Narrowing Based Constraint Solving for the Verification of Security Protocols*

Stéphanie Delaune[1,3] and Florent Jacquemard[2,3]

[1] France Télécom R&D
[2] INRIA, Research Unit Futurs, project SECSI
[3] Laboratoire Spécification & Vérification
ENS de Cachan & CNRS UMR 8643
61 avenue du Président Wilson,
94235 CACHAN Cedex, FRANCE
email: {stephanie.delaune,florent.jacquemard}@lsv.ens-cachan.fr

Abstract. We investigate the resolution of a class of symbolic constraints modulo an equational theory presented by a convergent rewriting system. These constraints are combinations of first-order equations and so-called deduction constraints which can be seen as restricted second-order unification problems. We propose an inference system based on basic narrowing techniques for deciding satisfiability of such constraints, and show its completeness and termination when the rewrite system satisfies some syntactic restrictions.
This result is applied to show NP-completeness of the cryptographic protocols insecurity problem for a bounded number of sessions, when the protocol and intruder semantics are defined by an arbitrary convergent rewrite system in our class. This generalizes former results, as we show that the use of parameterized semantics permits to weaken the security hypotheses for verification, i.e. to address a larger class of attacks.

1 Introduction

Security protocols are paramount in today's secure transactions through public channels. It is therefore essential to obtain through formal proofs as much as possible confidence in their correctness. Many works have been devoted to the use of formal methods in order to automate the proof of existence of logical attacks on such protocols.

This problem is undecidable in general, and the undecidability results from several factors: the ability of agents to generate fresh random data (nonces), the unlimited size of terms, the unboundedness of the number of sessions. Removing the last condition is however sufficient for decidability (while removing the others is not, see [DLM99,CC04,AC02]), and several decision procedures (at least NP-complete) have been proposed (under this condition) for different models of attackers [AL00,MS01,CE02,CLS03,CKR03,DJ04,RT01]. In these approaches, the cryptographic operations like encryption, signature, application of one-way functions *etc* are abstracted into function symbols and the messages are represented by logical terms rather than bit-strings. In this setting, logical attacks can be characterized by sequences of abstract messages exchanged by honest agents executing the protocol and a malicious agent (called the intruder), and searching for such attacks amounts to solving systems of symbolic constraints [AL00,MS01,CLS03]. Most of the former decision procedures are based on a symbolic constraint reduction system (i.e. a set of inference rules) which strongly depends on the capabilities of the intruder to analyze messages, and are therefore restricted to some particular intruder model.

* This work has been partly supported by the RNTL project PROUVÉ 03V360 and the ACI-SI Rossignol.

In this paper, we propose a generic narrowing based inference system for the resolution of constraints modulo a convergent rewriting system which defines the semantics of operators (including in particular cryptographic primitives for encryption and decryption). The constraints are combinations of first-order equations and so-called *deduction constraints* which correspond semantically to a restricted kind of second-order equations[4] $x(t_1, \ldots, t_n) = t$ where $t_1, \ldots, t_n, t$ are first order terms and x is a second order variable which can take its values in contexts made of public operators. We show that our constraint solving procedure is complete and terminating for a certain class of rewriting systems and constraints, and that it can be applied to decide the problem of protocol insecurity for a bounded number of sessions in non-deterministic polynomial time.

The advantages of this approach are twofold. On one hand, we have a generic decision procedure which can be applied to any model which can be axiomatized by rewriting systems in our class. Modeling the properties of cryptographic operators (and hence the capabilities of an intruder to analyze messages) with equational systems was already the approach of [DY83] which is often cited as the pioneer paper in the domain. The class of rewriting systems which are in the scope of our results contains the standard theory of [DY83] and other relevant theories like the theory of involution which is mentioned in [RT01]. Moreover, the usage of our constraint solving procedure is not limited to the verification of cryptographic protocols, though the restrictions were tailored for this application.

On the other hand, our framework permits the specification of protocols in a language which improves most of those used in the approaches cited above, both in readability and expressiveness. First, since we are able to deal with first-order equations, we can add some equations in protocol specifications, like in [CE02], in order to specify explicitly some tests performed by the participants at some stage of the protocol. Second, some *destruction operators* such as decryption or projections can be defined by the rewriting system, and these operators may be used in the protocol specifications, in order to express unambiguously the actions taken by the agents in protocol execution. For instance, if a protocol specifies that an agent A who knows a symmetric key K shall receive a ciphertext $\{N\}_K$ (number N encrypted with K), and answer N, it is often implicitly assumed that A must check whether this message is indeed a ciphertext and that it is really encrypted with K before trying to decipher it and posting the result. From a computational point of view, a decryption procedure satisfying such an assumption needs some kind of integrity checking [Bel96], which is generally not the case of procedures in use. In our settings, we can specify such a protocol in a more general way: A, upon receiving some message X, replies with $d(X, K)$. If X has the form $\{N\}_K$, then A's reply will be indeed simplified to N, thanks to the rewrite rule $d(\{x\}_y, y) \to x$ for the definition of the decryption operator d. This relaxes the above implicit assumptions concerning the verifications of X by A, and hence enables more attacks, as noticed in [Mil03].

After some motivating examples (Section 2) and preliminary definitions of our framework (Section 3), we investigate first in Section 4 the verification of ground constraints with a locality lemma from which it follows that this problem can be decided in polynomial time. Then, we introduce our inference system for constraint solving (Section 5) and prove its correctness, completeness and termination, and show that it provides a non-deterministic polynomial algorithm for the decision of constraint satisfiability. Finally, we show how to apply this procedure to the verification of cryptographic protocols for a bounded number of sessions, by encoding the existence of an attack into the satisfiability of set of constraints.

[4] Our procedure is however not comparable to general second order narrowing procedures such as in [Pre94].

2 Motivations

Consider the following protocol for a symmetric key exchange in an asymmetric cryptosystem. This is a simplification of the Denning-Sacco key distribution protocol [DS81], omitting certificates and timestamps.

$$\begin{array}{l} 0.\ A \to B : A, \{\{K_{ab}\}^a_{pub(A)^{-1}}\}^a_{pub(B)} \\ 1.\ B \to A : \{secret\}^s_{K_{ab}} \end{array}$$

In the first message, the agent A sends to B a freshly chosen symmetric key K_{ab} for further secure communications. This key is encrypted using an asymmetric encryption algorithm (denoted by $\{\ \}^a$) and the secret key of A, $pub(A)^{-1}$. The result of this encryption is later encrypted with B's public key $pub(B)$ so that only B shall be able to learn K_{ab}. Moreover, A appends its name at the beginning of the message so that the receiver B knows which public key to use in order to obtain K_{ab}. Then, B can extract the symmetric key K_{ab} and use it to encrypt (with a symmetric algorithm denoted $\{\ \}^s$) a secret code *secret* he wants to communicate to A (message 1).

It is well-known that the above common syntax used to describe cryptographic protocols is ambiguous. For this reason, in most approaches, protocols are specified as sequence of programs, one for each agent. In our running example, the program of B can be specified as follows:

$$B\text{'s role:}\quad \mathsf{recv}(x_A, \{\{x_{K_{ab}}\}^a_{pub(x_A)^{-1}}\}^a_{pub(x_B)}); \mathsf{send}(\{secret\}^s_{K_{ab}}) \qquad (1)$$

This version of the Denning-Sacco protocol is flawed: there exists an attack involving two sessions of the protocol and an intruder. In the first session, an honest and naive agent a playing A's role initiates voluntarily a communication with the intruder (without knowing he is an intruder). The intruder thus learns $a, \{\{K_{ab}\}^a_{pub(a)^{-1}}\}^a_{pub(I)}$, where $pub(I)$ is the intruder's public key. Hence, the intruder is able to extract the signed key $\{K_{ab}\}^a_{pub(a)^{-1}}$ and the key K_{ab} itself (we assume that he knows the public key of a). Thereafter, the intruder can fool an honest agent b playing B's role (in another session) by sending him $a, \{\{K_{ab}\}^a_{pub(a)^{-1}}\}^a_{pub(b)}$, which makes b believe that he has received a symmetric key K_{ab} from a. The secret in b's answer is thus not secure, because the intruder knows K_{ab}.

As noted in introduction, in the above program (1), we implicitly assume that the agent B checks that the second component of the received message is a ciphertext, with an encryption with the private key of x_A (the first component of the received tuple) and an encryption with his public key (the value of the variable x_B is the name of the agent B in the above program). We may want to specify a more lax agent B which is not able of such a check, and blindly applies the decryption algorithm twice to any received message. Such an agent B can be specified by the following program, which makes use of asymmetric decryption (ad) and left- and right-projection operators (resp. π_1 and π_2):

$$B\text{'s role:}\quad \mathsf{recv}(x); \mathsf{send}(\{secret\}^s_{ad(ad(\pi_2(x), pub(x_B)^{-1}), pub(\pi_1(x)))}) \qquad (2)$$

The answer of B in the above program shall be simplified by rewrite rules defining ad and π_1, π_2 presented later in Section 6.2. There are no ambiguities or implicit checks in program (2) and its verification is performed under security properties which are strictly more general (weaker) than for program (1). Indeed, there exists an attack of program (2) involving only one session, where the intruder does not need to wait for an honest agent to initiate a communication with him (see Section 6 for a complete description of this attack).

Moreover, we can also use equations in programs to express explicitly some checks performed by the agent B. Consider for instance a patched version of the above Denning-Sacco protocol:

$$\begin{array}{l} 0.\ A \to B : A, \{\{A, B, K_{ab}\}^a_{pub(A)^{-1}}\}^a_{pub(B)} \\ 1.\ B \to A : \{secret\}^s_{K_{ab}} \end{array}$$

Some redundancy as been added on purpose in the first message in order to prevent the above first attack. In our setting, the program for B's role can be specified as follows:

$$\begin{array}{l} \mathsf{recv}(x); \\ x_B = \pi_2(\pi_1(ad(ad(\pi_2(x), pub(x_B)^{-1}), pub(\pi_1(x))))); \\ \pi_1(x) = \pi_1(\pi_1(ad(ad(\pi_2(x), pub(x_B)^{-1}), pub(\pi_1(x))))); \\ \mathsf{send}\big(\{secret\}_{ad(ad(\pi_2(x), pub(x_B)^{-1}), pub(\pi_1(x)))}\big) \end{array}$$

With the first equation, B verifies whether he finds his name x_B at the second position of the ciphertext, and with the second equation he checks whether both occurrences of the name of agent A (before and inside the ciphertext) are the same.

The use of explicit destructors and equations allows also to address a broader class of protocols than the ones described in the standard role's model. For instance, the following protocol (see [Tur03]) can not be expressed in the standard role's model.

$$\begin{array}{l} 0.\ A \to B : \{M, B\}^s_K \\ 1.\ B \to A : B \\ 2.\ A \to B : K \\ 3.\ B \to A : M \end{array}$$

The message $\{M, B\}_K$ is seen as a variable x by the agent B who does not know the decryption key K, and one can not express that x must be decomposed after the reception of K in message 2 without the explicit use of a function symbol for symmetric decryption sd. In our approach B's role can be specified as follows:

$$B\text{'s role} \quad \mathsf{recv}(x); \mathsf{send}(x_B); \mathsf{recv}(y); \pi_2(sd(x, y)) = x_B; \mathsf{send}(\pi_1(sd(x, y)))$$

We shall consider in the next sections the problem of constraint solving before returning back to the verification of cryptographic protocols in Section 6.

3 Preliminaries

We now introduce some notations and basic definitions for terms and term rewriting systems (the reader may refer to [DJ90] for a comprehensive survey on term rewriting systems), and then proceed with the definition of the so-called deduction constraints.

3.1 Terms, Substitutions

We assume given a signature $\mathcal{F}$ and an infinite set of variables $\mathcal{X}$. The set $\mathcal{F}$ is partitioned into a subset $\mathcal{PF}$ of *private* functions symbols, and a subset $\mathcal{VF}$ of *visible* or *public* functions symbols. The set of terms built with $\mathcal{F}$ and $\mathcal{X}$ is denoted $\mathcal{T}(\mathcal{F}, \mathcal{X})$ and its subset of ground terms (terms without variables) $\mathcal{T}(\mathcal{F})$. We denote $vars(t)$ the set of variables occurring in a term $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $st(t)$ the set of subterms of t and $sst(t) = st(t) \setminus \{t\}$ the set of strict subterms of t. These notations are extended as expected to sets of terms and term rewriting systems. The positions in a term t are represented as sequence of integers and are denoted by $Pos(t)$. The empty sequence Λ denotes the top-most position. If p is a position of t, then $t|_p$ denotes the subterm of t at position p and $t[s]_p$ denotes the term obtained by replacement of $t|p$ by the term s. We denotes by $head(t)$ the root symbol of t.

A *replacement* is the term morphism extension of a finite mapping $\{s_1 \mapsto t_1, \ldots, s_n \mapsto t_n\}$ where $s_1, \ldots, s_n, t_1, \ldots t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If $t_1, \ldots t_n \in \mathcal{T}(\mathcal{F})$, the replacement is called *ground.* A *substitution* is a replacement which domain is a subset of $\mathcal{X}$. As usual, the application of a replacement σ to a term t and the composition of replacements σ_1 by σ_2 are written in postfix notation, respectively $t\sigma$ and $\sigma_1\sigma_2$. A substitution σ is *grounding for t* if $t\sigma \in \mathcal{T}(\mathcal{F})$.

In the paper, $|S|$ denotes the cardinal of the set S. The *size* $\|t\|$ of a term t is the number of positions in t. This notation is extended as expected to a set of terms ($\|T\|$). The *dag-size* $\|T\|_d$ of a set of terms T is the number of distinct subterms of T (*i.e.* it is the number of nodes in a representation of T as a dag with maximal sharing).

3.2 Term Rewriting Systems

A *term rewriting system* (TRS) is a finite set of *rewrite rules* $l \to r$ where $l \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $r \in \mathcal{T}(\mathcal{F}, vars(l))$. A term $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ rewrites to s by a TRS $\mathcal{R}$, denoted $t \to_{\mathcal{R}} s$ if there is a rewrite rule $l \to r$ in $\mathcal{R}$, a position p of t and a substitution σ such that $t|_p = l\sigma$ and $s = t[r\sigma]_p$. If $p = \Lambda$, we write $t \xrightarrow{\Lambda}_{\mathcal{R}} s$. We write $\xrightarrow{*}_{\mathcal{R}}$ for the reflexive and transitive closure of $\to_{\mathcal{R}}$ and $\xleftrightarrow{*}_{\mathcal{R}}$ for its reflexive, transitive and symmetric closure. A $\mathcal{R}$-*unifier* of two terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ (also called $\mathcal{R}$-*solution* of the *equation* $s = t$) is a substitution σ such that $s\sigma \xleftrightarrow[\mathcal{R}]{*} t\sigma$. If $\mathcal{R} = \emptyset$, we simply call σ an unifier. It is well-known that unifiable terms have a *most general unifier* (*mgu*), *i.e.* a substitution σ such that $\sigma \leq \tau$ (there exists ρ such that $\sigma\rho = \tau$) for every other unifier τ of s and t.

A TRS $\mathcal{R}$ is *terminating* if there are no infinite chains $t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} \ldots$, *confluent* if for all t_0, t_1, t_2 such that $t_1 \xleftarrow[\mathcal{R}]{*} t_0 \xrightarrow[\mathcal{R}]{*} t_2$, there exists t_3 such that $t_1 \xrightarrow[\mathcal{R}]{*} t_3 \xleftarrow[\mathcal{R}]{*} t_2$, and *convergent* if it is both terminating and confluent. A term t is in $\mathcal{R}$-*normal form* if there is no term s with $t \to_{\mathcal{R}} s$ and the set of $\mathcal{R}$-normal forms is denoted $NF_{\mathcal{R}}$. If $t \xrightarrow{*}_{\mathcal{R}} s$ and $s \in NF_{\mathcal{R}}$ then we say that s is a $\mathcal{R}$-normal form of t, and write $s = t\downarrow_{\mathcal{R}}$. A substitution σ is called $\mathcal{R}$-normal if for every variable $x \in dom(\sigma)$, $x\sigma \in NF_{\mathcal{R}}$.

Definition 1. *A TRS $\mathcal{R}$ is called* public-collapsing *if every rule $\ell \to r \in \mathcal{R}$ verifies the two following conditions:*

1. *$r \in vars(\ell)$ or $r \in \mathcal{T}(\mathcal{VF})\downarrow_{\mathcal{R}}$ and $r \neq \ell$,*
2. *if $\ell = f(l_1, \ldots, l_n)$ with $f \in \mathcal{VF}$, then for all $i \leq n$, for all position $p \in Pos(l_i)$ such that $l_i|_p = g(t_1, \ldots, t_m)$ with $g \in \mathcal{VF}$, either $g(t_1, \ldots, t_m) \in \mathcal{T}(\mathcal{VF})\downarrow_{\mathcal{R}}$, or there exists $j \leq m$ such that $t_j = r$.*

The following trivial lemma shall be used later while reasoning on public-collapsing systems.

Lemma 1. *Let $\mathcal{R}$ be a public-collapsing TRS and let $s, s_1, \ldots, s_n \in \mathcal{T}(\mathcal{F})$ be in $\mathcal{R}$-normal form. We have $s = f(s_1, \ldots, s_n)\downarrow_R$ iff $s = f(s_1, \ldots, s_n)$ or $f(s_1, \ldots, s_n) \xrightarrow{\Lambda}_{\mathcal{R}} s$.*

3.3 Intruder Deductions and Constraints

We assume from now on given a convergent public-collapsing TRS $\mathcal{R}$. We assume given a linear well-founded ordering $\prec$ on $\mathcal{T}(\mathcal{F})$ and a special term denoted by 0 such that $0 \in NF_{\mathcal{R}}$

and is minimal w.r.t. $\prec$. We shall use the extension $\ll$ of $\prec$ to multisets of ground terms. We are studying below the saturation of sets of ground terms under the application of visible function symbols of $\mathcal{VF}$ and rewrite rules of $\mathcal{R}$ ($\mathcal{R}$ is supposed to define the semantics of the symbols of $\mathcal{F}$). This aims, in the context of protocol verification (see Section 6.2), at modeling an intruder who is able to deduce messages from the ones collected on the insecure network.

Given a set of terms $T \subseteq \mathcal{T}(\mathcal{F})$, the *intruder set* $\mathcal{I}_{\mathcal{R}}(T)$ is the smallest, w.r.t. inclusion, subset of $\mathcal{T}(\mathcal{F})$ containing T, closed under $\xleftrightarrow{*}_{\mathcal{R}}$, and such that for all $t_1, \ldots, t_n \in \mathcal{I}_{\mathcal{R}}(T)$ and all $f \in \mathcal{VF}$ of arity n, $f(t_1, \ldots, t_n) \in \mathcal{I}_{\mathcal{R}}(T)$.

A *deduction constraint* is a tuple of terms written $t_1, \ldots, t_n \Vdash r$ where $t_1, \ldots, t_n, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. The terms $t_1, \ldots, t_n$ are called the *hypotheses* of the deduction constraint and r is called its *target*. A deduction constraint is said to be *basic* when $r \in \mathcal{X}$. Since the order of the hypotheses does not matter, we shall sometimes write a deduction constraint $T \Vdash r$ where T is the finite set $\{t_1, \ldots, t_n\}$. A *$\mathcal{R}$-solution* of a deduction constraint $T \Vdash r$ is a grounding substitution σ such that $r\sigma \in \mathcal{I}_{\mathcal{R}}(T\sigma)$.

Definition 2. *A finite set of deduction constraints $\mathcal{C}$ is* well-formed *if its elements can be ordered as $T_0 \Vdash r_0, \ldots, T_l \Vdash r_l$ such that the following conditions hold:*

1. $0 \in T_0$ *and* $st(\mathcal{R}) \cap \mathcal{T}(\mathcal{VF}) \downarrow_{\mathcal{R}} \subseteq T_0$,
2. *for all* $i < l$, $T_i \subseteq T_{i+1}$,
3. *for all* $i \leq l$, *for all* $x \in vars(T_i)$, *there exists* $j < i$ *such that* $x \in vars(r_j)$.

The definitions of constraints and solutions and the above restrictions have been validated by the application to the verification of protocols presented in Section 6. Intuitively, $T \Vdash r$ is true if, knowing all the terms in T, an intruder is able to construct r. The condition *1* imposes that some terms are in the hypotheses of all the deduction constraints. However it is not really a restriction since these terms, built with public symbols, can always be constructed by the intruder. Condition *2* captures the fact that the intruder never forgets information (every message read by the intruder is added to its knowledge) and Condition *3* says that every variable of $\mathcal{C}$ appears for the first time on the right side of a constraint. Indeed, in our application in Section 6, every variable of $\mathcal{C}$ corresponds to a message received by an agent following the protocol, and the intruder must be able to send such a message.

These conditions are invariant (under some conditions) by application of a substitution and normalization with $\mathcal{R}$, and this result shall be used later while reasoning on well-formed set of deduction constraints.

Lemma 2. *Given a finite well-formed set of deduction constraints $\mathcal{C} = \{T_0 \Vdash r_0, \ldots, T_l \Vdash r_l\}$ and a substitution σ, $\mathcal{C}\sigma$ is well-formed and if moreover for each $i \leq l$, $r_i\sigma \in NF_{\mathcal{R}}$, then $\mathcal{C}\sigma\downarrow_{\mathcal{R}}$ is well-formed.*

The proof of this lemma can be found in the long version of this paper [DJ04b]. Note that the hypothesis $r_i\sigma \in NF_{\mathcal{R}}$ is crucial. Indeed, let us consider for instance the well-formed $\mathcal{C} = \big\{T \Vdash sd(\{a\}_x^a, y); T, x \Vdash b\big\}$, and the substitution $\sigma = \{x \mapsto y\}$. The system $\mathcal{C}\sigma\downarrow_{\mathcal{R}} = \big\{\big\{T \Vdash a; T, x \Vdash b\big\}$ does not fulfill Condition *3* of Definition 2 above, because $sd(\{a\}_x^a, y)\sigma = sd(\{a\}_y^a, y) \notin NF_{\mathcal{R}}$.

3.4 Proof trees

We find convenient for the proofs of the next sections to represent the intruder deductions leading to a term of $\mathcal{I}_{\mathcal{R}}(T)$ by a proof tree describing the deduction steps.

Definition 3. *Given a finite set $T \subseteq \mathcal{T}(\mathcal{F})$ and $u \in \mathcal{T}(\mathcal{F})$, a* proof *$P$ of $T \vdash_{\mathcal{R}} u$ is a tree labeled by terms of $\mathcal{T}(\mathcal{F})$ such that:*

- *every leaf of P is labeled with $v \downarrow_{\mathcal{R}}$ for some $v \in T$,*
- *every internal node of P with n sons $P_1, \ldots, P_n$ whose roots are respectively labeled with $v_1, \ldots, v_n$ is labeled by $f(v_1, \ldots, v_n) \downarrow_{\mathcal{R}}$ for some $f \in \mathcal{VF}$,*
- *the root of P is labeled with $u \downarrow_{\mathcal{R}}$, this label is denoted $root(P)$.*

The size of a proof P is the number of its nodes.

Note that with this definition, every label of a proof is in $NF_{\mathcal{R}}$. A proof P of $T \vdash_{\mathcal{R}} u$ (not reduced to a leaf) is called a *composition proof* if its direct subtrees $P_1, \ldots, P_n$ are such that $root(P) = f(root(P_1), \ldots, root(P_n))$ for some $f \in \mathcal{VF}$. Otherwise, it is called a *decomposition proof* and, by Lemma 1, it means that there exists $f \in \mathcal{VF}$ such that $f(root(P_1), \ldots, root(P_n)) \xrightarrow{\Lambda}_{\mathcal{R}} root(P)$.

Example 1. Assume that $sd \in \mathcal{VF}$ and $\mathcal{R} = \{sd(\{x\}_y^s, y) \to x\}$, and let $T = \{\{m_1\}_k^s, k, m_2\}$. The proof on the left below is a decomposition proof ($m_1 = sd(\{m_1\}_k^s, k) \downarrow_{\mathcal{R}}$) and the one on the right is a composition proof (because $sd(m_2, k) \in NF_{\mathcal{R}}$).

$$\frac{T \vdash_{\mathcal{R}} \{m_1\}_k^s \quad T \vdash_{\mathcal{R}} k}{T \vdash_{\mathcal{R}} m_1} \qquad\qquad \frac{T \vdash_{\mathcal{R}} m_2 \quad T \vdash_{\mathcal{R}} k}{T \vdash_{\mathcal{R}} sd(m_2, k)}$$

Lemma 3. *Given a finite set $T \subseteq \mathcal{T}(\mathcal{F})$ and $u \in \mathcal{T}(\mathcal{F})$, $u \in \mathcal{I}_{\mathcal{R}}(T)$ iff there exists a proof of $T \vdash_{\mathcal{R}} u$.*

4 Checking Ground Constraints

In this section, we show how to solve deduction constraints without variables, *i.e.* how to decide, given a finite set $T \subseteq \mathcal{T}(\mathcal{F})$ such that $st(\mathcal{R}) \cap \mathcal{T}(\mathcal{VF}) \downarrow_{\mathcal{R}} \subseteq T$, $0 \in T$ and given a term $u \in \mathcal{T}(\mathcal{F})$, whether $u \in \mathcal{I}_{\mathcal{R}}(T)$ holds or not. Following the approach of [CLS03], we show first that $u \in \mathcal{I}_{\mathcal{R}}(T)$ ensures the existence of a *local* proof, *i.e.* a proof which only involves terms in $st(T \downarrow_{\mathcal{R}} \cup \{u \downarrow_{\mathcal{R}}\})$. Then, we show that using this result, we can determine in polynomial time in the size of T and u, whether $u \in \mathcal{I}_{\mathcal{R}}(T)$.

Lemma 4 (locality). *Let T be a finite subset of $\mathcal{T}(\mathcal{F})$ such that $st(\mathcal{R}) \cap \mathcal{T}(\mathcal{VF}) \downarrow_{\mathcal{R}} \subseteq T$ and $0 \in T$, and let a term $u \in \mathcal{T}(\mathcal{F})$. Every minimal size proof P of $T \vdash_{\mathcal{R}} u$ is labeled by terms in $st(T \downarrow_{\mathcal{R}} \cup \{u \downarrow_{\mathcal{R}}\})$ and if moreover P is a decomposition proof then it is labeled by terms in $st(T \downarrow_{\mathcal{R}})$.*

Proof. (sketch, see [DJ04b] for details). We prove the two results simultaneously by induction on the proof P. The only difficult case is when we have to take into account a rewriting step after the application of a visible function symbol, *i.e* when P is a decomposition proof. Clearly, $root(P)$ is a subterm of one of the direct subproof of P, however it remains to show that the root of the direct subproofs of P are labeled with subterms of T. It is treated by case analysis on the condition verified by the rewrite rule involved in the reduction. □

Now, using this locality Lemma 4, we show that we can decide in polynomial time whether $u \in \mathcal{I}_{\mathcal{R}}(T)$.

Proposition 1. *Given a finite set $T \subseteq \mathcal{T}(\mathcal{F})$ such that $st(\mathcal{R}) \cap \mathcal{T}(\mathcal{VF}) \downarrow_{\mathcal{R}} \subseteq T$ and $0 \in T$, and a term $u \in \mathcal{T}(\mathcal{F})$, whether $u \in \mathcal{I}_{\mathcal{R}}(T)$ can be decided in polynomial time in $\|T \cup \{u\}\|_d$.*

Proof. (sketch, see [DJ04b] for details). By Lemmas 3 and 4, if $u \in \mathcal{I}_{\mathcal{R}}(T)$ then there exists a proof P of $T \vdash_{\mathcal{R}} u$ labeled only with terms in $st(T \downarrow_{\mathcal{R}} \cup \{u \downarrow_{\mathcal{R}}\})$. To decide the existence of such a proof tree, we construct (following [McA93]), a set $\mathcal{S}$ of ground Horn clauses of size polynomial in $\|T \cup \{u\}\|_d$ which implements a marking of every ground subterms $t \in st(T \downarrow_{\mathcal{R}} \cup \{u \downarrow_{\mathcal{R}}\})$ such that there exists a proof of $T \vdash_{\mathcal{R}} t$. Therefore, the existence of a proof of $T \vdash_{\mathcal{R}} u$ is equivalent to the HORN-SAT problem for $\mathcal{S}$, and hence this problem can be solved in polynomial time. □

5 Satisfiability of Well-Formed Sets of Deduction Constraints

We shall lift the decision result of Section 4 with a non-deterministic polynomial time procedure to decide the satisfiability w.r.t. $\mathcal{R}$ of well-formed sets of basic deduction constraints (with variables) and equations.

5.1 Constraints Transformation Rules

We present in Figure 1 a set of transformation rules which operate on tuples of the form $(\mathcal{P}, \mathcal{C}, \mathcal{S})$, called *constraints systems* where:

- $\mathcal{P}$ is a set of equations and basic deduction constraints,
- $\mathcal{C}$ is a set of deduction constraints,
- $\mathcal{S}$ is a set of equations in solved form representing bindings in the solution, *i.e.* $\mathcal{S} = \{x_1 = t_1, ..., x_n = t_n\}$ where each $x_i \in \mathcal{X}$ and has only one occurrence in $\mathcal{S}$.

We may associate a substitution $\{x_1 \mapsto t_1, ..., x_n \mapsto t_n\}$ to the third component $\mathcal{S}$ of a system. Below, we shall make no distinction between $\mathcal{S}$ and its associated substitution.

Definition 4. *A $\mathcal{R}$-solution of a system $(\mathcal{P}; \mathcal{C}; \mathcal{S})$ is a grounding substitution σ such that σ is a $\mathcal{R}$-solution of each deduction constraint in $\mathcal{P} \cup \mathcal{C}$, σ is a $\mathcal{R}$-solution of each equation in $\mathcal{P}$, and σ is an unifier of each equation in $\mathcal{S}$.*

Given an initial system of the form $(\mathcal{B} \cup \mathcal{E}, \emptyset, \emptyset)$, where $\mathcal{B}$ is a finite well-formed set of basic deduction constraints and $\mathcal{E}$ is a finite set of equations, the repeated non-deterministic application of the rules of Figure 1 shall terminate (Section 5.2) and produce (in at least one derivation branch) a system in solved form $(\emptyset, \emptyset, \mathcal{S})$ (such systems always have a $\mathcal{R}$-solution) iff $(\mathcal{B} \cup \mathcal{E}, \emptyset, \emptyset)$ has a $\mathcal{R}$-solution (Sections 5.3 and 5.4). This gives a non-deterministic polynomial time procedure for the decision of the satisfiability which is shown NP-hard in Section 7.

From now on, we shall note $\Longrightarrow_{\mathsf{N}}$, $\Longrightarrow_{\mathsf{U}}$,... the binary relation defined by the application respectively of the above rule (N), (U)..., $\Longrightarrow$ denotes the union of all these relations and $\Longrightarrow^+$ and $\Longrightarrow^*$ are the respective transitive and reflexive-transitive closures of $\Longrightarrow$.

$$\frac{\mathcal{P} \cup \{e[u]\};\ \mathcal{C};\ \mathcal{S}}{\mathcal{P} \cup \{e[r]\};\ \mathcal{C}\eta;\ \mathcal{S}\eta \cup \eta}\ (\mathsf{N})$$

Narrowing
e is an equation or a deduction constraint, $u \notin \mathcal{X}$, $l \to r$ is a fresh variant of a rule of $\mathcal{R}$, $\eta = mgu(l\mathcal{S}, u\mathcal{S})$, $root(l) = root(u)$.

$$\frac{\mathcal{P} \cup \{t_1 = t_2\};\ \mathcal{C};\ \mathcal{S}}{\mathcal{P};\ \mathcal{C}\eta;\ \mathcal{S}\eta \cup \eta}\ (\mathsf{U})$$

Syntactic Unification
$\eta = mgu(t_1\mathcal{S}, t_2\mathcal{S})$.

$$\frac{\mathcal{P} \cup \{c\};\ \mathcal{C};\ \mathcal{S}}{\mathcal{P};\ \mathcal{C} \cup \{c\mathcal{S}\};\ \mathcal{S}}\ (\mathsf{B})$$

Blocking
c is a deduction constraint.

$$\frac{\mathcal{P};\ \mathcal{C};\ \mathcal{S}}{\mathcal{P};\ \mathcal{C}\{x \mapsto t\};\ \mathcal{S}\{x \mapsto t\} \cup \{x \mapsto t\}}\ (\mathsf{VE})$$

Variable Elimination
$x \in vars(\mathcal{C}), t \in st(\mathcal{C}) \setminus vars(\mathcal{C})$,
there is no occurrence of x in t.

$$\frac{\mathcal{P};\ \mathcal{C} \cup \{T \Vdash u\};\ \mathcal{S}}{\mathcal{P};\ \mathcal{C};\ \mathcal{S}}\ (\mathsf{G})$$

Ground
if all the terms in T and u are ground
and $u \in \mathcal{I}_{\mathcal{R}}(T)$.

Fig. 1. Constraint transformation rules

5.2 Termination

Proposition 2 (Termination). *The relation $\Longrightarrow$ is strongly terminating. Moreover, given a system $(\mathcal{P}_0; \emptyset; \emptyset)$, for every transformation sequence $(\mathcal{P}_0; \emptyset; \emptyset) \Longrightarrow (\mathcal{P}_1; \mathcal{C}_1; \mathcal{S}_1) \Longrightarrow \ldots \Longrightarrow (\mathcal{P}_n; \mathcal{C}_n; \mathcal{S}_n)$, the length n, the number of successors of every $(\mathcal{P}_i, \mathcal{C}_i, \mathcal{S}_i)$ with $\Longrightarrow$ and the value $\|\mathcal{P}_i\| + \|\mathcal{C}_i \cup \mathcal{S}_i\|_d$. (for every $i \leq n$) are polynomial in $\|\mathcal{P}_0\|$ and $\|\mathcal{R}\|$.*

Proof. (sketch, see [DJ04b] for details). Let the complexity of system $(\mathcal{P}; \mathcal{C}; \mathcal{S})$ be a tuple ordered lexicographically with the following components:

1. $|\mathcal{P}|$, the number of deduction constraints and equations in $\mathcal{P}$,
2. $nb(\mathcal{P})$, the number of terms in $st(\mathcal{P})$ which are unifiable with a left member of a rule of $\mathcal{R}$,
3. $nbv(\mathcal{C})$, the number of distinct variables in $\mathcal{C}$,
4. $|\mathcal{C}|$, the number of deduction constraints in $\mathcal{C}$.

We can show that each rule of Figure 1 reduces the complexity, hence that $\Longrightarrow$ terminates. □

5.3 Correctness

The following proposition shows that the constraint system defined in Figure 1 is correct.

Proposition 3 (Correctness). *For every system $(\mathcal{P}; \emptyset; \emptyset)$, if $(\mathcal{P}; \emptyset; \emptyset) \Longrightarrow^* (\emptyset; \emptyset; \mathcal{S})$ then $(\mathcal{P}; \emptyset; \emptyset)$ has a $\mathcal{R}$-solution.*

Proof. (sketch, see [DJ04b] for details). By induction on the length of the derivation, we show for every rule (R), that if $(\mathcal{P}_1; \mathcal{C}_1; \mathcal{S}_1) \Longrightarrow_R (\mathcal{P}_2; \mathcal{C}_2; \mathcal{S}_2)$ and the second system $(\mathcal{P}_2; \mathcal{C}_2; \mathcal{S}_2)$ has a $\mathcal{R}$-solution σ, then σ is also a $\mathcal{R}$-solution of $(\mathcal{P}_1; \mathcal{C}_1; \mathcal{S}_1)$. □

5.4 Completeness

We show now the completeness of the constraint system defined in Figure 1 (Proposition 4). We shall first give three technical lemmas (their complete proofs can be found in [DJ04b]): Lemma 5 and Lemma 6 (which allows to apply replacements on proof tree labels) are used in the proof of Lemma 7, which is used in the proof of the Proposition 4 to establish the completeness of the rule (VE).

Lemma 5. *Let $T \subseteq NF_{\mathcal{R}}$ be such that $st(\mathcal{R}) \cap \mathcal{T}(\mathcal{VF}) \downarrow_{\mathcal{R}} \subseteq T$, let $u \in NF_{\mathcal{R}}$, $v \in st(u)$ such that $v \notin st(T)$, and let P be a proof of $T \vdash_{\mathcal{R}} u$. There exists a composition proof of $T \vdash_{\mathcal{R}} v$.*

Lemma 6. *Let $v = g(v_1, .., v_k) \in NF_{\mathcal{R}} \setminus (st(\mathcal{R}) \cap \mathcal{T}(\mathcal{VF}) \downarrow_{\mathcal{R}})$, with $g \in \mathcal{VF}$, let δ be the replacement $\delta = \{v \mapsto 0\}$ and let $u_1, \ldots, u_n \in NF_{\mathcal{R}}$ and $u = f(u_1, \ldots, u_n) \downarrow_{\mathcal{R}}$ for some $f \in \mathcal{VF}$. If $u \neq v, v_1, \ldots, v_k$, then $u\delta = f(u_1\delta, \ldots, u_n\delta) \downarrow_{\mathcal{R}}$.*

Given two substitutions σ_1 and σ_2, we write $\sigma_1 \ll \sigma_2$ iff $\{x\sigma_1 | \ x \in dom(\sigma_1)\} \ll \{x\sigma_2 | \ x \in dom(\sigma_2)\}$.

Lemma 7. *Let σ be a minimal (w.r.t. $\ll$) $\mathcal{R}$-solution of a well-formed set $\mathcal{C}$ of deduction constraints such that all the terms in $\mathcal{C}\sigma$ are in $NF_{\mathcal{R}}$. For all $x \in vars(\mathcal{C})$, there exists $t \in st(\mathcal{C}) \setminus vars(\mathcal{C})$ such that $t\sigma = x\sigma$.*

Proof. (sketch). Let $(C_1, \ldots, C_\ell)$ be a sequence of the constraints of $\mathcal{C}$ as in Definition 2, and for each $i \leq \ell$, let S_i and r_i be respectively the set of hypotheses and target of C_i, and $C_i\sigma$ be the (ground) constraint obtained from C_i by instantiating all the terms in its hypotheses and target with σ.

We reason by contradiction. Assume that there exists $x \in vars(\mathcal{C})$ such that for all $t \in st(\mathcal{C}) \setminus vars(\mathcal{C})$, $t\sigma \neq x\sigma$, and let δ be the replacement $\{x\sigma \mapsto 0\}$. We show that $\sigma' := \sigma\delta$ is also a $\mathcal{R}$-*solution* of $\mathcal{C}$, which contradicts the minimality hypothesis. By Definition 2, if $x\sigma \in st(s\sigma)$ for some hypothesis $s \in S_i$, then there exists $j < i$ such that $x\sigma \in st(r_j\sigma)$ and this allows us to define $m = \min\{j \mid x\sigma \in st(r_j\sigma)\}$.

For each $i < m$, $x\sigma \notin st(C_i\sigma)$, hence $C_i\sigma' = C_i\sigma$ and σ' is a $\mathcal{R}$-*solution* of C_i.

Let $i \geq m$ and let P_i be a proof of $S_i\sigma \vdash_{\mathcal{R}} r_i\sigma$. By Lemma 3, there exists a proof P_m of $S_m\sigma \vdash_{\mathcal{R}} r_m\sigma$ on which we can apply Lemma 5 and deduce that a composition proof P_x of $S_m\sigma \vdash_{\mathcal{R}} x\sigma$. After applying some transformations on the proof tree P_i, using subproofs of P_x and Lemma 6, we obtain a proof P_i'' of $S_i\sigma' \vdash_{\mathcal{R}} r_i\sigma'$, showing that σ' is a $\mathcal{R}$-*solution* of C_i. □

Proposition 4 (Completeness). *Let $\mathcal{B}$ be a finite well-formed set of basic deduction constraints, $\mathcal{E}$ a finite set of first-order equations. If $(\mathcal{B} \cup \mathcal{E}; \emptyset; \emptyset)$ has a $\mathcal{R}$-solution, then there exists a sequence of reductions of the form $(\mathcal{B} \cup \mathcal{E}, \emptyset, \emptyset) \Longrightarrow^* (\emptyset, \emptyset, \mathcal{S})$.*

Proof. (sketch, see [DJ04b] for details). We show, by induction on the complexity of systems, the more general result that if there exists a $\mathcal{R}$-normal solution σ of a system $(\mathcal{P}, \mathcal{C}, \mathcal{S}')$ such that the set of deduction constraints in $\mathcal{P}\mathcal{S}' \downarrow_{\mathcal{R}} \cup \mathcal{C}$ is well-formed, and the terms in $\mathcal{C}\sigma$ are in $NF_{\mathcal{R}}$, then there exists a sequence of reductions of the form $(\mathcal{P}, \mathcal{C}, \mathcal{S}') \Longrightarrow^* (\emptyset, \emptyset, \mathcal{S})$.

The base case $(\emptyset, \emptyset, \mathcal{S}')$ is trivial. For the induction step, we assume that σ is a minimal (w.r.t. $\ll$) $\mathcal{R}$-normal solution as above, and we show that for each case of $(\mathcal{P}, \mathcal{C}, \mathcal{S}')$, we can apply one of the constraint transformation rules of Figure 1, and that the system obtained has a $\mathcal{R}$-normal solution σ' of the above form. The difficult case is the application of the rule (VE). It is treated by using Lemma 7. To conclude, we observe that the above result can be applied to $(\mathcal{B} \cup \mathcal{E}; \emptyset; \emptyset)$. $\square$

Using the above Propositions, we deduce a NP decision procedure for the decision of satisfiability. The proof of the following theorem can be found in [DJ04b].

Theorem 1. *Given a convergent public-collapsing TRS $\mathcal{R}$, a finite well-formed set $\mathcal{B}$ of basic deduction constraints and a finite set $\mathcal{E}$ of equations, the existence of $\mathcal{R}$-solution of $\mathcal{B} \cup \mathcal{E}$ is decidable in non-deterministic polynomial time.*

6 Application to the Verification of Security Protocols

We apply the constraint solving procedure of the previous section in order to obtain a non-deterministic polynomial time decision algorithm for the verification of security properties of cryptographic protocols, assuming a bounded number of sessions. Our algorithm works by associating a set of constraints to a protocol and a security property (Section 6.3), and it is parameterized by a convergent public-collapsing TRS $\mathcal{R}$ which defines the semantics of operators. Some examples of appropriate TRS are given in Section 6.2. As motivated in Section 2, our approach permits to verify protocols modeled in a more expressive language than in the standard approach, and under weaker security hypotheses.

6.1 Protocol Semantics

We consider a simple representation of cryptographic protocols and their execution by agents which should fit with most of the formalisms in use.

A *protocol* is a finite set of programs, each program being a finite sequence of *instructions* of the form $\mathsf{recv}(x); \mathcal{E}; \mathsf{send}(s)$ with $x \in \mathcal{X}$, $s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $\mathcal{E}$ is a set (possibly empty) of equations on terms of $\mathcal{T}(\mathcal{F}, \mathcal{X})$.

Example 2. The first variant of the Denning-Sacco protocol described in Section 2 is made of two programs:

$$A\text{'s role} : \mathsf{recv}(x_0^0); x_0^0 = 0; \mathsf{send}\big(\langle x_A^0, \{\{x_{Kab}^0\}^a_{pub(x_A^0)^{-1}}\}^a_{pub(x_B^0)}\rangle\big); \mathsf{recv}\big(x_1^0\big); \mathsf{send}(0)$$
$$B\text{'s role} : \mathsf{recv}\big(x_0^1\big); \mathsf{send}\big(\{x_S^1\}^s_{ad(ad(\pi_2(x_0^1), pub(x_B^1)^{-1}), pub(\pi_1(x_0^1)))}\big)$$

The symbols $x_0^0, x_1^0, x_A^0 \ldots$ are all distinct variables of $\mathcal{X}$. The second instruction of program A implements only the reception of the last message by A.

Given a protocol $\mathcal{P}$, an agent executing a program p of $\mathcal{P}$ is represented by a *process* (p, σ) where σ is a ground substitution whose domain is a subset of *vars*(p). A *configuration* is a pair (S, N) where S is a finite set of processes whose programs have disjoint sets of variables, and N is a finite set of ground terms representing the network controlled by an intruder. We define small step semantics for the execution of processes. Each step changes the running configuration $\big(\{(p, \sigma)\} \cup S, N\big)$ to $\big(\{(p', \sigma')\} \cup S, N'\big)$ if $p = \mathsf{recv}(x); \mathcal{E}; \mathsf{send}(s); p'$ and there exists a $\mathcal{R}$-solution θ of the equations in $\mathcal{E}\sigma$ such that $x\theta \in \mathcal{I}_{\mathcal{R}}(N)$, $\sigma' = \sigma\theta$ (execution of $\mathsf{recv}(x)$ and control of the conditions in $\mathcal{E}$), and $N' = N \cup \{s\sigma'\}$ (execution of $\mathsf{send}(s)$).

We call an initial configuration of (S_0, N_0) of $\mathcal{P}$ *runnable* if $0 \in N_0$ and at each execution step as above, $s\sigma'$ is ground. It means that every agent is able to construct a term to be sent with the substitution in its initial process (its initial knowledge) or with the values received from other agents.

Example 3. The sequence of processes $((p_0, \sigma_0), (p_1, \sigma_1))$, where p_0 and p_1 are respectively the programs A'a role and B's role of Example 2 and σ_0 and σ_1 are described below, is a runnable initial configuration for the protocol of Example 2 (a, b, k, s are constants):

$$\sigma_0 = \{\, x_A^0 \mapsto a, x_B^0 \mapsto b, x_{Kab}^0 \mapsto k \,\} \quad \sigma_1 = \{\, x_B^1 \mapsto b, x_S^1 \mapsto s \,\}$$

6.2 Useful Public-Collapsing Theories

Let $\mathcal{PF} = \{\_^{-1}\}$ and $\mathcal{VF} = \{\{\_\}_\_^s, sd(\_,\_), \{\_\}_\_^a, ad(\_,\_), \langle\_,\_\rangle, \pi_1(\_), \pi_2(\_), pub(\_)\}$. We give in this section some examples of relevant theories for the application we are considering which fall in the class of convergent public-collapsing TRS built on this signature.

Dolev-Yao theory. The following TRS corresponds to the theory of [DY83] for public key encryption (with an additional rule for symmetric keys decryption). This theory has been studied in many works but, as noted in Section 2, the use of explicit that decryption and projections symbols and equations in protocol specifications permits to generalize other approaches.

$$sd(\{x\}_y^s, y) \to x,\ ad(\{x\}_y^a, y^{-1}) \to x,\ ad(\{x\}_{y^{-1}}^a, y) \to x,\ {x^{-1}}^{-1} \to x,\ \pi_i(\langle x_1, x_2\rangle) \to x_i\ (i = 1, 2)$$

Inverse-key theory. The two following rules extend the Dolev-Yao theory: $\{sd(x,y)\}_y^a \to x$, $\{ad(x,y)\}_{y^{-1}}^a \to x$. They are useful when we assume that decryption is just an encryption with the inverse key like for the cryptosystem RSA.

Theory of involution. It is mentioned in [RT01] and can also be encoded by a convergent public collapsing TRS by adding the following rule to the standard theory: $\{\{x\}_y^a\}_{y^{-1}}^a \to x$. This approach improves the model presented in [RT01] since we consider cases where the rules are applied everywhere in terms and not only at the top of messages.

Probabilistic encryption. We can consider rules such as: $dec(enc(m,k,r),k) \to m$, where enc represents an encryption algorithm which takes a message m, a key k and a random input r.

6.3 Unsecurity Verification via Constraint Solving

We are interested here in verifying whether, given a protocol $\mathcal{P}$, a runnable initial configuration (S_0, N_0) of $\mathcal{P}$, some given ground term s remains secret (i.e. $s \notin \mathcal{I}_\mathcal{R}(N)$) in every configuration (S, N) reachable from (S_0, N_0). Otherwise, we say that there is a $\mathcal{R}$-*attack* on $\mathcal{P}$ for s and (S_0, N_0). Following the constraint solving approach of [AL00,MS01,CLS03], this problem can be addressed as the guess of an interleaving I of executions steps by the processes of S_0 (the reachability sequence) and the satisfiability of a set $\mathcal{C}$ of basic deduction constraints and equations which express the feasibility of this interleaving I. Due to lack of space, we shall not give the formal definition of an interleaving I and the general construction of $\mathcal{C}$ from I. The reader is refereed to [DJ04b] for a more complete presentation. We shall instead describe here the construction of $\mathcal{C}$ on our running example.

Example 4. As announced in Section 2, there is an attack on the protocol of Example 2, starting with the initial configuration (S_0, N_0) with S_0 given in Example 3, and $N_0 = \{0, a, b, pub(a), pub(b)\}$, when $\mathcal{R}$ is the standard Dolev-Yao theory of Section 6.2. In this attack, an intruder, claiming to be a (process p_0) sends to b (process p_1) the "message" $\langle a, \{0\}^a_{pub(b)}\rangle$. The answer of b is then $\{s\}_{ad(ad(\pi_2(\langle a,\{0\}^a_{pub(b)}\rangle),pub(b)^{-1}),pub(\pi_1(\langle a,\{0\}^a_{pub(b)}\rangle)))} \xrightarrow[\mathcal{R}]{*} \{s\}^s_{ad(0,pub(a))}$ and s is revealed since the encryption key $ad(0, pub(a))$ belongs to $\mathcal{I}_{\mathcal{R}}(N_0)$. The interleaving describing the trace of the attack is the sequence of length one $((1, 0))$, (it consists in a single instruction 0 of process p_1), and the set of basic deduction constraints and equations $\mathcal{C}$ associated to this interleaving is:

$$\{N_0 \Vdash x^1_0; N_0, \{s\}_{ad(ad(\pi_2(x^1_0),pub(b)^{-1}),pub(\pi_1(x^1_0)))} \Vdash x; x = s\}$$

Note that the subset of deduction constraints of $\mathcal{C}$ is well formed. The first deduction constraint expresses that the process p_0 is able to receive the expected message x^1_0, *i.e.* that the intruder can deduce it from its initial knowledge N_0 ($x^1_0 \in \mathcal{I}_{\mathcal{R}}(N_0)$). The second deduction constraint expresses that from p_0's answer and N_0, the intruder is able to deduce x. Finally, the last equation expresses that x is the secret. Hence, the $\mathcal{R}$-solvability of $\mathcal{C}$ implies the disclosure of s, starting with state (S_0, N_0). We can check that $\sigma = \sigma_1 \cup \{x^1_0 \mapsto \langle a, \{0\}^a_{pub(b)}\rangle, x \mapsto s\}$ is a solution.

With the construction of $\mathcal{C}$ from a chosen interleaving I (its size is polynomial in the sizes of $\mathcal{P}$, S_0, N_0 and s) and the Theorem 1, we obtain the following theorem.

Theorem 2. *The existence of a $\mathcal{R}$-attack on a given protocol $\mathcal{P}$ for a given secret $s \in \mathcal{T}(\mathcal{F})$ and a given runnable initial configuration (S_0, N_0) is decidable in non-deterministic polynomial time.*

6.4 Related Works

Modeling the behavior of a cryptosystem in terms of rewrite rules is more expressive than the standard approach which consist in modeling cryptosystems in terms of free algebras. Some recent works [Mil03,LM04] compare both approaches, for the case of decryption operators, and give conditions under which security for the free algebra implies security for the rewrite rule model. Hence, under these conditions, explicit decryption operator is unnecessary because it does not enable any new attacks, and formal cryptographic protocol analysis can be made in the free algebra model. We show in this paper that the verification of protocol insecurity in models with rewrite rules for explicit destructors has the same theoretical complexity as in free algebras models.

In [CLT03], the authors prove the decidability of the deducibility by intruder for a class of equational theories. However this class is incomparable with ours. Indeed, for example they allow the homomorphism property but not the idempotence property. In [AC04], it is shown that the problems of deducibility and indistinguishability (static equivalence) are both decidable in PTIME in a model with explicit destructors and equational theories slightly more general than those considered here. Note that these two works are limited to a passive attacker (who can only listen to messages) whereas we treat both cases of passive (PTIME decision procedure) and active attackers (NP decision procedure).

7 NP-hardness

We show now that the existence of a $\mathcal{R}$-attack is a NP-hard problem for polynomial time reductions by reduction of 3-SAT. The proof is inspired from the one given by [RT01]. However, the protocol built from the given instance of 3-SAT is reduced to a minimum thanks to the flexibility of our formalism concerning the choice of a rewriting system.

Let $X_1, ..., X_n$ be propositional variables and let us consider the following instance of 3-SAT: $\bigwedge_{i=1}^{m}(X_{\alpha_{i,1}}^{\epsilon_{i,1}} \vee X_{\alpha_{i,2}}^{\epsilon_{i,2}} \vee X_{\alpha_{i,3}}^{\epsilon_{i,3}})$ where $\alpha_{i,j} \in 1..n$ and $\epsilon_{i,j} \in \{0,1\}$ and X^1 means X, X^0 means $\neg X$. We use a signature made of $\mathcal{VF} = \{0, 1, \pi_1(\_), \ldots, \pi_n(\_), \langle\_, \ldots, \_\rangle\}$ and $\mathcal{PF} = \{\_ \wedge \_, \_ \vee \_, \neg\_, \mathit{secret}\}$, where $\langle\ \rangle$ has arity n, $\pi_1, \ldots, \pi_n$ are unary and *secret* is a constant. Let $\mathcal{R}$ be a convergent public collapsing TRS which defines the truth tables of $\wedge$, $\vee$, $\neg$ with $0 \wedge 0 \to 0 \ldots$ (0 is false and 1 is true) and the projections with $\pi_i(\langle x_1, \ldots, x_n\rangle) \to x_i$ for $i = 1, \ldots, n$. We construct a protocol $\mathcal{P}$ with only one program made up of one instruction:

$$\mathsf{recv}(x); f_1(x) \wedge \ldots \wedge f_m(x) = 1; \mathsf{send}(\mathit{secret})$$

where, for all $i \leq n$, $f_i(x) = \pi_{\alpha_{i,1}}(x)^{\epsilon_{i,1}} \vee \pi_{\alpha_{i,2}}(x)^{\epsilon_{i,2}} \vee \pi_{\alpha_{i,3}}(x)^{\epsilon_{i,3}}$ (we omit the parenthesis in the expressions with $\wedge$ and $\vee$) . Finally, let S_0 contains one process (p_0, σ_0) with $\sigma_0 = \emptyset$ and $N_0 = \{0, 1\}$. We can show that there exists a $\mathcal{R}$-attack on $\mathcal{P}$ for *secret* and (S_0, N_0) iff the instance of 3-SAT has a solution represented by $x = \langle X_1, \ldots, X_n\rangle$ (each X_i is 0 or 1) and this term x is in $\mathcal{I}_{\mathcal{R}}(N_0)$. Hence, the existence of $\mathcal{R}$-attack (Theorem 2) is a NP-complete problem, and, with the construction of Section 6.3, it implies that the problem of solvability of well-formed sets of basic intruder constraints and equations (Theorem 1) is also NP-complete.

8 Conclusion

We have defined a complete inference system for solving equations and deduction constraints modulo convergent and public-collapsing TRS, and we have shown how it provides a generic non-deterministic polynomial time procedure for the verification of security of cryptographic protocols in presence of a finite number of sessions, and with the addition of operators whose semantics are defined by a convergent public-collapsing TRS.

A natural extension to this work is the search of public collapsing theories other than those described in Section 6.2, for the weakening of security hypotheses. For instance, one may want to consider dictionary attacks [DJ04]. An exclusive or operator $+$ can be axiomatized by the rewrite rules $x + x \to 0$, $x + 0 \to x$, $x + x + y \to y$ and associativity and commutativity (AC) of $+$. The three first rules fulfill our public-collapsing condition. Hence, we should consider to extend our solving procedure to a procedure modulo AC in order to deal with xor, like [CKR03,CLS03]. We could also study the generalization of the class of convergent TRS handled. An application could be for instance to model honest protocol transitions by rewrite rules, making the guess of interleaving in the procedure of Theorem 2 unnecessary.

At last, and this is a more difficult task, we could try to extend our result to the decision of static equivalence (following [AC04]). A solution could be to extend the class of constraints under consideration. As noted in introduction, deduction constraints correspond to second order equations (modulo a convergent TRS) of the form $x(t_1, \ldots, t_n) = t$. Being able to deal with equations of the form $x(t_1, \ldots, t_n) = x(s_1, \ldots, s_n)$ could permit us to study properties related to observation equivalence, hence to consider some properties more general than the weak secrecy.

References

[AC02] R.M. Amadio and W. Charatonik. On name generation and set-based analysis in the Dolev-Yao model. In *Proc. of the 13th International Conference on Concurrency Theory (CONCUR)*, 2002.

[AC04] M. Abadi and C. Cortier. Deciding knowledge in security protocols under equational theories. In *Proc. of the 31st International Colloquium on Automata, Languages and Programming (ICALP)*, 2004.

[AL00] R.M. Amadio and D. Lugiez. On the reachability problem in cryptographic protocols. In *Proc. of the 12th International Conference on Concurrency Theory (CONCUR)*, 2000.

[Bel96] S. Bellovin. Problem Areas for the IP Security Protocols," . In *Proc. of the 6th Usenix Unix Security Symposium*, 1996.

[CC04] H. Comon and V. Cortier. Tree Automata with One Memory, Set Constraints and Cryptographic Protocols. *Theoretical Computer Science*, 2004. To appear.

[CE02] R. Corin and S. Etalle. An Improved Constraint-Based System for the Verification of Security Protocols. In *Static Analysis, 9th International Symposium, SAS*, 2002.

[CKR03] Y. Chevalier, R. Kuester, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with Xor. In *Proc. of the 18th International Annual IEEE Symposium on Logic in Computer Science (LICS)*, 2003.

[CLS03] H. Comon-Lundh and V. Shmatikov. Intruder Deductions, Constraint Solving and Insecurity Decision in Presence of Exclusive or. In *Proc. of the 18th International Annual IEEE Symposium on Logic in Computer Science (LICS)*, 2003.

[CLT03] H. Comon-Lundh and R. Treinen, Easy Intruder Deduction. Technical Report LSV-03-8, Lab. Specification and Verification, ENS de Cachan, 2003.

[DJ90] N. Dershowitz and J.-P. Jouannaud. *Rewrite Systems.* Elsevier and MIT Press, 1990.

[DJ04] S. Delaune and F. Jacquemard. A Theory of Dictionary Attacks and its Complexity. In *Proc. of the 17th IEEE Computer Security Foundations Workshop (CSFW)*, 2004.

[DJ04b] S. Delaune and F. Jacquemard. Narrowing Based Constraint Solving for the Verification of Security Protocols. Research Report LSV-04-8, Lab. Specification and Verification, 2004.

[DLM99] N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Proc. of Workshop on Formal Methods and Security Protocols (FMSP)*, 1999.

[DS81] D.E. Denning and G.M. Sacco. Timestamps in Key Distribution Protocols. In *Communications of the ACM*, 1981.

[DY83] D. Dolev and A. Yao. On the Security of Public Key Protocols. In *Proc. of IEEE Transactions on Information Theory*, 1983.

[LM04] C. Lynch and C. Meadows. On the Relative Soundness of the Free Algebra Model for Public Key Encryption. In *Proc. of the 4th Workshop on Issues in the Theory of Security (WITS)*, 2004.

[McA93] D. McAllester. Automatic recognition of tractability in inference relations. In *Journal of the ACM*, 1993.

[Mil03] M. Millen. On the Freedom of Decryption. In *Information Processing Letters*, 2003.

[MS01] J. Millen and V. Shmatikov. Constraint Solving for Bounded-Process Cryptographic Protocol Analysis. In *Proc. of the 8th Conference on Computer and Communications Security (CCS)*, 2001.

[Pre94] C. Prehofer. Higher-Order Narrowing. In *Proc. of the 9th International Annual IEEE Symposium on Logic in Computer Science (LICS), 1994.*

[RT01] M. Rusinowitch and M. Turuani. Protocol Insecurity with Finite Number of Sessions is NP-complete. In *Proc. of the 14th IEEE Computer Security Foundations Workshop (CSFW)*, 2001.

[Tur03] M. Turuani. *Sécurité des Protocoles Cryptographiques: Décidabilité et Complexité.* PhD thesis, Université Henri Poincaré - Nancy 1, 2003.