{D}_0) = \overline{U}(\mathcal{D}_1)$, the number of goals is exponentially bounded. Now, we will minimize the number of vertices in $U(\mathcal{D}_1)$. Let V denotes the set of vertices of $\mathcal{D}_1$, let $U = U(\mathcal{D}_1)$, and $\overline{U} = \overline{U}(\mathcal{D}_1)$. Let $\prec$ be a linear ordering on V compatible with the DAG ordering (i.e. if v is a descendant of v' then $v \prec v'$). Let $v_1 \prec \cdots \prec v_{N-1}$ be all the vertices of $\overline{U}$. For $k = 0, \ldots, N$, we define the k-th segment U_k of $\mathcal{D}$ by the following equations: $U_0 = \{u \in U \mid u \prec v_1\}$, $U_N = \{u \in U \mid v_{N-1} \prec u\}$, and for $k = 1, \ldots, (N-1)$, $U_k = \{u \in U \mid v_{k-1} \prec u \prec v_k\}$. Note that $\bigcup_{k=1}^{N} U_k = U$.

For a vertex v, let $\rho(v) = \{u \mid u \text{ is a goal and } v \overset{\mathcal{D}_1}{\rightsquigarrow}{}^{*} u\}$. Let $v, v' \in U_k$ (for $k = 0, \ldots, N$). Suppose that $\rho(v) = \rho(v')$ and $\delta(v) = \delta(v')$. Then we have either $v < v'$ or $v' < v$. Let us assume that $v < v'$ holds. Let us remove v and replace it by v' (i.e. whenever v was a child of u, we make v' a child of u instead). For each $r \in \delta(v')$, let $\beta(v', r) = \beta(v, r)$. One can prove that what we have obtained is an ADAG. We repeat this procedure until the ADAG has no two distinct vertices $v, v' \in U_k$, for some k, with $\rho(v) = \rho(v')$ and $\delta(v) = \delta(v')$.

Because $|\overline{U}| = |\overline{U}(\mathcal{D}_0)|$ is exponentially bounded and $N = |\overline{U}|$, to complete the proof it is enough to show that each U_k is exponentially bounded. Let M denote the number of goals of the resulting ADAG (which is equal to the number of goals of $\mathcal{D}_0$) and K denote the number of distinct possible values of δ. One can show that each path in U_k is not longer than $M \cdot K$ (since vertices from U_k can have at most one parent, the values of $\rho(u)$ can only decrease along a path). One can also show that, if $v, v' \in U_k$ are not on the same path, then $\rho(v) \cap \rho(v') = \emptyset$, and thus, the number of distinct (maximal) paths in U_k is bounded by M. Hence, the size of U_k is bounded by $M^2 \cdot K$ which is exponential w.r.t. the size of (P, Φ). □

5 Conclusions

We have introduced a new formalism to model recursive cryptographic protocols. In this formalism, one can express protocols such that participants are able to send many messages in one step, to compare, and to store messages. Usefulness of the proposed model is illustrated by an example. We have proven that the insecurity problem of protocols with selecting theories w.r.t. a bounded number of sessions is decidable in NEXPTIME.

The proof technique used in this paper (stage theories, representing attacks by ADAGs) is, in its outline, an adaptation of the method used in [17] to prove NP-completeness of insecurity of (non-recursive) protocols, where the initial knowledge of the intruder is a regular language of terms. In [17], however, the minimization of an ADAG is relatively simple and straightforward, whereas in this paper, it is the main technical difficulty.

Future work. The exact complexity of the problem of deciding protocols with selecting theories is not known. Another open problem is decidability of security of protocols with selecting theories and *with complex keys*.

References

1. Roberto M. Amadio and Witold Charatonik, *On name generation and set-based analysis in the Dolev-Yao model*, CONCUR, Lecture Notes in Computer Science, vol. 2421, Springer, 2002, pp. 499–514.
2. G. Ateniese, M. Steiner, and G. Tsudik, *Authenticated group key agreement and friends*, Proceedings of the 5th ACM Conference on Computer and Communication Serucity (CCS'98), ACM Press, 1998.
3. J. Bryans and S.A. Schneider, *CSP, PVS, and a recursive authentication protocol*, DIMACS Workshop on Formal Verification of Security Protocols, 1997.
4. J.A. Bull and D.J. Otway, *The authentication protocol*, Technical Report DRA/CIS3/PROJ/CORBA/SC/1/CSM/436-04/-03, Defence Research Agency, Malvern, UK, 1997.
5. Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani, *Deciding the security of protocols with Diffie-Hellman exponentiation and products in exponents*, FSTTCS, 2003.
6. ______, *An NP decision procedure for protocol insecurity with XOR*, LICS, 2003.
7. H. Comon and V. Shmatikov, *Is it possible to decide whether a cryptographic protocol is secure or not?*, Journal of Telecommunications and Information Technology, special issue on cryptographic protocol verification **4** (2002), 5–15.
8. H. Comon-Lundh and V. Shmatikov, *Intruder deductions, constraint solving and indecurity decision in presence of exclusive or*, LICS, 2003.
9. D. Dolev and A.C. Yao, *On the security of public-key protocols*, IEEE Transactions on Information Theory **29** (1983), 198–208.
10. N.A. Durgin, P.D. Lincoln, J.C. Mitchell, and A. Scedrov, *Undecidability of bounded security protocols*, Workshop on Formal Methods and Security Protocols (FMSP'99), 1999.
11. S. Even and O. Goldreich, *On the security of multi-party ping-pong protocols*, Technical Report 285, Israel Institute of Technology, 1983.
12. Ralf Küsters and Thomas Wilke, *Automata-based analysis of recursive cryptographic protocols*, Technical Report IFI 0311, CAU Kiel, 2003.
13. ______, *Automata-based analysis of recursive cryptographic protocols*, STACS, Lecture Notes in Computer Science, vol. 2996, Springer, 2004, pp. 382–393.
14. Catherine Meadows, *Formal methods for cryptographic protocol analysis: Emerging issues and trends*, IEEE Journal on Selected Areas in Communication **21** (2003), no. 1, 44–54.
15. L.C. Paulson, *Mechanized proofs for a recursive authentication protocol*, 10th IEE Computer Security Foundations Workshop (CSFW-10), IEEE Press, 1997.
16. Michaël Rusinowitch and Mathieu Turuani, *Protocol insecurity with a finite number of sessions, composed keys is NP-complete*, Theor. Comput. Sci. **1-3** (2003), no. 299, 451–475.
17. Tomasz Truderung, *Regular protocols and attacks with regular knowledge*, Proceedings of CADE 2005, LNCS, Springer, 2005, to appear.