Extending the Dolev-Yao Intruder for Analyzing an Unbounded Number of Sessions*

Yannick Chevalier[1], Ralf Küsters[2], Michaël Rusinowitch[1], Mathieu Turuani[1], Laurent Vigneron[1]

[1] LORIA-INRIA-Universités Henri Poincaré, Nancy 2
54506 Vandoeuvre-les-Nancy cedex, France
email:{chevalie,rusi,turuani,vigneron}@loria.fr

[2] Department of Computer Science
Stanford University, Stanford CA 94305, USA
email: kuesters@theory.stanford.edu

Abstract. We propose a protocol model which integrates two different ways of analyzing cryptographic protocols: i) analysis w.r.t. an unbounded number of sessions and bounded message size, and ii) analysis w.r.t. an a priori bounded number of sessions but with messages of unbounded size. We show that in this model secrecy is DEXPTIME-complete. This result is obtained by extending the Dolev-Yao intruder to simulate unbounded number of sessions.

1 Introduction

Formal analysis has been very successful in finding flaws in published cryptographic protocols [7]. Even fully automatic analysis of such protocols is possible, based on models for which security is decidable or based on approximations (see, e.g., [13, 19, 18, 1, 16], and [17, 9] for an overview of the different approaches, decidability, and complexity theoretic results).

The decidability of security, or more precisely secrecy, of protocols heavily depends on whether in the analysis an unbounded number of sessions of a protocol is taken into account or only an a priori bounded number. In the former case, secrecy is in general undecidable [1, 13, 14], with only a few exceptions [13, 11, 2, 8]. One such exception, which is of particular interest in this paper, is that secrecy is DEXPTIME-complete when the message size is bounded and nonces, i.e., newly generated constants, are disallowed [13]. In what follows, let us call this setting the *bounded message model*. In the latter case, in which the number of sessions is bounded, secrecy is known to be NP-complete [19], even when there is no bound on the size of messages, complex keys are allowed, i.e., keys that may be complex messages, and messages can be paired to form larger messages. We will refer to this setting as the *unbounded message model*.

* This work was partially supoorted by PROCOPE and IST AVISPA. The second author was also supported by the DFG.

In this paper, we integrate the two models — the bounded and the unbounded message model, and thus, integrate two different approaches for protocol analysis: i) analysis w.r.t. an unbounded number of sessions, which has the advantage that the exact sessions to be analyzed do not need to be provided beforehand, but where a bound on the size of messages is put, and ii) analysis which is rather detailed since the size of messages is not bounded, but where only explicitly given sessions are analyzed. More precisely, we consider a protocol model in which there are two kinds of principals, *bounded message* and *unbounded message* principals, or *bounded* and *unbounded* principals for short, which only accept messages of bounded size from the environment or messages of unbounded size, respectively. Conversely, in a protocol run, bounded principals may be involved in an unbounded number of sessions while unbounded principals run in at most one session. The communication between the principals is controlled by the standard Dolev-Yao intruder, in particular, the size of the messages the intruder may produce is not bounded. Just as in the bounded and unbounded message model, the principals and the intruder are not allowed to generate nonces. Our model, in what follows referred to as *integrated model*, comprises both the bounded and the unbounded message model: If in the integrated model the set of bounded principals is empty, then the model coincides with the unbounded message model, and if the set of unbounded principals is empty, then this gives the bounded message model.

The main result shown in this paper is that secrecy in the integrated model is DEXPTIME-complete. The main difficulty is to establish the complexity upper bound. The key idea is as follows: To deal with the bounded principals in the integrated model, and thus, the unbounded number of sessions, the bounded principals are turned into intruder rules, and thus, they extend the ability of the intruder to derive new messages. These intruder rules can be applied by the intruder an arbitrary number of times and in this way simulate the unbounded number of sessions. More precisely, we will extend the standard Dolev-Yao intruder by oracle rules, i.e., intruder rules that satisfy certain properties, and show that insecurity w.r.t. a set of *unbounded* principals and the extended intruder is in NP (given an oracle for applying the oracle rules). — This result is obtained in a similar way as the one in [5], although the kind of oracle rules considered in [5] is quite different from the rules studied here. — We then turn the bounded principals into oracle rules, show that these rules in fact simulate the bounded principals, and prove that the rules can be applied in exponential time. These steps are non-trivial. From this, we conclude the desired complexity upper bound, i.e., obtain a deterministic exponential time algorithm for deciding secrecy in the integrated model.

As we will see in Section 3, the integrated model is not more powerful than the unbounded message model in the sense that from every protocol in the integrated model one can construct a protocol in the unbounded message model such that one protocol preserves secrecy only if the other one does. Moreover, feeding this constructed protocol into an algorithm for analyzing protocols in the unbounded message model, yields an alternative way of deciding secrecy in the integrated model. However, since the number of (unbounded) principals in the constructed protocol grows exponentially, using the NP-completeness result shown in [19], this reduction only provides an NEXPTIME decision algorithm. (Note that, together with the main result of this paper and the result shown in [19], the existence of a polynomial time reduction from the integrated model to the unbounded message model would imply NP=EXPTIME.) More importantly, the constructed protocol is too big for current analysis tools in the unbounded messages model, e.g., [3, 18], since they can only handle a small number of principals. Conversely, in our decision algorithm, we not only reduce secrecy in the integrated model to secrecy in the unbounded message model but in addition extend the Dolev-Yao intruder to simulate the bounded principals. In this way, we avoid creating new (unbounded) principals. In addition to the improved complexity theoretic result, this approach seems to be much better amenable to practical implementations. In fact, in [6] an implementation is presented for an intruder with capabilities similar to those needed here.

Structure of the paper. In the following section, the protocol and intruder model is presented. Then we state the main result (Section 3). In Section 4, the intruder extended by oracle rules is introduced and it is shown that insecurity is in NP given an oracle for applying oracle rules. We then, Section 5, turn bounded principals into oracle rules, and by applying the result from Section 4 establish the complexity upper bound. We conclude in Section 6.

We refer the reader to our technical report [4] for full proofs and a formal description of the Three-Pass Mutual Authentication ISO Protocol in our model.

2 Problem Definition

We now provide a formal definition of our protocol and intruder model. We first define terms and messages, then protocols, and finally the intruder and attacks.

2.1 Terms and Messages

Terms are defined according to the following grammar:

$$term ::= \mathcal{A} \,|\, \mathcal{V} \,|\, \langle term, term \rangle \,|\, \{term\}^s_{term} \,|\, \{term\}^p_{\mathcal{K}}$$

where $\mathcal{A}$ is a finite set of constants (*atomic messages*), containing principal names, nonces, keys, and the atomic messages secret and I (the intruder's name); $\mathcal{K}$ is a subset of $\mathcal{A}$ denoting the set of public and private keys; and $\mathcal{V}$ is a finite set of variables. We assume that there is a bijection $\cdot^{-1}$ on $\mathcal{K}$ which maps every public (private) key k to its corresponding private (public) key k^{-1}. The binary symbol $\langle \cdot, \cdot \rangle$ is called *pairing*, the binary symbol $\{\cdot\}^s_\cdot$ is called *symmetric encryption*, the binary symbol $\{\cdot\}^p_\cdot$ is *public key encryption*. Note that a symmetric key can be any term and that for public key encryption only atomic keys (namely, public and private keys from $\mathcal{K}$) can be used.

Variables are denoted by x, y, terms are denoted by s, t, u, v, and finite sets of terms are written $E, F, \ldots$, and decorations thereof, respectively. We abbreviate $E \cup F$ by E, F, the union $E \cup \{t\}$ by E, t, and $E \setminus \{t\}$ by $E \setminus t$. The cardinality of a set S is denoted by $\mathsf{card}(S)$.

For a term t and a set of terms E, $Var(t)$ and $Var(E)$ denote the set of variables occurring in t and E, respectively.

A *ground term* (also called *message*) is a term without variables. A *(ground) substitution* is a mapping from $\mathcal{V}$ to the set of (ground) terms. The application of a substitution σ to a term t (a set of terms E) is written $t\sigma$ ($E\sigma$), and is defined as usual.

The set of *subterms* of a term t, denoted by $Sub(t)$, is defined as follows:

- If $t \in \mathcal{A} \cup \mathcal{V}$, then $Sub(t) = \{t\}$.
- If $t = \langle u, v \rangle$, $\{u\}_v^s$, or $\{u\}_v^p$, then $Sub(t) = \{t\} \cup Sub(u) \cup Sub(v)$.

Let $Sub(E) = \bigcup_{t \in E} Sub(t)$. We define the size of a term and a set of terms basically as the size of the representation as a dag. That is, the *size* $|t|$ ($|E|$) of a term t (a set of terms E) is $\mathsf{card}(Sub(t))$ ($\mathsf{card}(Sub(E))$).

2.2 Protocols

We now define principals and protocols.

Definition 1. *A* principal *Π is a finite linear ordering of rules of the form $R \to S$ where R and S are terms. We assume that every variable in S occurs in R or on the left-hand side of a rule preceding $R \to S$. The rules are called* principal rules.

A protocol *P is a tuple $(\mathcal{F}_u, \mathcal{F}_b, E_I, \mathcal{D})$ where $\mathcal{F}_u$ and $\mathcal{F}_b$ are finite unions of principals, and thus, partially ordered sets, E_I is a finite set of messages with $I \in E_I$, and $\mathcal{D}$ is some representation of a finite set of messages such that the dag size of messages in the set represented by $\mathcal{D}$ is lineary bounded in the size of the representation of $\mathcal{D}$.*

Given a protocol P, in the following we will assume that $\mathcal{A}$ is the set of constants occurring in P. We define the *size* $|P|$ of P as the number of different subterms in $\mathcal{F}_u$, $\mathcal{F}_b$, and E_I plus the size of the representation of $\mathcal{D}$. For instance, $\mathcal{D}$ may be a non-negative integer n (encoded in unary) representing the set of all messages of dag size $\leq n$. This implies that the dag size of the set of messages represented by $\mathcal{D}$ is exponentially bounded in the size $|P|$ of the protocol. We define $Var(P)$ to be the set of variables occurring in P.

The idea behind the definition of a protocol is as follows. In an attack on P the intruder may use every principal in $\mathcal{F}_u$ at most once but the principals in $\mathcal{F}_b$ maybe used as often as the intruder wishes. In other words, the principals in $\mathcal{F}_b$ may be involved in an unbounded number of sessions in one attack while the principals in $\mathcal{F}_u$ only participate in at most one session. It is well-known that deciding the security of a protocol w.r.t. an intruder who may use an unbounded number of sessions and may produce messages of unbounded size is undecidable

[13, 2]. For this reason, we will restrict the messages that can be substituted for variables of rules in $\mathcal{F}_b$ to belong to the finite domain $\mathcal{D}$. However, we put no restrictions on the variables of rules in $\mathcal{F}_u$, i.e., these variables can be substituted by messages of unbounded size. We therefore refer to principals in $\mathcal{F}_u$ as *unbounded* and to those in $\mathcal{F}_b$ as *bounded.* A rule of an unbounded principal is called *unbounded* and analogously a rule of a bounded principal is *bounded.* In the following section, attacks are defined formally and the relationship to other models is further discussed. As mentioned, our technical report [4] contains a formal description of a protocol in our protocol model.

2.3 The Intruder and Attacks

Our intruder model follows the Dolev-Yao intruder [12]. That is, the intruder has complete control over the network and he can derive new messages from his initial knowledge and the messages received from honest principals during protocol runs. To derive a new message, the intruder can compose and decompose, encrypt and decrypt messages, in case he knows the key. What distinguishes our model from most other models in which security is decidable is that the intruder may use the (bounded) principals as often as he wishes to perform his attack. As mentioned in the introduction, to deal with this, in Section 4 we will extend the intruder by so-called oracle rules.

The intruder derives new messages from a given (finite) set of messages by applying rewrite rules. A *rewrite rule* (or *t-rule*) L is of the form $M \to t$ where M is a finite set of messages and t is a message. Given a finite set E of messages, the rule L *can be applied* to E if $M \subseteq E$. We define the *step relation* $\to_L$ induced by L as a binary relation on finite sets of messages. For every finite set of messages E: $E \to_L E, t$ (recall that E, t stands for $E \cup \{t\}$) if L is a t-rule and L can be applied to E. If $\mathcal{L}$ denotes a (finite or infinite) set of intruder rules, then $\to_{\mathcal{L}}$ denotes the union $\bigcup_{L\in\mathcal{L}} \to_L$ of the step relations $\to_L$ with $L \in \mathcal{L}$. With $\to_{\mathcal{L}}^*$ we denote the reflexive and transitive closure of $\to_{\mathcal{L}}$.

The set of rewrite rules the intruder can use is listed in Table 1. These rules are called *(Dolev-Yao) intruder rules.* In the table, a, b denote (arbitrary) messages, K is an element of $\mathcal{K}$, and E is a finite set of messages.

The intruder rules are denoted as shown in Table 1. We consider $L_{p1}(\langle a, b\rangle)$, $\ldots, L_{sd}(\{a\}^s_b)$ and $L_c(\langle a, b\rangle), \ldots, L_c(\{a\}^s_b)$ as singletons. Note that the number of decomposition and composition rules is always infinite since there are infinitely many messages a, b.

We further group the intruder rules as follows. In the following, t ranges over all messages.

- $L_d(t) := L_{p1}(t) \cup L_{p2}(t) \cup L_{ad}(t) \cup L_{sd}(t)$. In case, for instance, $L_{p1}(t)$ is not defined, i.e., the head symbol of t is not a pair, then $L_{p1}(t) = \emptyset$; analogously for the other rule sets,
- $L_d := \bigcup_t L_d(t)$, $L_c := \bigcup_t L_c(t)$,
- $\mathcal{L}_{DY} := L_d \cup L_c$ (where DY stands for "Dolev and Yao").

	Decomposition rules	Composition rules
Pair	$L_{p1}(\langle a,b\rangle)$: $\langle a,b\rangle \to a$	$L_c(\langle a,b\rangle)$: $a,b \to \langle a,b\rangle$
	$L_{p2}(\langle a,b\rangle)$: $\langle a,b\rangle \to b$	
Asymmetric	$L_{ad}(\{a\}^p_K)$: $\{a\}^p_K, K^{-1} \to a$	$L_c(\{a\}^p_K)$: $a, K \to \{a\}^p_K$
Symmetric	$L_{sd}(\{a\}^s_b)$: $\{a\}^s_b, b \to a$	$L_c(\{a\}^s_b)$: $a, b \to \{a\}^s_b$

Table 1. Intruder Rules

The set of messages the intruder can derive from a (finite) set E of messages is:

$$d_{DY}(E) := \bigcup\{E' \mid E \to^*_{\mathcal{L}_{DY}} E'\}.$$

Before we can define attacks on a protocol $P = (\mathcal{F}_u, \mathcal{F}_b, E_I, \mathcal{D})$, we need some new notions.

Given a partially ordered set $\mathcal{F}$ of principal rules with associated ordering $<$, an *execution ordering* π for $\mathcal{F}$ is a bijective mapping from some subset $\mathcal{F}'$ of $\mathcal{F}$ into $\{1, \ldots, \mathsf{card}(\mathcal{F}')\}$ such that $L < L'$ implies $\pi(L) < \pi(L')$ for every $L, L' \in \mathcal{F}'$. The *size* of π is $\mathsf{card}(\mathcal{F}')$.

The *partially ordered set of instantiations of the bounded principals in* P is $\mathcal{F}_b^{\mathcal{D}} := \{\Pi\sigma' \mid \Pi \in \mathcal{F}_b \text{ and } \sigma' : Var(\Pi) \to \mathcal{D}\}$. The *partially ordered set induced by* P is $\mathcal{F}_P := \mathcal{F}_u \cup \mathcal{F}_b^{\mathcal{D}}$.

We are now prepared to define attacks. In an attack, a principal Π performs his sequence (linear ordering) of principal rules $R_1 \to S_1, \ldots, R_n \to S_n$ one after the other. Note that the different rules may share variables which are subsituted by the same message and in this way model the (unbounded) memory of a principal. When in step i a message m is received, then m is matched against R_i yielding a matcher σ (if any) with $R_i\sigma = m$ and Π returns $S_i\sigma$ as output. Variables in R_i and S_i which occurred in a previous step, and thus, have been assigned a message already, are substituted by this message. As mentioned in Section 2.2, the intruder may use an unbounded principal, i.e., a principal in $\mathcal{F}_u$, at most once, and he may use every bounded principal, i.e., a principal in $\mathcal{F}_b$, as often has he wishes (any time with a possibly different matching). The difference between unbounded and bounded principals is as follows: While an unbounded principal accepts every message as long as it matches the current input pattern R_i, a bounded principal expects that the variables are filled with elements of the domain represented by $\mathcal{D}$. Thus, $\mathcal{F}_b^{\mathcal{D}}$ is the set of instances of bounded principals the intruder may use to perform an attack. Note that subsequent use of an instance after the first time does not yield new knowledge. Therefore, we assume w.l.o.g. that bounded principal instances in $\mathcal{F}_b^{\mathcal{D}}$ are used only once. Note, however, that $\mathcal{F}_b^{\mathcal{D}}$ contains different (an exponential number of) instances of one bounded principal. Altogether, the intruder may use every principal in $\mathcal{F}_P$ once. For a subset of these principals he (nondeterministically) chooses some execution ordering and then tries to produce input messages for the principal rules. These input messages are derived from the intruder's initial knowledge and the output messages produced by executing the principal rules.

The aim of the intruder is to derive the message secret. Formally, attacks are defined as follows.

Definition 2. *Let $P = (\mathcal{F}_u, \mathcal{F}_b, E_I, \mathcal{D})$ be a protocol and let $\mathcal{F}_P$ be the partially ordered set induced by P. An $\mathcal{L}_{DY}$-*attack *(or simply* attack*) on P is a tuple (π, σ) where π is an execution ordering on $\mathcal{F}_P$, of size k, and σ is a ground substitution of the variables occurring in P such that*

$$R_i\sigma \in d_{DY}(S_0, S_1\sigma, ..., S_{i-1}\sigma)$$

for every $i \in \{1, \ldots, k\}$ where $R_i \to S_i = \pi^{-1}(i)$, and

$$\mathsf{secret} \in d_{DY}(S_0, S_1\sigma, ..., S_k\sigma).$$

The decision problem we are interested in is the following set of protocols where we assume the terms occurring in a protocol to be given as dags.

$$\textsc{Insecure} := \{P \mid \text{ there exists an } \mathcal{L}_{DY}\text{-attack on } P\}.$$

If we restrict the set $\mathcal{F}_b$ of bounded principals to be the empty set (and in this case we do not need $\mathcal{D}$), then this is the case of protocol analysis w.r.t. a bounded number of sessions and unbounded message size as considered, for instance, in [15, 19, 18], and called unbounded message model in the introduction. On the other hand, if we restrict $\mathcal{F}_u$ to be an empty set, then this is basically the case of protocol analysis w.r.t. an unbounded number of sessions but with bounded message size as studied in [13], and called bounded message model in the introduction. We note, however, that in contrast to [13], here we allow the intruder to derive messages of arbitrary size, only the size of messages accepted by the (bounded) principals is bounded. Also, we allow complex rather than only atomic keys.

Summing up, with the protocol and the intruder model considered here, we integrate the bounded and the unbounded message models.

3 Main Result

The main result of this paper is:

Theorem 1. *The problem* Insecure *is DEXPTIME-complete.*

In [13], it has been shown that deciding secrecy in the bounded model, i.e., an unbounded number of sessions but bounded messages, is DEXPTIME-complete. Since here we extend this setting, it is not surprising that for Insecure we also obtain DEXPTIME-hardness. In fact, one can use the same reduction, namely a reduction from the recognition problem for Datalog programs [10], as in [13].

In [19], it has been shown that deciding Insecure for protocols P without bounded principals (i.e., $\mathcal{F}_b = \emptyset$) is NP-complete. We can use this result to also obtain an upper bound for Insecure in the general case: Let $P = (\mathcal{F}_u, \mathcal{F}_b, E_I, \mathcal{D})$. Observe that $P \in \textsc{Insecure}$ iff $P' = (\mathcal{F}_u \cup \mathcal{F}_b^{\mathcal{D}}, \emptyset, E_I, \emptyset) \in$

INSECURE. The protocol P' can be handled with the algorithm proposed in [19]. However, since P' may be of size exponential in the size of P this only shows that INSECURE is in NEXPTIME. Thus, the main problem in proving Theorem 1 is to establish the tight upper bound.

The main idea of this proof is as follows: We first extend capabilities of the Dolev-Yao intruder by so-called oracle rules, i.e., intruder rules which satisfy certain conditions. For this extended intruder we show that insecurity for protocols without bounded principals is in NP given an oracle for performing oracle rules (Theorem 2). We then turn the set $\mathcal{F}_b^{\mathcal{D}}$ of instantiated bounded principals into intruder rules and show that these rules are in fact oracle rules. This will yield the claimed exponential time upper bound (Section 5).

In the following section oracle rules are introduced and the NP-decision algorithm is presented.

4 A General Framework

We now extend the Dolev-Yao intruder by oracle rules, which are intruder rules satisfying certain conditions, and show that insecurity in presence of such an extended intruder for protocols without bounded principals is in NP given a procedure for applying oracle rules. We first introduce oracle rules and then present the NP algorithm.

4.1 Extending the Dolev-Yao Intruder by Oracle Rules

In the rest of this paper, let L_o denote a (finite or infinite) set of rewrite rules of the form $M \to t$ where M is a finite set of messages and t is a message. In Definition 4, we will impose restrictions on this set and then call it the set of oracle rules. The subset of L_o consisting of t-rules is denoted by $L_o(t)$. The union of the Dolev-Yao intruder rules and the oracle rules is denoted by $\mathcal{L}_{DY\mathcal{O}} := \mathcal{L}_{DY} \cup L_o$ and called *oracle intruder rules.* Define $\mathcal{L}_c := L_c \cup L_o$ to be the set of composition rules, $\mathcal{L}_c(t) := L_c(t) \cup L_o(t)$, and $\mathcal{L}_d(t)$ to be the set of all decomposition t-rules in Table 1.

The set $d_{DY\mathcal{O}}(E)$ of messages the intruder can derive from E using the rules $\mathcal{L}_{DY\mathcal{O}}$ is defined analogously to $d_{DY}(E)$. Also, $\mathcal{L}_{DY\mathcal{O}}$*-attacks* are defined analogously to $\mathcal{L}_{DY}$-attacks.

Given finite sets of messages E, E', an *($\mathcal{L}_{DY\mathcal{O}}$-)derivation* D of length n, $n \geq 0$ from E to E' is a sequence of steps of the form $E \to_{L_1} E, t_1 \to_{L_2} \cdots \to_{L_n} E, t_1, \ldots, t_n$ with messages $t_1, \ldots, t_n$, $E' = E \cup \{t_1, \ldots, t_n\}$, and $L_i \in \mathcal{L}_{DY\mathcal{O}}$ such that $E, t_1, \ldots, t_{i-1} \to_{L_i} E, t_1, \ldots, t_i$ and $t_i \notin E \cup \{t_1, \ldots, t_{i-1}\}$, for every $i \in \{1, \ldots, n\}$. The rule L_i is called the *ith rule* in D and the step $E, t_1, \ldots, t_{i-1} \to_{L_i} E, t_1, \ldots, t_i$ is called the *ith step* in D. We write $L \in D$ to say that $L \in \{L_1, \ldots, L_n\}$. If $\mathcal{L}'$ is a set of rewrite rules, then we write $\mathcal{L}' \notin D$ to say $\mathcal{L}' \cap \{L_1, \ldots, L_n\} = \emptyset$. The message t_n is called the *goal* of D.

We also need *well formed* derivations which are derivations where every message generated by an oracle intruder rule is a subterm of the goal or a subterm of a term in the initial set of messages.

Definition 3. *Let* $D = E \rightarrow_{L_1} \ldots \rightarrow_{L_n} E'$ *be a derivation with goal* t. *Then,* D *is* well formed *if for every* $L \in D$ *and every* t': $L \in \mathcal{L}_c(t')$ *implies* $t' \in Sub(E, t)$, *and* $L \in \mathcal{L}_d(t')$ *implies* $t' \in Sub(E)$.

We can now define oracle rules. Condition 1. in the following definition requires the existence of well formed derivations. This will allow us to bound the length of derivations and the size of messages needed in derivations. The remaining conditions are later used to bound the size of the substitution σ of an attack.

Definition 4. *Let* L_o *be a (finite or infinite) set of rules and* P *be a protocol. Then,* L_o *is a* set of oracle rules *(w.r.t.* $L_c \cup L_d$ *as defined above) iff there exists a polynomial* $p(\cdot)$ *such that:*

1. *For every message* t, *if* $t \in d_{DY\mathcal{O}}(E)$, *then there exists a well formed derivation from* E *with goal* t.
2. *If* $F \rightarrow_{L_o(t)} F, t$ *and* $F, t \rightarrow_{L_d(t)} F, t, a$, *then there exists a derivation* D *from* F *with goal* a *such that* $L_d(t) \notin D$.
3. *For every rule* $F \rightarrow t \in L_o(t)$ *we have* $|t| \leq p(|P|)$ *and for all* $t' \in F$, $|t'| \leq p(|P|)$.

In what follows, we always assume that L_o is a set of oracle rules. We call a protocol P of the form $(\mathcal{F}_u, \emptyset, E_I, \emptyset)$ *restricted*. We want to decide the insecurity of a restricted protocol w.r.t. an intruder using $\mathcal{L}_{DY\mathcal{O}}$, i.e., the Dolev-Yao intruder rules plus the oracle rules. Formally, the decision problem we are interested in is the following set of *restricted* protocols P:

$\textsc{Insecure}\mathcal{O} := \{P \mid$ there exists an $\mathcal{L}_{DY\mathcal{O}}$-attack on the restricted protocol $P\}$

4.2 An NP Decision Algorithm

The following theorem is used to prove Theorem 1.

Theorem 2. *Let* L_o *be a set of oracle rules. Given a procedure (an oracle) for deciding* $E \rightarrow_{L_o} t$ *for every finite set* E *of messages and message* t *in constant time,* $\textsc{Insecure}\mathcal{O}$ *can be decided by a nondeterministic polynomial time algorithm.*

The NP decision procedure is given in Figure 1. In (1) and (2) of the procedure, an attack (π, σ) is guessed of size polynomially bounded in n. Then, it is checked whether this is in fact an attack.

Obviously, the procedure is sound. As for completeness, one needs to show that it suffices to only consider substitutions bounded as done in the procedure. This is proved in Section 4.3, Theorem 3.

To show that the procedure is in fact an NP procedure given a procedure for deciding $E \rightarrow t \in L_o$, we prove that (3) and (4) can be decided by an NP algorithm. Given that $|R_i\sigma, S_0\sigma, \ldots, S_{i-1}\sigma|$ is polynomially bounded in $|P|$ for

Input: restricted protocol $P = (\mathcal{F}_u, \emptyset, S_0, \emptyset)$ with $n = p(|P|)$, where $p(\cdot)$ is the polynomial associated to the oracle rules, and $V = Var(P)$.

1. Guess an execution ordering π for P. Let k be the size of π. Let $R_i \to S_i = \pi^{-1}(i)$ for $i \in \{1, \ldots, k\}$.
2. Guess a normalized ground substitution σ such that $|\sigma(x)| \leq 3n^2$ for all $x \in V$.
3. Test that $R_i\sigma \in d_{DY\mathcal{O}}(\{S_0\sigma, \ldots, S_{i-1}\sigma\})$ for every $i < k$.
4. Test that $\mathsf{secret} \in d_{DY\mathcal{O}}(\{S_0\sigma, \ldots, S_k\sigma\})$.
5. If each test is successful, then answer "yes", and otherwise, "no".

Fig. 1. NP Decision Procedure for Insecurity

every $i \leq k$ (see Corollary 1), it suffices to show that the following problem belongs to NP (given the decision procedure for $E \to t \in^? L_o$):

$$\text{DERIVE} := \{(E, t) \mid \text{ there exists an } \mathcal{L}_{DY\mathcal{O}}\text{-derivation from } E \text{ with goal } t\ \}.$$

In this problem, E and t are assumed to be represented as dags. The following lemma follows quite easily from the existence of well formed derivations (see [4] for the proof).

Lemma 1. *Given a procedure for deciding $E \to^? t \in L_o$,* DERIVE *can be decided in nondeterministic polynomial time.*

¿From Theorem 3 proved in the following section, completeness of the procedure depicted in Figure 1 follows.

4.3 Polynomial Bounds on Attacks

To show completeness of the NP decision algorithm depicted in Figure 1, we need to prove that it suffices to consider substitutions bounded as in the second step of this algorithm. To this end, we consider an attack of minimal size, a so-called normal attack, and show that the size of this attack can be bounded as stated in the algorithm.

Given an attack (π, σ) on a protocol P define $|\sigma| := \Sigma_{x \in Var(P)} |\sigma(x)|$. We say that the attack (π, σ) is *normal* if $|\sigma|$ is minimal, i.e., for every attack (π', σ'), $|\sigma| \leq |\sigma'|$. Clearly, if there is an attack, there is a normal attack. Note also that a normal attack is not necessarily uniquely determined.

The next lemma says that normal attacks can always be constructed by linking subterms that are initially occurring in the problem specification or by terms bounded by $p(|P|)$. This will allow us to bound the size of attacks as desired (Theorem 3 and Corollary 1). To state the lemma, we need some notation.

Let P, R_i, S_i, (π, σ), V, $p(\cdot)$, and k be defined as in Figure 1. Let $\mathcal{SP} = Sub(\{R_j | j \in \{1, \ldots, k\}\} \cup \{S_j | j \in \{0, \ldots, k\}\})$. We recall that $\mathcal{A} \subseteq \mathcal{SP}$.

Definition 5. *Let t and t' be two terms and θ a ground substitution. Then, t is a θ-*match *of t', denoted $t \sqsubseteq_\theta t'$, if t is not a variable, and $t\theta = t'$.*

In [4], we prove:

Lemma 2. *Given a normal attack* (π, σ)*, for all variables* x*:* $|\sigma(x)| \leq p(|P|)$ *or there exists* $t \in \mathcal{S}P$ *such that* $t \sqsubseteq_\sigma \sigma(x)$*.*

Using this lemma, it is now easy to bound the size of every $\sigma(x)$ (see [4] for the proof):

Theorem 3. *For every protocol* P*, if* (π, σ) *is a normal attack on* P*, then* $|\{\sigma(x) \,|\, x \in Var\}| \leq 3 \cdot p(|P|)^2$*, where* $|P|$ *is the size of* P *as defined in Section 2.2 and* $p(\cdot)$ *is the polynomial associated to the set of oracle rules.*

From this, we easily obtain:

Corollary 1. *For every protocol* P *and normal attack* (π, σ) *on* P*:* $|R_i\sigma, S_0\sigma, \ldots, S_{i-1}\sigma|$ *and* $|Secret, S_0\sigma, \ldots, S_k\sigma|$ *can be bounded by a polynomial in* $|P|$ *for every* $i \in \{1, \ldots, k\}$*.*

5 Proof of the Complexity Upper Bound

We now show the complexity upper bound claimed in Theorem 1. In what follows, let $P = (\mathcal{F}_u, \mathcal{F}_b, E_I, \mathcal{D})$ be a protocol.

The idea of the proof is to turn the partially ordered set $\mathcal{F}_b^{\mathcal{D}}$ of instantiated bounded principals into oracle rules and then use Theorem 2.

The conversion of $\mathcal{F}_b^{\mathcal{D}}$ is carried out in two steps. First, this set is turned into a set of so-called aggregated rules. Then, the rules are turned into oracle rules.

5.1 Aggregated Rules

The set $\mathcal{F}_b^{\mathcal{D}}$ consists of a finite set of (instantiated) principals Π. Assume that the linear ordering associated to Π is $<$, $\Pi = \{R_0 \to S_0, \ldots, R_{n-1} \to S_{n-1}\}$, and $R_i \to S_i < R_j \to S_j$ for every $i < j$.

Now, replace every $R_i \to S_i$ in Π by a rewrite rule $\{R_0, \ldots, R_i\} \to S_i$. We denote the resulting set by Π_{agg} and call this set the *aggregated version of* Π. Let $\mathcal{F}_{agg}$ denote the set obtained from $\mathcal{F}_b^{\mathcal{D}}$ by replacing every principal by its aggregated version. We call this set *the set of aggregated rules induced by* P. Define $\mathcal{L}_{agg} := \mathcal{L}_{DY} \cup \mathcal{F}_{agg}$, the *set of aggregated intruder rules (induced by* P*)*. Note that $\mathcal{L}_{agg}$ depends on P. However, for simplicity, we omit P in the notation of this set.

The set of terms the intruder can derive from E using $\mathcal{L}_{agg}$ is defined as:

$$d_{agg}(E) := \bigcup \{E' \mid E \to^*_{\mathcal{L}_{agg}} E'\}.$$

An $\mathcal{L}_{agg}$*-attack* on P is defined analogously to $\mathcal{L}_{DY}$-attacks.

The following lemma states that there is an $\mathcal{L}_{DY}$-attack on P iff there exists an $\mathcal{L}_{agg}$-attack on P when the bounded principals of P are removed. In other words, the bounded principals are moved to the intruder. The proof of this lemma is straightforward.

Lemma 3. *There exists an $\mathcal{L}_{DY}$-attack on $P = (\mathcal{F}_u, \mathcal{F}_b, E_I, \mathcal{D})$ iff there exists an $\mathcal{L}_{agg}$-attack on $(\mathcal{F}_u, \emptyset, E_I, \emptyset)$.*

¿From this and if $\mathcal{F}_{agg}$ were oracle rules (in the sense of Definition 4), INSECURE $\in$ *DEXPTIME* would immediately follow from Theorem 2. In general the set $\mathcal{F}_{agg}$ does not meet the restrictions on oracle rules. Therefore, we define *principal oracle rules* meeting the restrictions on oracle rules. In what follows, they are formally defined and it is shown that whether $E \to t$ is such a rule can be decided in exponential time. Then, we show that these rules can replace aggregated rules and that they are oracle rules. Together with Theorem 2, this will yield Theorem 1.

5.2 Principal Oracle Rules

Let $\mathsf{Sub}_r(\mathcal{F}_{agg})$ denote the set of subterms occurring on the right hand-side of rewrite rules in $\mathcal{F}_{agg}$.

Definition 6. *A* principal oracle rule *induced by a protocol P is a rewrite rule of the form $E \to t$ where E is some finite set of ground terms with $|u| \leq |P|^2$ for every $u \in E$ and $t \in \mathsf{Sub}_r(\mathcal{F}_{agg})$ such that $t \in d_{agg}(E)$. Let $\mathcal{F}_p$ denote the set of principal oracle rules induced by P.*

Note that in the above definition $|t| \leq |P|^2$ and $\mathcal{F}_{agg} \subseteq \mathcal{F}_p$.

We now show that principal oracle rules can be decided in exponential time.

Proposition 1. *For every E and t, it can be decided in exponential time in the dag size of E and P whether $E \to t \in \mathcal{F}_p$.*

The key to the proof of this proposition is the following lemma, which is proved in [4]. Intuitively, it states that $\mathcal{L}_{agg}$-derivations are well-formed in the sense that the messages produced in each step of the derivation are subterms of a certain set of messages. Note that $\mathcal{L}_{agg}$-derivations are derivations, as defined in Section 4.1, which use only intruder rules from $\mathcal{L}_{agg}$. In what follows, let $\mathcal{H}$ denote the set of subterms of $\mathcal{F}_{agg}$.

Lemma 4. *Assume that $E \to t \in \mathcal{F}_p$. Let D denote a derivation from E with goal t over $\mathcal{L}_{agg}$ of minimal length. Then, $u \in Sub(E, t, \mathcal{H})$ for every message u such that there exists a u-rule in D.*

Now to test whether $E \to t \in \mathcal{F}_p$ one can iteratively apply rules in $\mathcal{L}_{agg}$ to E that create subterms of $E, t, \mathcal{H}$. Let E' be the resulting set of terms. Then, Lemma 4 ensures that $t \in E'$ iff $E \to t \in \mathcal{F}_p$. It is easy to see that E' can be computed in time exponential in the size of P. This completes the proof of Proposition 1.

5.3 Principal Oracle Rules can Replace Aggregated Rules

Let $\mathcal{L}_p := \mathcal{L}_{DY} \cup \mathcal{F}_p$ be the set of *principal intruder rules*. The set of terms the intruder can derive from E using $\mathcal{L}_p$ is defined as $d_p(E) := \bigcup\{E' \mid E \to^*_{\mathcal{L}_p} E'\}$. An $\mathcal{L}_p$*-attack* on P is defined analogously to $\mathcal{L}_{DY}$-attacks.

Obviously, $d_{agg}(E) = d_p(E)$ for every finite set E of messages. As an immediate consequence, we obtain:

Lemma 5. *Let $P = (\mathcal{F}_u, \mathcal{F}_b, E_I, \mathcal{D})$ be a protocol and let $\mathcal{L}_{agg}$ and $\mathcal{L}_p$ be the aggregated and principal intruder rules induced by P. Then, there exists an $\mathcal{L}_{agg}$-attack on $(\mathcal{F}_u, \emptyset, E_I, \emptyset)$ iff there exists an $\mathcal{L}_p$-attack on this protocol.*

Together with Lemma 3 this yields:

Proposition 2. *Let $P = (\mathcal{F}_u, \mathcal{F}_b, E_I, \mathcal{D})$ be a protocol and let $\mathcal{L}_p$ be the principal intruder rules induced by P. Then, there exists an $\mathcal{L}_{DY}$-attack on P iff there exists an $\mathcal{L}_p$-attack on $(\mathcal{F}_u, \emptyset, E_I, \emptyset)$.*

5.4 Principal Oracle Rules are Oracle Rules

In what follows, we identify L_o with $\mathcal{F}_p$, and show that L_o is a set of oracle rules. By definition of $\mathcal{F}_p$, the last condition on oracle rules (Definition 4, 3.) is met with $p(n) = n^2$.

The following lemma shows the second condition in the definition of oracle rules.

Lemma 6. *If $F \rightarrow_{L_o(t)} F, t$ and $F, t \rightarrow_{L_d(t)} F, t, a$, then there exists a derivation D from F with goal a such that $L_d(t) \notin D$.*

Proof. The proof is obvious. It suffices to observe that $a \in d_p(F) \cap \mathsf{Sub}_r(\mathcal{F}_{agg})$, and thus, $F \rightarrow a \in \mathcal{F}_p$. $\square$

The next lemma, shown in [4], states that if a derivation exists, then also a well formed derivation.

Lemma 7. *If $t \in d_p(E)$, then there exists a well formed derivation with goal t.*

The two lemmas imply:

Proposition 3. *The set L_o of principal oracle rules is a set of oracle rules.*

Now, together with Theorem 2 and Proposition 1 this shows the complexity upper bound claimed in Theorem 1.

6 Conclusion

We have proposed a protocol model which integrates what we have called the unbounded and the bounded message models, and we have shown that deciding secrecy in our model is EXPTIME-complete. For this purpose we have extended the Dolev-Yao intruder in a general framework by oracle rules and applied this framework to handle an unbounded number of sessions.

In future work, we will investigate in how far this framework can be applied to yield other interesting extensions of the Dolev-Yao intruder. Another question is whether the oracle rules introduced here can be combined with those considered in [5], with the potential of even more powerful intruders, e.g., those combining unbounded number of sessions with the XOR operator.

References

1. R.M. Amadio and W. Charatonik. On Name Generation and Set-Based Analysis in the Dolev-Yao Model. In *CONCUR 2002*, LNCS 2421, pages 499–514. Springer-Verlag, 2002.
2. R.M. Amadio, D. Lugiez, and V. Vanackère. On the symbolic reduction of processes with cryptographic functions. Technical Report RR-4147, INRIA, 2001.
3. A. Armando, D. Basin, M. Bouallagui, Y. Chevalier, L. Compagna, S. Mödersheim, M. Rusinowitch, M. Turuani, L. Viganò, and L. Vigneron. The AVISS Security Protocol Analysis Tool. In *CAV 2002*, LNCS 2404, pages 349–353. Springer, 2002.
4. Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani, and L. Vigneron. Extending the Dolev-Yao Intruder for Analyzing an Unbounded Number of Sessions. Technical Report available at `http://www.inria.fr/rrrt/liste-2003.html`. To appear.
5. Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with XOR. In *Proceedings of LICS 2003*, 2003. To appear.
6. Y. Chevalier and L. Vigneron. Automated unbounded verification of security protocols. In *CAV 2002*, Springer, 2002.
7. J. Clark and J. Jacob. *A Survey of Authentication Protocol Literature*, 1997. Web Draft Version 1.0 available from http://citeseer.nj.nec.com/.
8. H. Comon-Lundh and V. Cortier. *New decidability results for fragments of first-order logic and application to cryptographic protocols.* In *Proc. 14th Int. Conf. Rewriting Techniques and Applications (RTA'2003)*, Valencia, Spain, June 2003, volume 2706 of LNCS. To appear.
9. 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. To appear.
10. E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. In *CCC'97*, pages 82–101. IEEE Computer Society, 1997.
11. D. Dolev, S. Even, and R.M. Karp. On the Security of Ping-Pong Protocols. *Information and Control*, 55:57–68, 1982.
12. D. Dolev and A.C. Yao. On the Security of Public-Key Protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.
13. N.A. Durgin, P.D. Lincoln, J.C. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Workshop on Formal Methods and Security Protocols (FMSP'99)*, 1999.
14. S. Even and O. Goldreich. On the Security of Multi-Party Ping-Pong Protocols. In *FOCS'83*, pages 34–39, 1983.
15. A. Huima. Efficient infinite-state analysis of security protocols. In *Workshop on Formal Methods and Security Protocols (FMSP'99)*, 1999.
16. R. Küsters. On the decidability of cryptographic protocols with open-ended data structures. In *CONCUR 2002*, LNCS 2421, pages 515–530. Springer-Verlag, 2002.
17. C. Meadows. Open issues in formal methods for cryptographic protocol analysis. In *DISCEX 2000*, pages 237–250. IEEE Computer Society Press, 2000.
18. J. K. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *CCS 2001*, pages 166–175. ACM Press, 2001.
19. M. Rusinowitch and M. Turuani. Protocol Insecurity with Finite Number of Sessions is NP-complete. In *CSFW-14*, pages 174–190, 2001.