Protocol Verification Via Rigid/Flexible Resolution

Stépphanie Delaune[2], Hai Lin[1], and Christopher Lynch[1]

[1] Clarkson University, Potsdam, NY 13699-5815, USA
[2] LORIA, CNRS & INRIA project Cassis, Nancy, France

Abstract. We propose a decision procedure, *i.e.* an inference system for clauses containing rigid and flexible variables. Rigid variables are only allowed to have one instantiation, whereas flexible variables are allowed as many instantiations as desired. We assume a set of clauses containing only rigid variables together with a set of clauses containing only flexible variables. When the flexible clauses fall into a particular class, we propose an inference system based on ordered resolution that is sound and complete and for which the inference procedure will halt.

An interest in this form of problem is for cryptographic protocol verification for a bounded number of protocol instances. Our class allows us to obtain a generic decidability result for a large class of cryptographic protocols that may use for instance CBC (Cipher Block Chaining) encryption and blind signature.

1 Introduction

In refutational theorem proving, we must determine if a set of clauses is unsatisfiable. Clauses are implicitly universally quantified, so we allow an unbounded number of renamed copies of each clause. Each copy represents a different instance of the original clause. In proving *rigid unsatisfiability* [1], we ask whether a set of clauses is unsatisfiable, allowing only one instance of each clause. For example, the set of clauses

$$\{I(a),\ I(b),\ \neg I(x) \vee I(f(x)),\ \neg I(f(a)) \vee \neg I(f(b))\}$$

is unsatisfiable but is not rigidly unsatisfiable, because proving unsatisfiability requires two instances of the third clause.

Rigid theorem proving has been used in tableau style theorem provers, but not often in saturation and resolution-based theorem proving. It is generally used to prove unsatisfiability of a set of clauses, by repeatedly solving the rigid satisfiability problem, and continually adding new renamed copies of each clause, because a set of formulas is unsatisfiable if and only if there is a finite number of copies of each clause which make it rigidly unsatisfiable.

Here we give a new use for rigid theorem proving. For example, in cryptographic analysis, we get a reasonable confidence in the protocol security if we

N. Dershowitz and A. Voronkov (Eds.): LPAR 2007, LNAI 4790, pp. 242–256, 2007.

show that, say after any 2 or 3 sessions of the protocols there is no attack. Since the problem is well-known to be undecidable for an unbounded number of protocol instances, many papers study the security problem under this assumption [4,5,15].

However, the problem is more complicated than this. We need rigid clauses to restrict the number of times an action is performed. But simultaneously, we may also want to model certain properties. For example, in cryptographic protocol analysis, the intruder has the ability to construct and deconstruct messages, to encrypt messages and to decrypt messages if the key is known. It is not realistic to bound the number of times that these properties are applied. Therefore, when describing processes, it makes sense to use rigid variables to model actions which are performed a bounded number of times, and flexible variables to model properties which it makes no sense to bound. Therefore, in this paper we introduce the idea of rigid theorem proving modulo a flexible theory. We will use unary predicates to model the state of the world. Actions can be relatively complicated, whereas properties should be simple, according to the theory we are working in.

Unsatisfiability is an undecidable property for first order logic, but rigid unsatisfiability is Σ_2^p-complete, while for Horn clauses, rigid unsatisfiability is *NP*-complete [12]. In this paper, we consider Horn clauses, since that is a natural way to model actions. Our intended application is cryptographic protocol analysis, and Horn clauses are sufficient to model many interesting properties.

We show that rigid/flexible Horn clauses unsatisfiability is decidable for a large class of rigid theories (including those that model cryptographic protocols) and a simple class of flexible theories. The flexible theories we consider model the standard Dolev-Yao model [10] for cryptographic protocol analysis, but also more complicated theories, such as the prefix theory and blind signature theory. The prefix theory related to Cipher Block Chaining (CBC) is important since it is a common encryption mode. It allows an attacker to get from an encrypted message the encryption of any of its prefixes. The blind signature scheme is employed in the design of several E-voting protocols (*e.g.* [11]). It allows an agent (*e.g.* a voter) to have a message (*e.g.* his vote) signed blindly by an authority. This scheme offers an intruder new attacks opportunities.

Ordered Resolution is a powerful inference system for first order logic. So, in this paper, we define a modification of Ordered Resolution, called *Protocol Resolution*. The main result of this paper is the definition of the rigid/flexible unsatisfiability problem, the decidability of a resolution inference system for a useful class of problems, and its application to cryptographic protocol analysis. After the preliminaries (see Section 2), we define how we model protocols in Section 3. Next we define the security problem we want to solve. We give our inference system (Section 5), followed by a termination argument (Section 6) and a completeness argument (Section 7). Then we discuss related and future work. Missing proofs and lemmas can be found at http://people.clarkson.edu/~linh/SHC07.pdf

2 Preliminaries

2.1 Term Algebra

Let $\mathcal{F}$ be a finite set of function symbols with arity and $\mathcal{X}$ be an infinite set of variables. The set of terms on $\mathcal{F}$ and $\mathcal{X}$ is denoted $\mathcal{T}(\mathcal{F},\mathcal{X})$ and $\mathcal{T}(\mathcal{F})$ for ground terms. We note $vars(t)$ the set of variables occurring in $t \in \mathcal{T}(\mathcal{F},\mathcal{X})$. We distinguish two kinds of variables, the *rigid* ones denoted by $X, Y, \ldots$, and the *non-rigid* ones also called *flexible* that we denote by $x, y, \ldots$. Rigid variables are only allowed to have one instantiation, whereas non-rigid variables are allowed as many instantiations as desired. A *substitution* σ is a mapping from a finite subset of $\mathcal{X}$ called its domain and written $dom(\sigma)$ to $\mathcal{T}(\mathcal{F},\mathcal{X})$. Substitutions are extended to endomorphisms of $\mathcal{T}(\mathcal{F},\mathcal{X})$ as usual. If t is a term then $t\sigma$ obtained by applying σ to t is defined as usual. A substitution σ is *grounding* for t if $t\sigma$ is ground. If M and N are terms then a unifier of M and N is a substitution σ such that $M\sigma = N\sigma$. It is well-known that unifiable terms have a 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.

2.2 Clauses

Let $\mathcal{P}$ be a finite set of unary predicate symbols. We assume given a well-founded and total precedence ordering on $\mathcal{P}$, denoted by $>$. *Atoms* A are of the form $I(t)$ where $I \in \mathcal{P}$ and t is a term. *Literals* L are either positive literals $+A$ (or simply A) or negative literals $-A$ where A is an atom. A *clause* is a finite set of literals. We denote by $At(C)$ the set of atoms that appear in a clause C. If C_1 and C_2 are clauses, $C_1 \vee C_2$ denotes $C_1 \cup C_2$ and we denote by $\Box$ the empty clause. A *Horn clause* is a clause that contains at most one positive literal that is called the *head*. For Horn clauses we use the alternative notation $A_1, \ldots, A_n \Rightarrow A_0$ to denote $-A_1 \vee \ldots \vee -A_n \vee A_0$. A clause with no head is called a *goal*. A clause is a *non-rigid* or *flexible clause* (resp. rigid) if it only involves flexible (resp. rigid) variables. A *constrained Horn clause*, which is of the form $\Gamma \Rightarrow A_0 \ [\![C]\!]$, is a Horn clause together with a constraint. The constraint C is a set of equations between terms in $\mathcal{T}(\mathcal{F},\mathcal{X})$. The constraint is omitted when C is empty.

A *partial ordered interpretation* $\mathcal{J}$ w.r.t. $\prec$ is a set of ground literals such that $A \in \mathcal{J}$ iff $-A \notin \mathcal{J}$, and if $+A \in \mathcal{J}$ (resp. $-A \in \mathcal{J}$) and $B \prec A$ then $+B \in \mathcal{J}$ or $-B \in \mathcal{J}$. A ground clause C is *false* in $\mathcal{J}$ if, for every literal $\pm A \in C$, the opposite literal $\mp A$ belongs to $\mathcal{J}$. A clause C is *unsatisfiable* in the partial interpretation $\mathcal{J}$ if there exists a substitution θ grounding for C such that $C\theta$ is false in $\mathcal{J}$. Let $\mathcal{C}_r$ be a set of rigid Horn clauses and $\mathcal{C}_f$ be a set of flexible Horn clauses which contains only flexible variables. We say that $\mathcal{C}_r$ is *unsatisfiable modulo* $\mathcal{C}_f$ if there exists a substitution θ grounding for $\mathcal{C}_r$ such that $\mathcal{C}_r\theta \cup \mathcal{C}_f$ is unsatisfiable.

Example 1. Consider the following set of Horn clauses:

$$\mathcal{C} := \{ \Rightarrow I(a);\ \Rightarrow I(b);\ I(x) \Rightarrow I(f(x));\ I(f(a)), I(f(b)) \Rightarrow \bot \}$$

If the clause $I(x) \Rightarrow I(f(x))$ is flexible then $\mathcal{C}$ is unsatisfiable whereas it is satisfiable if the clause is assumed to be rigid.

2.3 Ordered Resolution

We consider a liftable[1] ordering $\prec$, total on ground atoms. This property is crucial for the completeness of ordered resolution. Let A_1, A_2 and B be three atoms, C_1 and C_2 be Horn clauses and $\sigma = \mathsf{mgu}(A_1, A_2)$.
The *resolution rule* is defined by:

$$\frac{C_1 \stackrel{\mathsf{def}}{=} \Gamma_1 \Rightarrow A_1 \qquad \Gamma_2, A_2 \Rightarrow B \stackrel{\mathsf{def}}{=} C_2}{\Gamma_1\sigma, \Gamma_2\sigma \Rightarrow B\sigma}$$

Ordered resolution (w.r.t. $\prec$) requires that $A_1\sigma$ is *maximal* in $\Gamma_1\sigma, \Gamma_2\sigma \Rightarrow B\sigma$, *i.e.* there is no atom $A \in At(\Gamma_1\sigma, \Gamma_2\sigma \Rightarrow B\sigma)$ such that $A_1\sigma \prec A$. The clause $\Gamma_1\sigma, \Gamma_2\sigma \Rightarrow B\sigma$ is called the *resolvent* of C_1 and C_2. In the remainder of the paper, we will consider the embedding ordering $\preceq_{emb}$.

$t \preceq'_{emb} s$ if one of the following is true:

- $t = s$
- s is of the form $f(s_1, \ldots, s_n)$ and there exists some $1 \leq i \leq n$ s.t. $t \preceq'_{emb} s_i$.
- t is of the form $f(t_1, \ldots, t_n)$, s is of the form $f(s_1, \ldots, s_n)$ and for all $1 \leq i \leq n$: $t_i \preceq'_{emb} s_i$.

For instance, $f(x, z) \preceq_{emb} f(g(x, y), z)$. We assume the embedding ordering is extended to a total ordering. It is extended to atoms as follows: $P(t) \preceq_{emb} P'(t')$ if and only if $t \preceq_{emb} t'$. It can also be extended to clauses by considering clauses as multi-sets of atoms.

A set S of (flexible) clauses is *saturated* by ordered resolution w.r.t. $\prec_{emb}$ if for every resolvent C obtained by ordered resolution from S and for every partial ordered interpretation $\mathcal{J}$, if C is unsatisfiable in $\mathcal{J}$ then S is unsatisfiable in $\mathcal{J}$.

3 Modeling Protocols

The aim of this section is to introduce the class of clauses we consider. We also explain how protocols can be modeled using these clauses.

3.1 Intruder Clauses

The intruder is able to analyze messages that pass over the network. For example, if he sees an encrypted message and he knows the decryption key, he can decrypt it. This can be modeled by $I(\{x\}_y), I(y) \Rightarrow I(x)$. Intuitively, the predicate I represents the knowledge of the intruder and $I(m)$ means that the intruder

[1] An order $\prec$ is said *liftable* if for any two terms u, v and for any substitutions θ, $u \prec v$ implies $u\theta \prec v\theta$.

knows the message m. Other examples of clauses, very useful and classical to model the intruder capabilities are given below:

$$\mathcal{C}_{\mathsf{DY}} := \begin{cases} I(x),\ I(y) \Rightarrow I(\{x\}_y) & \text{symmetric encryption} \\ I(\{x\}_y),\ I(y) \Rightarrow I(x) & \text{symmetric decryption} \\ I(x),\ I(y) \Rightarrow I(\langle x, y\rangle) & \text{pairing} \\ I(\langle x, y\rangle) \Rightarrow I(x) & \text{first projection} \\ I(\langle x, y\rangle) \Rightarrow I(y) & \text{second projection} \\ I(x),\ I(y) \Rightarrow I(\mathsf{sgn}(x, y)) & \text{signing} \\ I(\mathsf{sgn}(x, y)),\ I(y) \Rightarrow I(x) & \text{verifying the signature} \end{cases}$$

Note that each of these clauses contains at most one function symbol. That is why we introduce the following definition.

Definition 1 (Intruder clause). *Let $\mathcal{P}$ be a set of unary predicate symbols. An* intruder clause *on $\mathcal{P}$ is a flexible Horn clause having a positive literal and is of the form*

$$\pm P_0(f(x_1, \ldots, x_n)) \vee \bigvee_{j=1}^{m} \pm P_j(x_{i_j}).$$

where

- *$x_{i_j} \in \{x_1, \ldots, x_n\}$ for every j such that $1 \leq j \leq m$, and*
- *$P_j \in \mathcal{P}$ for every j such that $0 \leq j \leq m$.*

If the atom of the form $P_0(f(x_1, \cdots, x_n))$ occurs positively then the intruder clause is called a constructor clause. *Otherwise it is called a* destructor clause.

Such kind of clauses do not allow us to model some cryptographic primitives which are useful. This is our main motivation to extend this class by adding some special clauses.

3.2 Extending the Intruder Power

We want to deal with some clauses that do not satisfy the conditions given in Definition 1. The intruder clauses $\mathcal{C}_{\mathsf{DY}}$ given in Section 3.1 represents the classical Dolev-Yao intruder [10], *i.e.* assuming *perfect cryptography*. In particular, it is assumed that an attacker can not learn anything from an encrypted message $\{m\}_k$ except if he knows the decryption key. This assumption is too strong in some situations. Depending on the implementation of the cryptographic primitives, the attacker may be able to deduce more messages.

Prefix and suffix properties. The prefix property is the ability of an intruder to get from an encrypted message the encryption of any of its prefixes: from a message $\{\langle x, y\rangle\}_z$, he can deduce the message $\{x\}_z$. This can be easily encoded in our formalism by the clause:

$$C_{\mathsf{pre}} := I(\{\langle x_1, x_2\rangle\}_{y_2}) \Rightarrow I(\{x_1\}_{y_2}).$$

This property strongly depends on the encryption algorithm. A relatively good method of encrypting several blocks of data is Cipher Block Chaining (CBC). In

such a system, the encryption of message block sequence $P_1P_2\cdots P_n$[2] with the key K is $C_0C_1\cdots C_n$ where $C_0 = I$ (initialization block) and $C_i = \{C_{i-1} \oplus P_i\}_K$. Hence, if $C_0C_1\cdots\cdots C_n = \{P_1P_2\cdots P_n\}_K$ then $C_0C_1C_2\cdots C_i = \{P_1P_2\cdots P_i\}_K$, that is to say an intruder can get $\{x\}_z$ from $\{\langle x, y\rangle\}_z$ if the length of x is a multiple of the block length used by the cryptographic algorithm. This property can be used by an intruder to mount some attacks on several well-known protocols [6] (*e.g.* Denning-Sacco protocol [8], Needham-Schroeder symmetric key protocol [16]). In [14], S. Kremer and M. Ryan notice that one can also reuse any postfix $C_{i+1}\cdots C_n$ of a cipher $C_0C_1C_2\cdots C_n$ as a valid cipher. This can also be modeled by a Horn clause:

$$C_{\mathsf{suf}} := I(\{\langle x_1, x_2\rangle\}_{y_2}) \Rightarrow I(\{x_2\}_{y_2}).$$

Blind signature. Digital signatures are often used in cryptographic protocols. A particular class of signature scheme is the blind signature scheme which is often used in electronic voting protocols (*e.g.* the protocol due to Fujioka *et al.* [11]). The idea of the protocol is that the voter first sends his vote hidden with a blinding factor r: $\mathsf{bld}(v, r)$. This is used to ensure that the value of his vote will not be revealed to anyone even the administrator who has to sign his vote. Then the administrator signs this message without knowing exactly the message he is signing and he sends back the result, *i.e.* $\mathsf{sgn}(\mathsf{bld}(v, r), ska)$ to the voter. Now the voter can unblind the message to obtain $\mathsf{sgn}(v, ska)$, *i.e.* the signature of his vote. This property of blind signature can be modeled in Horn clauses by considering the following clauses:

$$\mathcal{C}_{\mathsf{bld}} := \begin{cases} I(x),\ I(y) \Rightarrow I(\mathsf{bld}(x, y)) \\ I(\mathsf{bld}(x, y)),\ I(y) \Rightarrow I(x) \end{cases}$$

$$C_{\mathsf{sgn}} := I(\mathsf{sgn}(\mathsf{bld}(x_1, x_2), y_2), I(x_2)\ \Rightarrow\ I(\mathsf{sgn}(x_1, y_2)))$$

Note that the two clauses in $\mathcal{C}_{\mathsf{bld}}$ are intruder clauses whereas the last one is not. This is our main motivation to be able to deal with more complex clauses than the ones we have introduced previously.

Definition 2 (special clause). *Let $\mathcal{P}$ be a finite set of unary predicate symbols. A* special clause *on $\mathcal{P}$ is a flexible Horn clause of the form* $P_0(f_0(g(x_1, \ldots, x_p), y_2, \ldots, y_q)), P_1(x_{i_1}), \ldots, P_m(x_{i_m}) \Rightarrow P_{m+1}(f_0(x_{i_0}, y_2, \ldots, y_q))$ *where*

- $f_0 \neq g$,
- $x_{i_j} \in \{x_1, \ldots, x_p\}$ *for every* j *such that* $0 \leq j \leq m$, *and*
- $P_j \in \mathcal{P}$ *for every* j *such that* $0 \leq j \leq m + 1$.

Such a clause is said to be a j-special clause *when* $x_{i_0} = x_j$.

Example 2. The Horn clauses C_{pre} and C_{sgn} are 1-special clauses whereas C_{suf} is a 2-special clause.

[2] $P_1P_2P_3\cdots P_n$ should be written $\langle\ldots\langle\langle P_1, P_2\rangle, P_3\rangle, \cdots, P_n\rangle$.

3.3 Protocol Clauses

As a running example we consider the Needham-Schroeder symmetric key protocol [16]. However, note that our definition of a protocol clause (see Definition 3) is general enough to deal with many other protocols. This protocol aims at establishing a fresh shared symmetric key K_{ab} and mutually authenticating the participants: in every session, the value of K_{ab} has to be known only by the participants playing the roles of A, B and S in that session. Messages 1 to 3 perform the distribution of the fresh shared symmetric key K_{ab} and messages 4 and 5 are for mutual authentication of A and B. The fields N_a and N_b are nonces, *i.e.* random numbers, generated by A and B respectively. In the first message, A tells the server that he wants to communicate with B. The server replies with a message (message 2) which contains the nonce N_a, a fresh session K_{ab} and a ciphertext, namely $\{K_{ab}, A\}_{K_{bs}}$, that A has to forward to B. This is done in the third step. Finally, B challenges A by sending him a nonce N_b encrypted with the session key K_{ab} and A answers to this challenge.

$$
\begin{array}{lll}
1. & A \rightarrow S & : A, B, N_a \\
2. & S \rightarrow A & : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}} \\
3. & A \rightarrow B & : \{K_{ab}, A\}_{K_{bs}} \\
4. & B \rightarrow A & : \{N_b\}_{K_{ab}} \\
5. & A \rightarrow B & : \{\mathsf{succ}(N_b)\}_{K_{ab}}
\end{array}
$$

The operator succ represents the increment operation. Note that the clauses $I(x) \Rightarrow I(\mathsf{succ}(x))$ and $I(\mathsf{succ}(x)) \Rightarrow I(x)$ allow us to model the fact that the attacker is able to increment or decrement an integer. Such clauses are intruder clauses. In our setting, we can model the role of A played by a with the agent b as follows:

$$
\text{Role } A := \left\{ \begin{array}{r}
\Rightarrow I_1(\langle\langle a, b\rangle, n_a\rangle) \\
I_2(\{\langle\langle\langle n_a, b\rangle, X_2\rangle, X_3\rangle\}_{K_s(a)}) \Rightarrow I_3(X_3) \\
I_2(\{\langle\langle\langle n_a, b\rangle, X_2\rangle, X_3\rangle\}_{K_s(a)}), I_4(\{X_5\}_{X_2}) \Rightarrow I_5(\{\mathsf{succ}(X_5)\}_{X_2})
\end{array} \right.
$$

These rigid clauses represent an instance of role A. The role is executed by agent a who wants to communicate with b. Every variable is rigid and different clauses can share rigid variables. In such rules subscripts are used to ensure that messages sent at some step cannot be used to trigger a rule that has to be executed before. Hence, if we consider a single session of the protocol, a message m sent at the j^{th} step of the protocol would be denoted by $I_j(m)$. Moreover, in the third clause, we repeat the atom $I_2(\{\langle\langle\langle n_a, b\rangle, X_2\rangle, X_3\rangle\}_{K_s(a)})$ on the left hand side. This is useless in this particular situation. However, in some protocols, this can be used to ensure condition 1 of Definition 3.

Then to check whether the secrecy of a message m is preserved we add a Horn clause. For instance if we want to check the secrecy of the nonce n_a, we add the goal $I_5(n_a) \Rightarrow$ to the set of rigid Horn clauses modeling the protocol. Moreover, we can give some initial knowledge to the intruder by adding some clauses such as $I_0(a)$, $I_0(b)$, ... All these clauses are *protocol clauses*.

Definition 3 (Protocol clause). *A protocol clause is a rigid Horn clause that is either a goal clause or is of the form* $P_1(u_1), \ldots, P_n(u_n) \Rightarrow P_0(u_0)$ *where*

1. $vars(u_0) \subseteq vars(\{u_1, \ldots, u_n\})$, *and*
2. $P_0 \geq P_i$ *for every* i *such that* $1 \leq i \leq n$.

4 Security Problem

Definition 4 (intruder theory). *Let* I *be an unary predicate symbol. Let* $\mathcal{C}_I$ *be a finite set of intruder clauses built on* I *and* C_S *be a special clause built on* I *such that* $\mathcal{C}_I \cup \{C_S\}$ *is saturated by ordered resolution. The intruder theory, denoted* $\mathcal{I}_{\mathcal{P}}(\mathcal{C}_I \cup \{C_S\})$, *associated to* $\mathcal{C}_I \cup \{C_S\}$ *and built on* $\mathcal{P}$ *is the set of clauses which contains* $P_1(u_1), \ldots, P_n(u_n) \Rightarrow P_0(u_0)$ *if and only if*

- $P_0, \ldots, P_n \in \mathcal{P}$ *and* $P_0 \geq P_i$ *for each* i *such that* $1 \leq i \leq n$, *and*
- $I(u_1), \ldots, I(u_n) \Rightarrow I(u_0) \in \mathcal{C}_I \cup \{C_S,\ I(x) \Rightarrow I(x)\}$

The intruder knowledge always increases. Hence, if he knows message m at step k, he also knows m at step $k+1$. This is why we add the clause $I_i(x) \Rightarrow I_j(x)$ with $i \leq j$ to our intruder theory. We also assume that the operations available to the attacker are the same at each step.

It is easy to establish the following lemma which states that an intruder theory generated from a finite set of clauses that is saturated w.r.t. $\prec$ is also saturated w.r.t. $\prec$. This will allow us to not consider resolution steps between two flexible clauses in our resolution inference systems presented in Section 5.

Lemma 1. *Let* $\mathcal{C}_I$ *be a finite set of intruder clauses built on* I *and* C_S *be a special clause built on* I *such that* $\mathcal{C}_I \cup \{C_S\}$ *is saturated by ordered resolution w.r.t.* $\prec$. *We have that* $\mathcal{I}_{\mathcal{P}}(\mathcal{C}_I \cup \{C_S\})$ *is saturated by ordered resolution w.r.t.* $\prec$.

Example 3. $\mathcal{C}_{\mathsf{DY}} \cup \{C_{\mathsf{pre}}\}$, $\mathcal{C}_{\mathsf{DY}} \cup \{C_{\mathsf{suf}}\}$ and $\mathcal{C}_{\mathsf{DY}} \cup \mathcal{C}_{\mathsf{bld}} \cup \{C_{\mathsf{sgn}}\}$ are sets of clauses which are saturated by ordered resolution w.r.t. $\prec_{emb}$. Hence, they can be used to define an intruder theory.

In this paper we are interested in the so-called insecurity problem.

Insecurity problem

Input: A finite set $\mathcal{C}_P$ of protocol clauses built on $\mathcal{P}$, a finite set $\mathcal{C}_I$ of intruder clauses built on I and C_S be a special clause built on I such that $\mathcal{C}_I \cup \{C_S\}$ is saturated by ordered resolution. Let $\mathcal{I} = \mathcal{I}_{\mathcal{P}}(\mathcal{C}_I \cup \{C_S\})$.

Output: Is $\mathcal{C}_P$ unsatisfiable modulo $\mathcal{I}$? In other words, does there exist a substitution θ grounding for $\mathcal{C}_P$ such that $\mathcal{I} \cup \mathcal{C}_P\theta$ is unsatisfiable?

Attack on the Needham Shroeder symmetric key protocol with CBC. Beyond other existing attacks, O. Pereira and J.-J. Quisquater [17] presented the following flaw, based on the prefix property.

$$
\begin{array}{lll}
i.1 & a \rightarrow s & : a, b, n_a \\
i.2 & s \rightarrow a & : \{n_a, b, k_{ab}, \{k_{ab}, a\}_{k_{bs}}\}_{k_{as}} \\
ii.3 & I(b) \rightarrow a & : \{n_a, b\}_{k_{as}} \\
ii.4 & a \rightarrow I(b) & : \{n'_a\}_{n_a} \\
ii.5 & I(b) \rightarrow a & : \{\mathsf{succ}(n'_a)\}_{n_a}
\end{array}
$$

In a first session i, the attacker can listen to $\{n_a, b, k_{ab}, \{k_{ab}, a\}_{k_{bs}}\}_{k_{as}}$ and then, using the prefix property, he can compute $\{n_a, b\}_{k_{as}}$. This message can be sent at step 3 to the agent a who plays the role B in the session ii. Thus a can be fooled into accepting the publicly known nonce n_a as a secret key shared with b.

Such an attack can be retrieved in our formalism by considering one instance of the role B played by the agent a with b. This corresponds to the following two rigid clauses

$$I_3(\{\langle X, b\rangle\}_{K_s(a)}) \;\Rightarrow\; I_4(\{n'_a\}_X) \quad \text{and} \quad I_5(\{\mathsf{succ}(n'_a)\}_X) \Rightarrow I_6(\mathsf{end})$$

where end is a special constant used to model the fact that the agent a has executed his session until the end. We assume that the attacker has listened to a previous session between the two honest participants a and b, and has the following knowledge: $I_0(a),\ I_0(b),\ I_0(n_a),\ I_0(\{n_a, b, k_{ab}, \{k_{ab}, a\}_{K_s(b)}\}_{K_s(a)})$. Since a is playing a session with an honest agent b, if a executes his session until the end with b, the session key he received, represented by X, has to remain secret. Hence, we consider the following protocol clause in order to model the security property: $I_6(\mathsf{end}),\ I_6(X) \Rightarrow \bot$. We can easily show that the set of rigid clauses described above is unsatisfiable modulo $\mathcal{I}_{\mathcal{Q}}(\mathcal{C}_{\mathsf{DY}} \cup \{C_{\mathsf{pre}},\ I(x) \Rightarrow I(\mathsf{succ}(x))\})$ where $\mathcal{Q} = \{I_0, \ldots, I_6\}$.

The remainder of the paper is devoted to a proof that the insecurity problem is decidable. As a corollary, the insecurity problem in presence of a bounded number of sessions is decidable for the prefix intruder theory, for the suffix intruder theory and also for the blind signature intruder theory. Note that the conditions on what we have called a protocol clause are not restrictive w.r.t. our application. Hence, we can deal with a large class of cryptographic protocols. We first introduce and motivate our resolution method that is sound (Section 5). Then we establish termination (Section 6) and completeness (Section 7).

5 Inference Systems

First we present an inference system, $\mathcal{I}_{\mathsf{Ror}}$, which is a natural candidate to solve our problem (see Section 5.1). It is clearly sound and complete, but does not terminate (see Section 5.2). In Section 5.3 we provide our inference system, denoted by $\mathcal{I}_{\mathsf{Pr}}$, that allows us to solve the insecurity problem.

5.1 Constrained Rigid Ordered Resolution $\mathcal{I}_{\mathsf{Ror}}$

This inference system contains two kinds of inference rules. One allowing us to perform a resolution step between rigid clauses and two others allowing us to perform resolution steps between a rigid clause and a flexible one. Note that since we have assumed that our intruder theory is saturated w.r.t. $\prec$, we do not have to consider any resolution step between two intruder clauses.

Resolution involving two rigid clauses.

$$\frac{S \cup \{\Gamma_1 \Rightarrow A_1\ ;\ \Gamma_2, A_2 \Rightarrow B\}\ \ [\![C]\!]}{S \cup \{\Gamma_1 \Rightarrow A_1\ ;\ \Gamma_2, A_2 \Rightarrow B\ ;\ \Gamma_1, \Gamma_2 \Rightarrow B\}\ \ [\![C, A_1 = A_2]\!]}\ (\mathsf{RR_c})$$

Resolution involving a flexible clause.

$$\frac{S \cup \{\Gamma_2, A_2 \Rightarrow B\}\ [\![C]\!]}{S \cup \{\Gamma_2, A_2 \Rightarrow B\ ;\ \Gamma_1, \Gamma_2 \Rightarrow B\}\ [\![C, A_1 = A_2]\!]}\ (\mathsf{FR_c})$$

(i) $\Gamma_1 \Rightarrow A_1$ is a fresh renaming of a clause in $\mathcal{I}_{\mathcal{P}}(\mathcal{C}_I \cup \{C_S\})$, *(ii)* $A_1\sigma$ is strictly maximal in $(\Gamma_1 \Rightarrow A_1)\sigma$ where σ is the mgu of $C, A_1 = A_2$.

$$\frac{S \cup \{\Gamma_1 \Rightarrow A_1\}\ [\![C]\!]}{S \cup \{\Gamma_1 \Rightarrow A_1\ ;\ \Gamma_1, \Gamma_2 \Rightarrow B\}\ [\![C, A_1 = A_2]\!]}\ (\mathsf{RF_c})$$

(i) $\Gamma_2, A_2 \Rightarrow B$ is a fresh renaming of a clause in $\mathcal{I}_{\mathcal{P}}(\mathcal{C}_I \cup \{C_S\})$, *(ii)* $A_2\sigma$ is maximal in $(\Gamma_2, A_2 \Rightarrow B)\sigma$ where σ is the mgu of $C, A_1 = A_2$.

Definition 5 (constrained rigid global derivation). *A* constrained rigid global derivation *modulo an intruder theory $\mathcal{I}$ is a sequence $S_1\ [\![C_1]\!] \Rightarrow\!\!\!\!\Rightarrow \ldots \Rightarrow\!\!\!\!\Rightarrow S_n\ [\![C_n]\!]$ where each S_i is a set of rigid clauses and each C_i is a set of constraints. Moreover $S_{i+1}\ [\![C_{i+1}]\!]$ is obtained from $S_i\ [\![C_i]\!]$ by applying an inference rule of $\mathcal{I}_{\mathsf{Ror}}$.*

5.2 Difficulties with Termination

The first problem with termination is because we can perform a resolution step between a literal $I(X)$ of a rigid clause and an intruder clause such as $I(\langle x_1, x_2\rangle) \Rightarrow I(x_1)$. We obtain a new rigid literal $I(X_1)$ on which we can apply exactly the same inference rule. Hence a first solution consists in removing the possibility to perform a resolution step between a flexible clause and a rigid one when this allows us to introduce new rigid variables. This can be done by adding some side conditions on the rules (RF) and (FR). We retrieve termination but we lose completeness as illustrated by the example below:

Example 4. Consider the intruder theory made up of the rule $\mathcal{C}_{\mathsf{DY}} \cup \{C_{\mathsf{pre}}\}$ and the following set of rigid clauses:

$$\mathcal{C}_P := \left\{ \begin{array}{ll} \Rightarrow I(a) & I(X) \Rightarrow I(\{X\}_{\mathsf{h}(\langle a,b\rangle)}) \\ \Rightarrow I(b) & I(\{a\}_{\mathsf{h}(X)}) \Rightarrow \end{array} \right.$$

This set is unsatisfiable: consider the following substitution $\theta = \{X \mapsto \langle a, b\rangle\}$. However, we are not able to derive the empty clause with the inference system $\mathcal{I}_{\mathsf{Ror}}$ if we forbid resolution steps that introduce new rigid variables (*e.g.* the one with $I(X) \Rightarrow I(\{X\}_{\mathsf{h}(\langle a,b\rangle)}$ and C_{pre}). To retrieve completeness we have to introduce new inference rules that we have called *instantiation rules.*

5.3 Our Resolution Method: Protocol Resolution

By taking into account all these considerations, we obtain the inference system $\mathcal{I}_{\mathsf{Pr}}$ described below.

Resolution involving two rigid clauses.

$$\frac{S \cup \{\Gamma_1 \Rightarrow A_1 \ ; \ \Gamma_2, A_2 \Rightarrow B\}}{(S \cup \{\Gamma_1 \Rightarrow A_1 \ ; \ \Gamma_2, A_2 \Rightarrow B \ ; \ \Gamma_1, \Gamma_2 \Rightarrow B\})\sigma} \ (\mathsf{RR_p})$$

where $\sigma = \mathsf{mgu}(A_1, A_2)$.

$$\frac{S \cup \{\Gamma_1 \Rightarrow P(f_0(X, t_2, \ldots, t_q)) \ ; \ \Gamma_2, P(f_0(t'_1, \ldots, t'_q)) \Rightarrow B\}}{(S \cup \{\Gamma_1 \Rightarrow P(f_0(X, t'_2, \ldots, t_q)) \ ; \ \Gamma_2, P(f_0(t'_1, \ldots, t'_q)) \Rightarrow B\})\sigma} \ (\mathsf{RR_p} - \mathsf{Inst1})$$

where σ is the mgu of $\{t_i = t'_i \mid 2 \leq i \leq q\}$.

$$\frac{S \cup \{\Gamma_1 \Rightarrow P(f_0(X, t_2, \ldots, t_q)) \ ; \ \Gamma_2, P(t) \Rightarrow B\}}{(S \cup \{\Gamma_1 \Rightarrow P(f_0(X, t_2, \ldots, t_q)) \ ; \ \Gamma_2, P(t) \Rightarrow B\})\sigma} \ (\mathsf{RR_p} - \mathsf{Inst2})$$

where $\sigma = \mathsf{mgu}(t, t')$ and $t' \preceq_{emb} t_i$ for some $i \in \{2, \ldots, q\}$.

Resolution involving a flexible clause.

$$\frac{S \cup \{\Gamma_2, A_2 \Rightarrow B\}}{(S \cup \{\Gamma_2, A_2 \Rightarrow B \ ; \ \Gamma_1, \Gamma_2 \Rightarrow B\})\sigma} \ (\mathsf{FR_p})$$

where *(i)* $\Gamma_1 \Rightarrow A_1$ is a fresh renaming of a clause in $\mathcal{I}_{\mathcal{P}}(\mathcal{C}_I \cup \{C_S\})$, *(ii)* $\sigma = \mathsf{mgu}(A_1, A_2)$, *(iii)* $A_1\sigma$ is strictly maximal in $(\Gamma_1 \Rightarrow A_1)\sigma$, and *(iv)* A_2 is of the form $P(t)$ with $t \notin \mathcal{X}$.

$$\frac{S \cup \{\Gamma_1 \Rightarrow A_1\}}{(S \cup \{\Gamma_1 \Rightarrow A_1 \ ; \ \Gamma_1, \Gamma_2 \Rightarrow B\})\sigma} \ (\mathsf{RF_p})$$

where *(i)* $\Gamma_2, A_2 \Rightarrow B$ is a fresh renaming of a clause in $\mathcal{I}_{\mathcal{P}}(\mathcal{C}_I \cup \{C_S\})$, *(ii)* $\sigma = \mathsf{mgu}(A_1, A_2)$, *(iii)* $A_2\sigma$ is maximal in $(\Gamma_2, A_2 \Rightarrow B)\sigma$, and *(iv)* A_1 is of the form $P(t)$ with $t \notin \mathcal{X}$. Moreover, if $\Gamma_2, A_2 \Rightarrow B$ is a special clause, we require that t is not of the form $f_0(t_1, \ldots, t_q)$ with $t_1 \in \mathcal{X}$.

Definition 6 (rigid global derivation). *A* rigid global derivation *modulo an intruder theory $\mathcal{I}$ is a sequence $S_1 \Mapsto \ldots \Mapsto S_n$ where each S_i is a set of rigid clauses. Moreover S_{i+1} is obtained from S_i by applying an inference rule of $\mathcal{I}_{\mathsf{Pr}}$.*

By inspection of each inference rule, it is easy to establish the following result.

Proposition 1 (soundness). *Let $\mathcal{C}_P$ a finite set of protocol clauses built on $\mathcal{P}$ and $\mathcal{I}_{\mathcal{P}}(\mathcal{C}_I \cup \{C_S\})$ be the intruder theory associated to the finite set of intruder clauses $\mathcal{C}_I$ and the special clause C_S. Let $S_0 = \mathcal{C}_P$ and consider a rigid global derivation $S_0 \Mapsto S_1 \ldots \Mapsto S_n$ modulo $\mathcal{I}_{\mathcal{P}}(\mathcal{C}_I \cup \{C_S\})$. If $\Box \in S_i$ for some $i \leq n$ then $\mathcal{C}_P$ is unsatisfiable modulo $\mathcal{I}_{\mathcal{P}}(\mathcal{C}_I \cup \{C_S\})$.*

6 Termination

Proposition 2 (termination). *Let $\mathcal{C}_P$ be a finite set of protocol clauses built on $\mathcal{P}$ and $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$ be the intruder theory associated to the finite set of intruder clauses $\mathcal{C}_I$ and the special clause C_S. Every rigid global derivation w.r.t. $\mathcal{I}_{\mathsf{Pr}}$ issued from $\mathcal{C}_P$ has a finite length.*

7 Completeness

In this section, we will show that the Protocol Resolution inference system $\mathcal{I}_{\mathsf{Pr}}$ is complete. We will consider an unsatisfiable set of protocol clauses $\mathcal{C}_P$, meaning that there exists a substitution θ such that $\mathcal{C}_P\theta$ is unsatisfiable modulo a given intruder theory $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$. We will prove that if we cannot derive the empty clause from $\mathcal{C}_P$ using $\mathcal{I}_{\mathsf{Pr}}$ then there must be a *smaller* substitution σ witnessing the fact that $\mathcal{C}_P$ is unsatisfiable modulo $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$. If this notion of smaller is well-founded, this allows us to conclude the completeness of $\mathcal{I}_{\mathsf{Pr}}$.
First of all, we show that our inference system can only produce protocol clauses.

Lemma 2. *Let $\mathcal{C}_P$ be a finite set of protocol clauses built on $\mathcal{P}$ and $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$ be the intruder theory associated to the finite set of intruder clauses $\mathcal{C}_I$ and the special clause C_S. Let $\mathcal{C}'_P$ be such that $\mathcal{C}_P \Mapsto \mathcal{C}'_P$ and $C \in \mathcal{C}'_P$. Then C is a protocol clause.*

The standard definition of a set of clauses S to be saturated is that all inferences applied to S are redundant. We define *θ-pre-saturated*, for some substitution θ, to mean that all inferences *consistent with* θ are redundant.

Definition 7. *Let S be a set of protocol clauses. A clause C is redundant in S if there are some clauses $C_1, \ldots, C_n$ in S s.t. $C_i \preceq_{emb} C$ for all $i \leq n$ and $C_1, \ldots, C_n \models C$. S is saturated if for every S' s.t. $S \Mapsto S'$ and for every $C' \in S'$, we have that C' is redundant in S.*

Definition 8. *Let S be a set of protocol clauses and θ be a ground substitution. An inference is said to be* θ-consistent *if $\sigma \leq \theta$, where σ is the mgu of the inference. The set S is said to be* θ-pre-saturated *if for every set S' such that $S \Mapsto S'$ by a θ-consistent inference, for every $C' \in S'$, C' is redundant in S. A rigid global derivation is a* θ-derivation *if all the inferences involved in the derivation are θ-consistent.*

If the θ instance of a set of protocol clauses $\mathcal{C}_P$ is unsatisfiable modulo an intruder theory $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$, then the θ instance of a derived set of clauses is also unsatisfiable modulo $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$.

Lemma 3. *Let $\mathcal{C}_P$ be a set of protocol clauses and $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$ be an intruder theory. Let θ be a grounding substitution w.r.t. $\mathcal{C}_P$ witnessing the fact that $\mathcal{C}_P$ is unsatisfiable modulo $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$ and consider $\mathcal{C}_P \Mapsto \mathcal{C}'_P$ a θ-consistent inference. Then $\mathcal{C}'_P$ is unsatisfiable modulo $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$ and θ is a witness of this fact.*

For the completeness proof, we will need to define an ordering on variables, given a substitution, and we also define an ordering on substitutions.

Definition 9. *Let θ and θ' be substitutions for a set of variables V. A variable X is called θ-maximal if there is no Y such that $X\theta \prec Y\theta$. We define $\theta \prec \theta'$ if (i) For all x in V, $x\theta \prec x\theta'$ or $x\theta = x\theta'$. (ii) There is at least one variable y in V s.t. $y\theta \prec y\theta'$.*

Suppose that $\mathcal{C}_P$ is a set of rigid clauses, such that $\mathcal{C}_P$ is unsatisfiable modulo $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$, and $\mathcal{C}_P$ is presaturated under θ. In the full paper we give a series of lemmas to specify in which positions a maximal variable X can appear in a shortest proof of the unsatisfiability of C_P modulo $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$. A maximal variable can appear in a quite restricted number of places. Given that the number of places is restricted, it will be easy to show that the value of $X\theta$ can be changed and unsatisfiability modulo $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$ will be preserved. The series of lemmas culminates in the following lemma which shows that, given the above conditions, when the empty clause is not in $\mathcal{C}_P$, it is possible to replace θ by a smaller substitution σ, such that $\mathcal{C}_P\sigma$ is still unsatisfiable. The proof of the lemma is based on the fact that the substitutions in $\mathcal{I}_{\mathsf{Ror}}$ are stored in constraints. If we have a proof of unsatisfiability, then the constraints in the proof can be changed, so that the value of $X\theta$ is smaller, resulting in a new substitution σ, and we now have a proof of unsatisfiability of $\mathcal{C}_P\sigma$. The proof system for the new proof will be slightly different, and one that we might not want to use in practice. But it will be a sound inference system, and that is all we need to prove unsatisfiability.

Lemma 4. *Suppose that $\mathcal{C}_P$ is unsatisfiable modulo $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$ and let θ be a substitution witnessing this fact. Assume that the empty clause does not exist in the θ-pre-saturation of $\mathcal{C}_P$ using protocol resolution modulo $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$. Then there exists a smaller substitution σ witnessing the fact that $\mathcal{C}_P$ is unsatisfiable modulo $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$.*

The previous lemma is the crucial lemma to prove completeness. We showed that when the empty clause is not generated, then we can find a smaller substitution which preserves unsatisfiability. But since the ordering is well-founded, we cannot continually find smaller substitutions, which means that there must be some substitution for which our inference rules will deduce the empty clause.

Theorem 1 (completeness). *Let $\mathcal{C}_P$ be a finite set of protocol clauses built on $\mathcal{P}$ and $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$ be the intruder theory associated to the finite set of intruder clauses $\mathcal{C}_I$ and the special clause C_S. Let $S_0 = \mathcal{C}_P$. If there exists a substitution θ grounding for $\mathcal{C}_P$ witnessing the fact that $\mathcal{C}_P$ is unsatisfiable modulo $\mathcal{I}_\mathcal{P}(\mathcal{C}_I \cup \{C_S\})$ then there exists a rigid global derivation $S_0 \Rightarrow S_1 \ldots \Rightarrow S_n$ such that $\Box \in S_n$.*

8 Conclusion

We have defined the problem of rigid theorem proving modulo a flexible theory. We showed how to use that framework to model cryptographic protocol insecurity problems. Then we gave a resolution-based decision procedure for solving such problems in some interesting protocol theories.

Recently several procedures for deciding trace-based security properties have been proposed for a bounded [4,5,15] or unbounded number of sessions [3,7]. Several years ago, most of the work relied on the so called *perfect cryptography assumption*: one cannot learn anything from an encrypted message except if one knows the decryption key. Many papers relax this assumption in order to be closer to the implementation and to model more sophisticated cryptographic primitives. Among the works cited above, the closest to ours are [7,13] since they used a formalism based on Horn clauses. However, they only used flexible clauses. In [7], this allows them to deal with the insecurity problem in presence of an unbounded number of sessions but for a class of protocols more restrictive than ours (*single blind copying*). In [13] they only use flexible Horn clauses to deal with a bounded number sessions and hence they have to perform some approximation. Our generic result allows us to deal with a class of intruder theories similar to the one described in [2] and also to retrieve the decidability result about the prefix property that has been obtained in [4]. Both of these results [4,2] have been obtained in a completely different formalism.

Future work is to extend our idea to other interesting cryptographic protocol theories. In particular, we would like to add equality to our inference system to handle properties of cryptographic algorithms. We also plan to implement our idea. There are several possible ways to implement it. We could implement it exactly along the lines of this paper, as a nondeterministic procedure. It is also possible to save the constraints with each clause, and to only allow the empty clause if the constraint is satisfiable. The advantage of this method is that it could use a standard deterministic resolution-style inference procedure. Finally, we plan to implement it as an extension of the method given in [9], where rigid theorem proving problems are solved by encoding proofs as SAT problems.

References

1. Andrews, P.: Theorem proving via general matings. Journal of the ACM 28(2), 193–214 (1981)
2. Bernat, V., Comon-Lundh, H.: Normal proofs in intruder theories. In: ASIAN 2006. Proc. of 11th Asian Computing Science Conference, Tokyo, Japan. LNCS, Springer, Heidelberg (2006)
3. Blanchet, B.: An efficient cryptographic protocol verifier based on prolog rules. In: CSFW 2001. Proc. of 14th Computer Security Foundations Workshop, Cape Breton (Canada), pp. 82–96. IEEE Computer Society Press, Los Alamitos (2001)
4. Chevalier, Y., Küsters, R., Rusinowitch, M., Turuani, M.: An NP decision procedure for protocol insecurity with XOR. In: LICS 2003. Proc. of 18th Annual IEEE Symposium on Logic in Computer Science, Ottawa (Canada), pp. 261–270. IEEE Computer Society Press, Los Alamitos (2003)

5. Comon-Lundh, H., Shmatikov, V.: Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In: LICS 2003. Proc. of 18th Annual IEEE Symposium on Logic in Computer Science, Ottawa (Canada), pp. 271–280. IEEE Computer Society Press, Los Alamitos (2003)
6. Cortier, V., Delaune, S., Lafourcade, P.: A survey of algebraic properties used in cryptographic protocols. Journal of Computer Security 14(1), 1–43 (2006)
7. Cortier, V., Rusinowitch, M., Zalinescu, E.: A resolution strategy for verifying cryptographic protocols with cbc encryption and blind signatures. In: PPDP 2005. Proc. of 7th ACM-SIGPLAN International Conference on Principles and Practice of Declarative Programming, Lisboa (Portugal), pp. 12–22. ACM Press, New York (2005)
8. Denning, D.E., Sacco, G.M.: Timestamps in key distribution protocols. Communications of the ACM 24(8), 533–536 (1981)
9. Deshane, T., Hu, W., Jablonski, P., Lin, H., Lynch, C., McGregor, R.E.: Encoding first order proofs in sat. In: Pfenning, F. (ed.) CADE 2007. LNCS, vol. 4603, pp. 476–491. Springer, Heidelberg (2007)
10. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions in Information Theory 2(29), 198–208 (1983)
11. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: Zheng, Y., Seberry, J. (eds.) AUSCRYPT 1992. LNCS, vol. 718, pp. 244–251. Springer, Heidelberg (1993)
12. Goubault, J.: The complexity of resource-bounded first-order classical logic. In: Enjalbert, P., Mayr, E.W., Wagner, K.W. (eds.) STACS 94. LNCS, vol. 775, pp. 59–70. Springer, Heidelberg (1994)
13. Jacquemard, F., Rusinowitch, M., Vigneron, L.: Tree automata with equality constraints modulo equational theories. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 557–571. Springer, Heidelberg (2006)
14. Kremer, S., Ryan, M.: Analysing the vulnerability of protocols to produce known-pair and chosen-text attacks. In: SecCo 2004. Proc. 2nd International Workshop on Security Issues in Coordination Models, Languages and Systems, London, UK. ENTCS, Elsevier Science Publishers, Amsterdam (2005)
15. Millen, J., Shmatikov, V.: Symbolic protocol analysis with an abelian group operator or Diffie-Hellman exponentiation. Journal of Computer Security 13(3), 515–564 (2005)
16. Needham, R., Schroeder, M.: Using encryption for authentification in large networks of computers. Communications of the ACM 21(12), 993–999 (1978)
17. Pereira, O., Quisquater, J.-J.: On the perfect encryption assumption. In: WITS 2000. Proc. of 1st Workshop on Issues in the Theory of Security, Geneva (Switzerland), pp. 42–45 (2000)