Intruder Deductions, Constraint Solving and Insecurity Decision in Presence of Exclusive or

H. Comon-Lundh
LSV, INRIA and ENS Cachan
61 Avenue du Président Wilson
94235 Cachan cedex France
Hubert.Comon@lsv.ens-cachan.fr

V. Shmatikov
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025 U.S.A.
shmat@csl.sri.com

Abstract

We present decidability results for the verification of cryptographic protocols in the presence of equational theories corresponding to `xor` *and Abelian groups. Since the perfect cryptography assumption is unrealistic for cryptographic primitives with visible algebraic properties such as* `xor`*, we extend the conventional Dolev-Yao model by permitting the intruder to exploit these properties. We show that the ground reachability problem in NP for the extended intruder theories in the cases of* `xor` *and Abelian groups. This result follows from a normal proof theorem. Then, we show how to lift this result in the* `xor` *case: we consider a symbolic constraint system expressing the reachability (*e.g.*, secrecy) problem for a finite number of sessions. We prove that such constraint system is decidable, relying in particular on an extension of combination algorithms for unification procedures. As a corollary, this enables automatic symbolic verification of cryptographic protocols employing* `xor` *for a fixed number of sessions.*

1 Introduction

In this paper, we demonstrate that the reachability problem is decidable for cryptographic protocols that employ primitives with equational properties corresponding to `xor` and arbitrary Abelian groups. While the reachability problem for cryptographic protocols has received a lot of attention, in most approaches [2, 16, 4, 13] the underlying cryptographic primitives are modeled, following the so called "Dolev-Yao" model [7], as abstract data types without any algebraic properties. This simplification may be realistic for information-theoretically secure crypto schemes, but it is not valid for primitives such as `xor`, products, etc.

Algebraic properties of modular exponentation are exploited by popular cryptographic protocols such as those based on group Diffie-Hellman [19, 11]. Vernam cipher and cipher-block chaining mode for block ciphers rely on `xor` [12]. Their properties—associativity, commutativity, cancellation with inverses—are *not*, in general, hidden from the intruder, and abstracting away the intruder's ability to exploit them results in missed attacks. For example, the original version of Bull's recursive authentication protocol was formally proved correct in the Dolev-Yao model, but the protocol used `xor` for encryption and was thus vulnerable to an attack that exploited the self-cancellation property [15, 17]. Automatic verification of such protocols *must* take into account the fact that the cryptographic primitives obey certain equational theories. Though some procedures have been designed for verifying protocols in the presence of algebraic properties (*e.g.*, [11]), there was no relevant decision result in this case until now. The main contribution of this paper is to develop decision results for protocol insecurity in the presence of relevant equational theories.

Similar results have been obtained independently in [5] and are published in these proceedings. In that paper, the authors show that, for a finite number of sessions, the protocol insecurity problem is NP-complete in presence of XOR. In section 1.3, we compare their results and their techniques with ours.

1.1 Intruder deduction problem

The first question in automating cryptographic protocol verification is the following *intruder deduction problem*:

> Given a finite set of messages T and (presumably) secret s, can the intruder deduce s from T?

Formally, messages are terms over a finite alphabet, which includes, for instance, encryption. We write $T \overset{?}{\vdash} s$ as a shorthand for the intruder deduction problem. This decision problem depends on the deduction capabilities of the intruder. The most widely used deduction relation in this context is shown in figure 1 (for the case of a symmetric-key cipher) and known, following [7], as the *Dolev-Yao model*: the intruder can form pairs and ciphertexts from known terms, decompose pairs, and decrypt ciphertexts only when he knows the decryption key. This assumes *perfect cryptography*: the set of messages is supposed to be a free algebra, which, of course, is an unrealistic hypothesis as many cryptographic primitives do have some algebraic properties.

$T \stackrel{?}{\vdash} s$ can be decided in polynomial time for the Dolev-Yao intruder (*i.e.*, the deduction rules of figure 1). We observe that this result can be easily derived from a theorem by D. McAllester [10]: the intruder theory given by the rules of figure 1 is *local*. This means that $T \vdash s$ *iff* there is a proof which only involves subterms of terms in T, s. This is true if we add any set of function symbols with rules allowing the intruder to apply these function symbols to any term and, for some of them, to decompose the terms. We may also add other encryption functions, in particular those modeling a public-key cipher. It is also not hard to show that $T \stackrel{?}{\vdash} s$ is complete for PTIME.

In the first part of this paper, we consider the same question, relaxing, however, the perfect cryptography assumption. Our main motivation is to develop a decision technique for protocol security that will not miss attacks such as the Ryan-Schneider attack on the recursive authentication protocol [17] in which the attacker exploits algebraic properties of the encryption scheme (see section 4).

We introduce the $\oplus$ symbol, which is interpreted either as `xor`, or as an arbitrary Abelian group operator. We prove a locality theorem for the new set of deduction rules. However, due to the associativity and commutativity of $\oplus$, locality only implies that $T \stackrel{?}{\vdash} s$ is in NP.

Pairing (P) $$\frac{T \vdash u \quad T \vdash v}{T \vdash \langle u, v\rangle}$$

Encryption (E) $$\frac{T \vdash u \quad T \vdash v}{T \vdash [u]_v}$$

Unpairing (UL,UR) $$\frac{T \vdash \langle u, v\rangle}{T \vdash u} \qquad \frac{T \vdash \langle u, v\rangle}{T \vdash v}$$

Decryption (D) $$\frac{T \vdash [u]_v \quad T \vdash v}{T \vdash u}$$

Axioms (A) $$\frac{}{T \vdash u} \quad \text{if } u \in T$$

Figure 1. The Dolev-Yao intruder capabilities

1.2 Protocol verification via constraint solving

Cryptographic protocols can be seen as finite sequences of rules stating "If a principal A receives a message u, then she emits a message v". However, A may not have access to the entire message u. Assume for instance that $u = [[m]_{K_1}]_{K_2}$ (a message m cyphered using K_1 and then K_2) and that A knows the inverse of K_2, but not the inverse of K_1. Then A cannot distinguish this message from any $[m']_{K_2}$ since $[m]_{K_1}$ looks like any randomly generated message. This is modeled using terms with variables: u is written $[x]_{K_2}$, parts which cannot be further decomposed being abstracted with variables. Now, the protocol rules can be seen as pairs of terms with variables u, v, which must be read: "If A receives a message matching u, then she emits the corresponding message v". That is where an intruder can mislead a principal, replacing an instance of u with another instance that he was able to build.

Then, a sequence of messages $m_1, \ldots, m_k$ forms a valid protocol trace if each m_i is an instance $v\sigma$ such that the intruder can deduce $u\sigma$ for some protocol rule (u, v) (and the principal emitting m_i was in a state expecting a message $u\sigma$).

Protocol insecurity problem can thus be stated as finding a sequence of protocol rules $(u_1, v_1), \ldots, (u_n, v_n)$ (rules can be repeated, using distinct variables, for multi-sessions) and a substitution σ such that, for every $i < n$

$$T_0, v_1\sigma, \ldots, v_{i-1}\sigma \vdash u_i\sigma \tag{1}$$

and

$$T_0, v_1\sigma, \ldots, v_n\sigma \vdash s \tag{2}$$

for a given s which was expected to be kept secret. Such a formalization can be found in, e.g., [2, 9, 4, 13] and can be extended to trace-based properties other than secrecy.

In general finding such an n and a σ is undecidable. Undecidability results from several factors: the intruder's ability to generate fresh random data (nonce generation), the unboundedness of the number of sessions, the ability to form pairs, and the unboundedness of term sizes. Only the unbounded number of sessions is essential for undecidability. Removing one of the other three is not sufficient for decidability [6, 1, 8]. This is why the problem of reachability with a *bounded number of sessions* has drawn some attention. In this setting, n is bounded, we can guess the rules which are played at every step and reduce the insecurity problem to finding a substitution σ satisfying (1) and (2).

Several authors showed that reachability is decidable for a bounded number of sessions, *assuming perfect cryptography*. This line of research culminates with M. Rusinowitch and M. Turuani's result that this problem is co-NP-complete [16]. The decision technique used in [16, 13] consists, roughly, in first guessing an interleaving of the sessions, thus reducing the question to a single session, then expressing the problem as a constraint solving problem: $T_1 \overset{?}{\vdash} u_1, ..., T_n \overset{?}{\vdash} u_n$ where each individual constraint $T_i \overset{?}{\vdash} u_i$ is a lifting of the ground intruder deduction problem to terms with variables. This constraint solving problem is proved to be in NP if the intruder's capabilities are modeled by the deduction rules of figure 1.

In the second part of this paper, we introduce a new symbolic constraint system in which we can express the constraints $T_i \overset{?}{\vdash} u_i$ and prove that these constraints are decidable in the presence of `xor`. As a consequence, protocol insecurity is decidable for a bounded number of sessions.

Constraint solving proceeds in several steps. The first one requires an extended narrowing procedure: roughly, we want to anticipate all possible simplifications occurring after any instantiation or combination of subterms of the original constraint. Our algorithm is inspired by combination techniques for unification algorithms [3, 18] and unification modulo ACUN [14]. In the future, it should be possible to extend it to other equational theories, typically the theory of Abelian groups, for which in this paper we already prove NP-membership in the ground case.

1.3 Comparison with [5] (these proceedings)

In [5], the authors also consider the intruder deductions (ground reachability). They show a stronger result for the xor case: they prove that this problem is in PTIME, whereas we only prove an NP membership. On the other hand, we also consider the case of Abelian groups.

Now, for the protocol insecurity, in [5], the authors consider a restricted class of protocol rules: for instance a protocol containing the two following rules:

$A \to B: \quad N_A \oplus N'_A$

> A generates two random numbers (nonces) N_A and N'_A and sends to B their xor $N_A \oplus N'_A$

$B \to A: \quad N_A \oplus N_B$

> B generates a nonce N_B and replies sending $N_A \oplus N_B$

is dismissed in [5]. Such assumptions look reasonable since the principal B cannot retrieve N_A from $N_A \oplus N'_A$ without knowing N'_A.

All rules of figure 1 and, in addition:

Xor (X) $$\frac{T \vdash u \quad T \vdash v}{T \vdash u \oplus v}$$

Inversion (I) $$\frac{T \vdash u}{T \vdash I(u)}$$

Equality $$\frac{T \vdash t \quad t =_E u}{T \vdash u}$$

Figure 2. Intruder capabilities with equational theory

Our result does not assume any such restriction. In this respect, we prove a more general decidability result. However, in [5], they get an NP upper bound, which we don't have. They also abstract out intruder rules using oracles, and provide an example of application of such oracles.

In order to prove the NP decision result, [5] shows that, if there is an attack, then there is a polynomial size attack. We proceed in a completely different way. The protocol insecurity problem is expressed as a constraint satisfaction problem as explained above. We provide constraint solving rules, yielding a complete set of solved forms. Then *all* attacks are solutions of these solved forms and the protocol is insecure iff there is at least one solved form, which implies the decidability of protocols insecurity with `xor`. We hope that this technique can be generalized to other equational theories.

2 Intruder deductions in the presence of $\oplus$

2.1 The setting

We assume that messages are terms built over an alphabet $\mathcal{F}$ of function symbols containing constants, pairing $\langle \_, \_ \rangle$, encryption $[\_]_\_$, a unary symbol $I(\ )$ and a binary function symbol $\oplus$ used in infix notation. In this paper we will only consider symmetric-key encryption, but everything can be extended to several encryption symbols, including public/private-key cryptosystems and to other free symbols as well, such as one-way functions.

We extend the intruder deduction rules of figure 1 as displayed in figure 2.

For $=_E$, we consider the congruence generated by the equations of figure 3, which we will call AG (for *Abelian groups*) or the congruence generated by these equations and

$$\begin{array}{rcl} (x \oplus y) \oplus z & = & x \oplus (y \oplus z) \\ x \oplus y & = & y \oplus x \\ 0 \oplus x & = & x \\ x \oplus I(x) & = & 0 \\ I(I(x)) & = & x \\ I(x \oplus y) & = & I(y) \oplus I(x) \\ I(0) & = & 0 \end{array}$$

Figure 3. $\oplus$ axioms

the equation $I(x) = x$, which will be called the $\mathtt{xor}$ *theory*.

If we orient from left to right the equations of figure 3 other than associativity and commutativity, adding the extension $y \oplus x \oplus I(x) \to y$, we get a convergent rewrite system modulo associativity and commutativity of $\oplus$. This is also the case if we add the rule $I(x) \to x$ and simplify all the rules accordingly. Hence every term t has a unique normal form $t\downarrow$ up to associativity and commutativity.

If T is a finite set of terms, let $\mathsf{St}(T)$ be the least set of terms such that:

- If $t \in T$ then $t \in \mathsf{St}(T)$
- If $t \in \mathsf{St}(T)$ then $I(t)\downarrow \in \mathsf{St}(T)$
- If $\langle u, v\rangle \in \mathsf{St}(T)$ then $u, v \in \mathsf{St}(T)$
- If $[u]_v \in \mathsf{St}(T)$ then $u, v \in \mathsf{St}(T)$
- If $u_1 \oplus \ldots \oplus u_n \in \mathsf{St}(T)$ and $u_1, \ldots, u_n$ are not headed with $\oplus$, then $u_1, \ldots, u_n \in \mathsf{St}(T)$

When $I(x) = x$, $\mathsf{St}(T)$ is exactly the subterms of the *flatten* forms of terms in T, *i.e.*, the subterms when $\oplus$ is viewed as a variadic function symbol. In all cases, the number of elements in $\mathsf{St}(T)$ is linear in the size of T (the *size* of a set of terms is defined, as usual, as the sum of the number of nodes in each member of T).

Our goal in this section is to show that, if $T \vdash s$, then there is a normal form proof in which only terms in $\mathsf{St}(T \cup \{s\})$ appear. For example, one way to derive a from $\{a \oplus b, b, c\}$ is to construct $\langle b, c\rangle$, then $a \oplus b \oplus \langle b, c\rangle$, then $a \oplus \langle b, c\rangle$, and finally a. But this proof is not minimal since it uses the "unnecessary" term $\langle b, c\rangle$. The minimal proof consists in simply adding $I(b)$ and $a \oplus b$ to obtain a.

Since we will need to gather the $\oplus$ rule applications together, we replace the former (X) rule with the following general $\oplus$ rule, whose number of premises is unbounded:

$$\text{(GX)} \quad \frac{T \vdash u_1 \quad \ldots \quad T \vdash u_n}{T \vdash u_1 \oplus \ldots \oplus u_n}$$

Define $\mathsf{SS}(T)$ to be the closure of $\mathsf{St}(T)$ under $\oplus$:

$$\mathsf{SS}(T) = \{t_1 \oplus \ldots \oplus t_n \mid \{t_1, \ldots, t_n\} \subseteq \mathsf{St}(T)\}$$

Assuming that T, s are in normal form, if there is a proof of $T \vdash s$, we get another proof by normalizing the terms at each inference step: normalizing cannot prevent A, UL, UR or D and for the other rules equality steps can be pushed after their application. Therefore, we can assume that all terms are kept in normal form, a normalization step taking place after each inference rule. That is why, from now on, the equality rule will be implicit:

Definition 1 *A* proof *of $T \vdash u$ is a tree labeled with sequents $T \vdash v$ such that the root is labeled with $T \vdash v$ and every node labeled with $T \vdash v$ has n sons $T \vdash s_1, \ldots, T \vdash s_n$ such that* $\dfrac{T \vdash s_1 \cdots T \vdash s_n}{T \vdash w}$ *is an instance of one of the rules UL,UR,A,GX,E,P,D,I and $w\downarrow = v$.*

The *size* of a proof is the number of its nodes. We say that a proof is *simple* if every $T \vdash v$ occurs at most once on every branch. We start with a simple result on simple proofs:

Lemma 1 *If there is a simple proof $\mathcal{P}$ of one of the following forms:*

$$\frac{\dfrac{\vdots}{T \vdash \langle u, v\rangle}}{T \vdash u} \qquad \frac{\dfrac{\vdots}{T \vdash \langle v, u\rangle}}{T \vdash u} \qquad \frac{\dfrac{\vdots}{T \vdash [u]_v} \quad \dfrac{\vdots}{T \vdash v}}{T \vdash u}$$

Then $\langle u, v\rangle \in \mathsf{St}(T)$ (resp. $\langle v, u\rangle \in \mathsf{St}(T)$, resp. $[u]_v \in \mathsf{St}(T)$).

Proof:(sketch) Assume that the last rule is (UL). The other cases are similar. We construct inductively a branch of the proof whose every label $T \vdash w$ is such that $\langle u, v\rangle \in \mathsf{St}(w)$. Assume that the branch is constructed up to $T \vdash w$ and consider the subproof yielding $T \vdash w$. By the minimality hypothesis, either $w \neq \langle u, v\rangle$ or the last inference rule is not a pairing. Then, investigating all possible last rules, at least one of the premisses must contain $\langle u, v\rangle$ as a subterm. $\square$

2.2 Normal proofs

Definition 2 *A simple proof $\mathcal{P}$ of $T \vdash u$ is* normal *if*

- *either $u \in \mathsf{St}(T)$ and every non-leaf node of $\mathcal{P}$ is labeled $T \vdash v$ with $v \in \mathsf{St}(T)$*
- *or $\mathcal{P} = C[\mathcal{P}_1, \ldots, \mathcal{P}_n]$ where every proof $\mathcal{P}_i$ is a normal proof of some $T \vdash v_i$ with $v_i \in \mathsf{St}(T)$ and C is built using the inference rules P,E,GX,I only.*

Lemma 2 *If there is a proof of $T \vdash u$, then there is a normal proof of $T \vdash u$.*

Proof: Consider first the case where $u \in \mathsf{St}(T)$. We prove the result by induction on the size of the proof of $T \vdash u$. If the proof consists in an application of (A) only, the result is straightforward.

Let P be a proof of $T \vdash u$. We may assume w.l.o.g. that the proof is simple (otherwise simplify it and apply the induction hypothesis). Then consider all possible cases for the last inference:

If the last rule is (P) (or (E), which is handled in a similar way), then $u = \langle u_1, u_2 \rangle$. Observe that $u_1, u_2 \in \mathsf{St}(u) \subseteq \mathsf{St}(T)$. By induction hypothesis, there is a normal proof of $T \vdash u_1$ that involves only terms in $\mathsf{St}(T)$, and a normal proof of $T \vdash u_2$ that involves only terms in $\mathsf{St}(T)$, hence the result.

If the last rule is (I), then $u = I(v)\downarrow$ and there is a proof of $T \vdash v$, which is strictly smaller than $\mathcal{P}$. Moreover, $\mathsf{St}(T)$ is closed under inverses, hence $v \in \mathsf{St}(T)$. It follows, by induction hypothesis, that there is a normal proof $\mathcal{P}_1$ of $T \vdash v$. Completing $\mathcal{P}_1$ by (I) rule, we get a normal proof of $T \vdash u$.

If the last rule is (UL) (or (UR) or (D), which are handled in a similar way), assume that $\mathcal{P}$ is minimal (otherwise, the induction hypothesis applies for the shorter proof of $T \vdash u$). By lemma 1,

$$\mathcal{P} = \quad \cfrac{\cfrac{\vdots}{T \vdash \langle u, v\rangle}}{T \vdash u}$$

and $\langle u, v\rangle \in \mathsf{St}(T)$ (resp. $\langle v, u\rangle \in \mathsf{St}(T)$, resp. $[u]_v \in \mathsf{St}(T)$). Now, we apply the induction hypothesis and obtain that there is a proof of $T \vdash \langle u, v\rangle$ (resp. $T \vdash \langle v, u\rangle$, resp. $T \vdash [u]_v$) that involves only terms in $\mathsf{St}(T)$.

If the last rule is (GX) and we assume u is in normal form, consider the maximal context C such that $C[\mathcal{P}_1, \ldots, \mathcal{P}_n] = \mathcal{P}$ and every node of C is obtained by a (GX) rule or an (I) rule (such a context does exist since at least the root node is obtained in this way). We transform the proof, gathering together (GX) inferences and commuting (I) and (GX) in such a way that (I)'s occur before (GX).

The roots of $\mathcal{P}'_1, \ldots, \mathcal{P}'_n$ are respectively labeled with $T \vdash u_1 \oplus v_1, \ldots, T \vdash u_n \oplus v_n$ and $u = u_1 \oplus \ldots \oplus u_n$, $(v_1 \oplus \ldots \oplus v_n)\downarrow = 0$. We only consider here normal forms (each $u_i \oplus v_i$ is assumed to be in normal form), with the exception that some of the u_i, v_i can be 0.

For each $u_i \oplus v_i$, there are four possible cases:

Case A $v_i \neq 0$ and $(u_i \oplus v_i)\downarrow$ is headed with $\oplus$

Case B $v_i = 0$ and u_i is headed with $\oplus$

Case C $v_i = 0$ and u_i is not headed with $\oplus$

Case D $u_i = 0$ and v_i is not headed with $\oplus$

Observe that $(u_i \oplus v_i)\downarrow \in \mathsf{St}(T)$ for all terms $u_i \oplus v_i$ that fall into cases A,B or C: in case C, this follows from $(u_i \oplus v_i)\downarrow = u_i \in \mathsf{St}(u)$ and, in case A,B, $(u_i \oplus v_i)\downarrow \in \mathsf{St}(T)$. Indeed, by maximality of C, the last inference rule of $\mathcal{P}_i$ cannot be (GX) or (I). Since $u_i \oplus v_i$ is headed with $\oplus$, it cannot be (P) or (E). If it is (A), then the result is straightforward. In all other cases, we can apply lemma 1. Then either $\mathcal{P}_i = \mathcal{P}'_i$ and $(u_i \oplus v_i)\downarrow \in \mathsf{St}(T)$ or else $\mathcal{P}'_i$ is obtained from $\mathcal{P}_i$ by applying an (I) rule and $(I(u_i) \oplus I(v_i))\downarrow \in \mathsf{St}(T)$, which implies, by definition of $\mathsf{St}(T)$, that $I((I(u_i) \oplus I(v_i))\downarrow)\downarrow = (u_i \oplus v_i)\downarrow \in \mathsf{St}(T)$.

By induction hypothesis, this means that there exist normal proofs $\mathcal{P}''_i$ for $T \vdash (u_i \oplus v_i)\downarrow$ that belong to cases A, B, and C. (Note that, although $\mathcal{P}'$ may be larger than $\mathcal{P}$, each $\mathcal{P}'_i$ is strictly smaller than $\mathcal{P}$ since $\mathcal{P}$ contains at least one (GX) node in addition).

Now we will argue that a normal proof for $T \vdash u = u_1 \oplus \cdots \oplus u_n$ can be constructed using *only* terms $u_i \oplus v_i$ that fall into cases A, B, or C, and that all subproofs of $T \vdash v_i$ for $v_i \notin \mathsf{St}(T)$ (case D) can be simply removed from the proof of $T \vdash u$.

Suppose there is a total of m terms $u_i \oplus v_i$ that fall into cases A, B, or C ($m \leq n$). Rearrange the proofs so that $v_1, \ldots, v_k \in \mathsf{St}(T)$ (case A) and $v_{k+1}, \ldots, v_m = 0$ (case B or C).

For every term v in normal form, we define $\mathsf{atomic}(v)$ as follows:

- If v is not headed with $\oplus$, then $\mathsf{atomic}(v) = \{v\}$
- If $v = v_1 \oplus \ldots \oplus v_k$ and $v_1, \ldots, v_k$ are not headed with $\oplus$, then $\mathsf{atomic}(v)$ is the multiset $\{v_1, \ldots, v_k\}$.

Consider an index $i > m$. Now, since $v_1 \oplus \ldots \oplus v_n = 0$, there is a v_j such that $I(v_i) \in \mathsf{atomic}(v_j)$. However, $(u_j \oplus v_j)\downarrow$ cannot be headed with $\oplus$. For, if it were headed with $\oplus$, $(u_j \oplus v_j)\downarrow \in \mathsf{St}(T)$, as seen above, which implies $v_j \in \mathsf{SS}(T)$, hence $I(v_i) \in \mathsf{St}(T)$, which contradicts $v_i \notin \mathsf{St}(T)$. It follows that $v_j = I(v_i)$ and $u_j = 0$. Thus for every term v_i such that $i > m$, for every $j \leq m$, $I(v_i) \notin \mathsf{atomic}(v_j)$. It follows that $v_1 \oplus \ldots \oplus v_k = 0$.

$$\text{(GX)}\quad \dfrac{\dfrac{P_1}{T \vdash u_1 \oplus v_1} \quad \cdots \quad \dfrac{P_k}{T \vdash u_k \oplus v_k} \quad \dfrac{P_{k+1}}{T \vdash u_{k+1}} \quad \cdots \quad \dfrac{P_m}{T \vdash u_m}}{T \vdash \underbrace{u_1 \oplus \ldots \oplus u_m}_{=u}}$$

Figure 4. Rearranging the sums

Therefore, the normal proof for $u_1 \oplus \cdots \oplus u_k = u_1 \oplus v_1 \oplus u_2 \oplus v_2 \oplus \cdots \oplus u_n \oplus v_n$ can be constructed simply by adding $u_1 \oplus v_1, \ldots, u_k \oplus v_k$ in any order. Since $u_1 \oplus v_1, \ldots, u_k \oplus v_k \in \mathsf{St}(T)$, and $u = u_1 \oplus \cdots \oplus u_n$, this proof satisfies the definition of the normal proof.

Finally, construct the normal proof for u by adding normal proofs for $u_{k+1}, \ldots, u_m$ as shown in figure 4.

Now let us go back to the general case: consider a minimal size proof $\mathcal{P}$ of $T \vdash u$, not assuming that $u \in \mathsf{St}(T)$. The proof can be written $C[\mathcal{P}_1, \ldots, \mathcal{P}_n]$ where $\mathcal{P}_1, \ldots, \mathcal{P}_n$ are maximal subtrees of $\mathcal{P}$ whose roots are labeled with $T \vdash v_i$ respectively and $v_i \in \mathsf{St}(T)$ for every i.

Let us prove the result by induction on the size of C:

- If C is empty, then $u \in \mathsf{St}(T)$.
- If the last inference rule of C is (UL), (UR) or (D), then, by lemma 1, $u \in \mathsf{St}(T)$.
- In all other cases, we consider the sons of $\mathcal{P}$ and apply the induction hypothesis. ◻

2.3 Ground reachability is in NP

Theorem 3 *Given a finite set of (ground) terms T and a ground term u, the derivability of $T \vdash u$ is in NP, both in the* `xor` *case and in the case of Abelian groups.*

Proof: (sketch) Define *immediate derivability* as follows: $T \vdash u$ is immediately derivable if $\exists\, u_1, \ldots, u_n \in T$ such that $T \vdash u$ is obtained from $T \vdash u_1, \ldots, T \vdash u_n$ by a single application of an inference rule from (E,P,I,GX).
To prove membership in NP, we proceed as follows:

1. Guess a subset S of $\mathsf{St}(T \cup \{u\})$ containing u. Guess an ordering $s_1 > \ldots > s_n$ on $S \cap \mathsf{St}(T)$ and an ordering $u_1 > \ldots > u_m$ on the remaining part of S. Informally, we guess the order in which the subterms are derived.
2. For every $i = 1, ..., n$, check that $T \cup \{s_1, \ldots, s_{i-1}\} \vdash s_i$ is immediately derivable.
3. For every $i = 1, ..., m$, check that $T \cup \{s_1, \ldots, s_n\} \cup \{u_1, \ldots, u_{i-1}\} \vdash u$ is immediately derivable using rules from (E,P,I,GX) only.

This algorithm is in NP since there are at most $O(|\mathsf{St}(T \cup \{u\})|)$ steps (*i.e.*, polynomially many) and each step can be completed in non-deterministic polynomial time.

It is straightforward to see that if the algorithm succeeds, then $T \vdash u$ is derivable. For the converse, we rely on lemma 2: if $T \vdash u$ is derivable, then there is a normal proof of $T \vdash u$, from which we can derive an ordering on $\mathsf{St}(T, u)$. ◻

Let us point out that, as shown in [5], the applicability of GX can actually be checked in polynomial time in the `xor` case. Then it follows from their result that the derivability of $T \vdash u$ is actually in PTIME, at least in the `xor` case, using a marking algorithms of terms in $\mathsf{St}(T, u)$.

3 Constraint solving in the presence of `xor`

For a fixed number of sessions, the reachability problem reduces (in NP) to the reachability problem for a single session by guessing an interleaving of the processes representing each protocol role and then guessing at which step of the combined process the property is violated [13]. Then, as explained in section 1.2, the protocol insecurity problem reduces to constraint solving.

Our (reachability) constraints are finite conjunctions of expressions $T \Vdash t$ where T is a finite set of terms (with variables) and t is a term (with variable) and such that, for any two atomic constraints $T \Vdash t, T' \Vdash t'$ in C, either $T \subseteq T'$ or $T' \subseteq T$.

A *solution* of a constraint C is a substitution σ such that, for every $T \Vdash_i u$ in C, $T\sigma, u\sigma$ are ground and $T\sigma \vdash u\sigma$ is derivable (using the rules of figure 2), *i.e.*, the attacker can construct term u from the terms T available to it after all variables have been instantiated as specified by σ.

From now on, we only consider the `xor` theory, *i.e.*, the equations of figure 3 and the identity $I(x) = x$. We will discuss the additional difficulties in the Abelian group case.

To motivate our constructions, let us show some of the difficulties on examples.

Example 1
Consider the constraint $x \Vdash a$ where x is a variable. Then the set of solutions contains $a, \langle a, b\rangle, \langle [a]_b, b\rangle, \ldots$. It is infinite and cannot be represented as a set of instances of finitely many terms.

Example 2
$\{a\} \Vdash x \oplus y \oplus a, \{a, u(x, y), \ldots\} \Vdash v$. The constraint can be simplified to the equivalent one: $\{a\} \Vdash x', \{a\} \Vdash y', \{u(x' \oplus z, y' \oplus z), \ldots\} \Vdash v$, where x', y', z are new variables. In this example, though all variables initially occur on both sides, variable z occurs only on the left. This situation cannot occur in the restricted class of protocols considered in [5]. It cannot occur either in problems derived from protocol analysis.

Example 3
Let $T = \{b_2 \oplus b_3, a_1 \oplus a_3, x \oplus a_2 \oplus a_3\}$ and $u = b_1 \oplus b_3$. A solution of $T \Vdash u$ is, for instance, $x \mapsto \langle b_1 \oplus b_2, a_1 \oplus a_3\rangle \oplus a_1 \oplus a_2$.

Our first goal is to get rid of possible instantiations that collapse some of the subexpressions by application of the rules of figure 3.

Definition 3 *A substitution is* normalized, *if, for every variable x, $x\sigma$ is irreducible by the rules of figure 3.*
A normalized solution σ of C is collapse free *if:*

- *For any two distinct terms u, v in $\mathsf{St}(C)$, $u\sigma \downarrow \neq v\sigma \downarrow$*
- *For any distinct terms $u_1, \ldots, u_n \in \mathsf{St}(C)$, $(u_1 \oplus \ldots \oplus u_n)\sigma \downarrow = u_1\sigma \oplus \ldots \oplus u_n\sigma$.*

Example 4
Consider the constraint $x \oplus a \Vdash y \oplus b$. $\{x \mapsto a \oplus b; y \mapsto b \oplus b\}$ is a non-normalized solution, $\{x \mapsto a \oplus b; y \mapsto 0\}$, $\{x \mapsto c \oplus b; y \mapsto a \oplus c\}$, $\{x \mapsto b; y \mapsto a\}$ are non-collapse-free solutions. Actually, there is no collapse-free solution to this constraint since x must contain a b, which implies that $(x \oplus b)\sigma \downarrow \neq x\sigma \oplus b$

3.1 First reduction step

The goal of this section is to transform any term t into a finite set of terms $S(t)$ such that the set of instances of t (by a normalized substitution) is the set $\{u\theta \mid u \in S(t), \theta \text{ is a collapse-free substitution}\}$.

The first idea is to guess all possible equalities between subterms and solve them relying on unification procedures. However, this is not sufficient.

Example 5
Consider the term $t = f(x \oplus y, x)$ (and the theory of `xor`). Subterms are $t, x \oplus y, x, y$. Guessing equalities between these terms, we get several unsolvable systems; only three identifications yield solvable equations: the empty identification, $x = y$ and finally $x \oplus y = x$, which yields $y = 0$. With such identifications, we get the three terms $f(x \oplus y, x), f(0, x), f(x, x)$. Consider now $\sigma = \{x \mapsto a \oplus b\ y \mapsto a\}$. $t\sigma$ contains a redex and $t\sigma \downarrow = f(b, a \oplus b)$ is not an instance of any of the above three terms.

We use a procedure which is similar to the combination of unification procedures of [18, 3], but we insert step 3:

1. Introduce a new variable x_t for each non variable $t \in \mathsf{St}(C)$. We record this variable introduction in a set of equalities E containing $x_{f(t_1,\ldots,t_n)} = f(x_{t_1}, \ldots, x_{t_n})$, where, by convention, when t_i is a variable, x_{t_i} is the variable t_i itself.

2. Guess an equivalence relation $=_S$ on variables $x_t, t \in \mathsf{St}(C) \setminus Var(C)$. For each equivalence class, let $\widetilde{x}$ be a representative of x, G_0 be the set of representatives of x_t where t not headed with $\oplus$, and $G_\oplus$ be the set of representatives for x_t where t is headed with $\oplus$. By convention, we let $\widetilde{x} = x$ for $x \in Var(C)$.

3. Guess for each variable $x \in Var(C)$ a decomposition
$$x = \sum_{y \in G_0} \epsilon_{x,y} y \oplus \sum_{x \in S \subseteq Var(C)} \epsilon_{x,S} y_S$$
where $\epsilon_i \in \{0, 1\}$ and y_S are new variables. The equalities are recorded in a substitution σ_0.

4. Compute, for every variable $x \in \{\widetilde{x_t}, t \in \mathsf{St}(C)\} \cup Var(C)$, the normal form $\overline{x}$ defined as follows:
   - $\overline{x_{f(t_1,\ldots,t_n)}} \stackrel{\text{def}}{=} f(\widetilde{x_{t_1}}, \ldots, \widetilde{x_{t_n}})$
   - $\overline{x_{t_1 \oplus \ldots \oplus t_n}} \stackrel{\text{def}}{=} (\widetilde{x_{t_1}} \oplus \ldots \oplus \widetilde{x_{t_n}})\sigma_0 \downarrow$.
   - $\overline{x} \stackrel{\text{def}}{=} x\sigma_0$ if $x \in Var(C)$.

   And for other variables, $\overline{x_t} \stackrel{\text{def}}{=} \overline{\widetilde{x}_t}$

5. Compute the occurrence relation on variables as the smallest transitive relation s.t. $x > y$ if y occurs in $\overline{x}$.

6. Check satisfiability using the rules of figure 5, which express on one hand the independence of the representative choice for equivalence classes and, on the other hand, the absence of cycles.

Example 6
Assume that
$$\mathsf{St}(C) = \{z, y, z \oplus y, f(z \oplus y)\}.$$

$$\begin{array}{rcl}
x = f(x_1,\ldots,x_n) \in E & \Rightarrow & \overline{x} = f(\widetilde{x_1},\ldots,\widetilde{x_n}) \\
x = x_1 \oplus \ldots \oplus x_n \in E & \Rightarrow & \overline{x} = (\widetilde{x_1} \oplus \ldots \oplus \widetilde{x_n})\sigma_0 \downarrow \\
x > x & \Rightarrow & \perp
\end{array}$$

Figure 5. Checking the consistency of guesses

Step 1 introduces two variables $x_{z\oplus y}$ and $x_{f(z\oplus y)}$ and we have the equalities $x_{z\oplus y} = z \oplus y$ and $x_{f(z\oplus y)} = f(x_{z\oplus y})$.

At step 2, assume we don't guess any equality.

At step 3 we guess the decompositions:

$$\begin{array}{rcl}
z & = & \epsilon_{z,f(z\oplus y)} x_{f(z\oplus y)} \oplus \epsilon_{10}x_{10} \oplus \epsilon_{11}x_{11} \\
y & = & \epsilon_{y,f(z\oplus y)} x_{f(z\oplus y)} \oplus \epsilon_{01}x_{01} \oplus \epsilon_{11}x_{11}.
\end{array}$$

At step 4, we compute:

$$\begin{array}{rcl}
\overline{x_{z\oplus y}} & = & (\epsilon_{z,f(z\oplus y)} x_{f(z\oplus y)} \oplus \epsilon_{y,f(z\oplus y)} x_{f(z\oplus y)} \\
 & & \oplus \epsilon_{10}x_{10} \oplus \epsilon_{01}x_{01}) \downarrow \\
\overline{x_{f(z\oplus y)}} & = & f(x_{z\oplus y}) \\
\overline{z} & = & \epsilon_{z,f(z\oplus y)} x_{f(z\oplus y)} \oplus \epsilon_{10}x_{10} \oplus \epsilon_{11}x_{11} \\
\overline{y} & = & \epsilon_{y,f(z\oplus y)} x_{f(z\oplus y)} \oplus \epsilon_{01}x_{01} \oplus \epsilon_{11}x_{11}.
\end{array}$$

At step 5, we compute the occurrence relation. For instance, if we chose $\epsilon_{z,f(z\oplus y)} = 1$ and $\epsilon_{y,f(z\oplus y)} = 0$, then we get $x_{z\oplus y} > x_{f(z\oplus y)} > x_{z\oplus y}$, which will yield unsatisfiability at the next step. Similarly, choosing $\epsilon_{z,f(z\oplus y)} = 0$ and $\epsilon_{y,f(z\oplus y)} = 1$ yields unsatisfiability.

Assume now that we chose $\epsilon_{z,f(z\oplus y)} = 1$ and $\epsilon_{y,f(z\oplus y)} = 1$, then $\overline{x_{z\oplus y}} = \epsilon_{10}x_{10} \oplus \epsilon_{01}x_{01}$ and the system passes the satisfiability test.

For instance, if $\epsilon_{11} = 0$, $\epsilon_{01} = \epsilon_{10} = 1$, we get

$$\begin{array}{rcl}
\overline{z} & = & x_{f(y\oplus z)} \oplus x_{10} \\
\overline{y} & = & x_{f(y\oplus z)} \oplus x_{01} \\
\overline{x_{z\oplus y}} & = & x_{10} \oplus x_{01} \\
\overline{x_{f(z\oplus y)}} & = & f(x_{z\oplus y})
\end{array}$$

Lemma 4 *The set of equalities $\overline{x} = t$ defines a substitution θ on variables such that $x_t =_S x_u$ implies $t\theta \downarrow =_{AC} u\theta \downarrow$.*

It follows that the above procedure can be seen as a non-deterministic computation of a substitution θ.

Lemma 5 *If σ is a normalised solution of C, then there is a substitution θ, which is an output of the above procedure, there is a collapse-free solution σ' of $C\theta$ such that, for every $x \in Var(C)$, $x\sigma = x\theta\sigma'$.*

This lemma allows us to focus our attention on collapse-free solutions.

Extending the lemma to the case of Abelian groups is non-trivial. Nevertheless, we believe that there is a similar lemma for the Abelian groups, proceeding along the lines of the above example.

3.2 Second reduction step

Let $T_1 \subseteq \ldots \subseteq T_n$ be the left members of sequents in C. Guess for each i the set S_i of terms in $\mathsf{St}(T_i)$ that can be deduced from T_i. Guess a linear ordering $>_i$ on S_i. This guess has to be consistent: $S_i \subseteq S_{i+1}$ and $>_{i+1}$ extends $>_i$. Then replace each occurrence of T_i with $T_i \cup S_i$ and add, for every i and every $u \in S_i$ the constraint $T_i \cup \{v \in S_i \mid v <_i u\} \Vdash u$.

Then, close every left member by $\oplus$ (if $u \oplus v, v \oplus w \in T$, then $u \oplus w \in T$) and remove sequents $T \Vdash u$ such that $T \vdash u$ is derivable (with the rules of figure 2).

We write $C \rightsquigarrow_2 C'$ if C' can be obtained from C by the above-described transformation.

Lemma 6

- *Every σ which is a collapse-free solution of C, is also a collapse free solution of some C' such that $C \rightsquigarrow_2 C'$.*
- *for every collapse free solution σ of C, there is some C' such that $C \rightsquigarrow_2 C'$ and, for every $T \Vdash u \in C'$, for every $v \in \mathsf{St}(T)$ such that $T\sigma \vdash v\sigma$ is derivable, then either $v \in T$ or $T \vdash v$ is not derivable.*
- *If $C \rightsquigarrow_2 C'$ and $T \Vdash u \in C'$, then T is closed by $\oplus$.*
- *If $C \rightsquigarrow_2 C'$, then $\mathsf{St}(C') \subseteq \mathsf{St}(C)$ and every solution of C' is a solution of C*

Example 7
The following example will be reused later on. Assume we have the following constraint after step 1:

$$\begin{array}{rcl}
x_1, x_2 \oplus x_3 & \Vdash & b \oplus [x_4]_a \\
x_1, x_2 \oplus x_3, x_4 & \Vdash & \langle x_2, b \rangle
\end{array}$$

and let σ_0 be the following collapse-free solution of the constraint:

$$\{x_1 \mapsto \langle c, d \rangle; x_2 \mapsto c \oplus b; x_3 \mapsto d \oplus [\langle b, a \rangle]_a; x_4 \mapsto \langle b, a \rangle\}$$

after step 2 the second constraint is replaced with (among others)

$$\begin{array}{rcl}
x_1, x_2 \oplus x_3, x_4 & \Vdash & x_2 \\
x_1, x_2 \oplus x_3, x_4 & \Vdash & x_3 \\
x_1, x_2, x_3, x_4 & \Vdash & \langle x_2, b \rangle
\end{array}$$

whose σ_0 is a solution.

(P)	$T \Vdash \langle u, v\rangle$	$\rightsquigarrow$	$T \Vdash u, T \Vdash v$	
(E)	$T \Vdash [u]_v$	$\rightsquigarrow$	$T \Vdash u, T \Vdash v$	
(A)	$T \Vdash u$	$\rightsquigarrow$	$\top$	If $u \in T$
(S)	$T \Vdash u \oplus v$	$\rightsquigarrow$	$T \Vdash u, T \Vdash v$	
(X)	$T, v \oplus w \Vdash u \oplus v$	$\rightsquigarrow$	$T, v \oplus w \Vdash u \oplus w$	

Figure 6. Constraint transformation rules

3.3 Solved forms and completeness

Definition 4 *C is in* solved form *if there is an ordering $\geq_{occ}$ on $\mathsf{St}(C)$ such that:*

1. *for every sequent $T \Vdash u \in C$, u is either a variable or a sum $u_1 \oplus \ldots \oplus u_n$ and in the latter case:*
   - *(a) for every $i = 1, ...n$, there is a variable $x_{u_i} \in Var(T)$ such that $x_{u_i} >_{occ} u_i$, and*
   - *(b) either $x_{u_i} \in T$ or there is a term $x_{u_i} \oplus v_{u_i} \in T$ such that $x_{u_i} >_{occ} y$ for $y \in Var(v_{u_i})$*
2. *if $T \Vdash x \oplus v \in C$ (or $T \Vdash x \in C$) and $T, T' \Vdash u_1 \oplus \ldots \oplus u_n \in C$, then for every i, $x_{u_i} \neq x$*
3. *If $x \in Var(u)$, then $u \geq_{occ} x$.*

Lemma 7 *If σ is a collapse free solution of C (obtained by the previous transformation steps), then there is a C' such that $C \rightsquigarrow C'$ by the rules of figure 6 and C' is a solved form.*

Example 8
Consider again example 7. The constraint now becomes

$$\begin{array}{rcl} x_1, x_2 \oplus x_3 & \Vdash & b \oplus [x_4]_a \\ x_1, x_2 \oplus x_3, x_4 & \Vdash & x_2 \\ x_1, x_2 \oplus x_3, x_4 & \Vdash & x_3 \\ x_1, x_2, x_3, x_4 & \Vdash & b \end{array}$$

which is in solved form, with $x_b = x_2, x_{[x_4]_a} = x_3$ and the ordering $x_3 >_{occ} [x_3]_a >_{occ} x_4$. It can be easily checked that σ_0 is again a solution of the constraint.

For every solved form, it is easy to construct a (non necessarily collapse-free,but it doesn't matter) solution of the constraint, by induction on the variables ordered by $\geq_{occ}$:

Lemma 8 *Every solved form has a solution.*

3.4 The main result

Theorem 9 *The solvability of reachability constraints is decidable (in the case of* `xor`*).*

Proof: We may only consider normalized solutions. Given a constraint C, perform the first two reduction steps. They terminate and, according to lemmas 6, 5, C has a solution only if at least one of the resulting constraints C' has a collapse-free solution. And if some of the resulting constraints has a solution, then C has a solution.

Now, consider the rules of figure 6. They preserve the set $\mathsf{St}(C)$. Hence only finitely many distinct constraints can be derived using such rules. If the application of the rules does not terminate, that can only be due to a loop (with **X**). We simply avoid loops, keeping tracks of previous constraints. Then, by lemma 7, if C has a solution, then there is a constraint C' obtained by this (now terminating) set of rules such that C' is a solved form and C' has a solution. And conversely: the solutions of such C' are also solutions of C.

Then either all C' obtained in this way are not in solved form, in which case C has no solution or else there is a constraint C' in solved form, in which case C has a solution, thanks to lemma 8. □

Corollary 10 *In the presence of* `xor`*, the failure of secrecy or authentication is decidable for a fixed number of sessions.*

4 Example: Bull's authentication protocol

We illustrate the constraint solving technique of section 3 by applying it to Bull's recursive authentication protocol. The protocol was first proved correct [15], and then shown to be vulnerable [17] once properties of `xor` are taken into account. We consider the 3-party version of the protocol (see figure 7). It is assumed that $K_{a,b,c}$ are shared between S and A, B, C respectively, and h is a function symbol modeling a hash function. Among other goals, the protocol should prevent C from learning session key K_{ab}.

Secrecy of K_{ab} from C fails if the following constraint is satisfiable: $T = \{M_b, x, y, K_{ab} \oplus h(K_a, N_a), K_{ab} \oplus h(K_b, N_b), K_{bc} \oplus h(K_b, N_b), K_{bc} \oplus h(x, y)\} \Vdash K_{ab}$ (x, y represent terms under control of C). We guess that x, y are not instantiated, and the following order of subterm derivation: $h(x, y) < K_{bc} < K_{ab}$. Observe that $T \vdash h(x, y)$ in one step by applying function symbol h to x and y, $T \cup h(x, y) \vdash K_{bc}$ by (X), $T \cup h(x, y) \cup K_{bc} \vdash K_{ab}$ by two applications of (X): $((K_{ab} \oplus h(K_b, N_b)) \oplus (K_{bc} \oplus h(K_b, N_b))) \oplus K_{bc} = K_{ab}$. Therefore, after the second reduction step we obtain an empty constraint, proving that secrecy of K_{ab} is violated.

1. $A \rightarrow B : M_a = A, B, N_a, h(A, B, N_a)$
2. $B \rightarrow C : M_b = B, C, N_b, M_a, h(B, C, N_b, M_a)$
3. $C \rightarrow S : M_c = C, S, N_c, M_b, h(C, S, N_c, M_b)$
4. $S \rightarrow C : K_{ab} \oplus h(K_a, N_a), K_{ab} \oplus h(K_b, N_b),$
   $K_{bc} \oplus h(K_b, N_b), K_{bc} \oplus h(K_c, N_c)$
5. $C \rightarrow B : K_{ab} \oplus h(K_a, N_a), K_{ab} \oplus h(K_b, N_b),$
   $K_{bc} \oplus h(K_b, N_b)$
6. $B \rightarrow A : K_{ab} \oplus h(K_a, N_a)$

Figure 7. Recursive authentication protocol

5 Concluding remarks

We gave the first decision result for cryptographic protocols which does not assume the perfect cryptography. First, our work is incomplete in two respects:

(i) What is the exact complexity of solving the reachability constraint in the `xor` case?

(ii) What about an analog of theorem 9 in the case of Abelian groups?

It should be possible to answer the second question with a significant extra effort and time since we believe that there is an analog of lemma 5 for Abelian groups.

The techniques introduced in section 2 and the work of McAllester [10] raise a more general question:

> For which (relevant) equational theories is the reachability problem in PTIME ? NP? decidable?

The techniques introduced in section 3.1 also raise a general question in unification theory:

> For which equational theories can we restrict our attention to collapse-free solutions?

In other words: can we extend lemma 5 to arbitrary theories for which unification is finitary?

Finally, it would be nice to abstract out the lifting argument of section 3:

> Is there a general relationship between the complexity of reachability in the ground case and the solvability of reachability constraints?

Acknowledgments

We would like to thank the authors of [5] who pointed out a mistake in the complexity analysis provided in the submitted version of this paper, as well as several valuable comments from the referees. This work has been partially supported by the RNTL project EVA and by ACI Vernam.

References

[1] R. Amadio and W. Charatonik. On name generation and set-based analysis in the Dolev-Yao model. In *Proc. CONCUR '02*, volume 2421 of *LNCS*, pages 499–514, 2002.

[2] R. Amadio and D. Lugiez. On the reachability problem in cryptographic protocols. In *Proc. CONCUR '00*, volume 1877 of *LNCS*, pages 380–394, 2000.

[3] F. Baader and K. Schulz. Combining constraint solving. In *Constraints in Computational Logics (CCL '99)*, volume 2002 of *LNCS*, pages 104–158, 2001.

[4] M. Boreale. Symbolic trace analysis of cryptographic protocols. In *Proc. ICALP '01*, pages 667–681, 2001.

[5] Y. Chevalier, R. Kuester, M. Rusinowitch, and M. Turuani. An np decision procedure for protocol insecurity with xor. In *These proceedings*, 2003.

[6] H. Comon, V. Cortier, and J. Mitchell. Tree automata with one memory, set constraints, and ping-pong protocols. In *Proc. ICALP '01*, pages 682–693, 2001.

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

[8] N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Proc. Workshop on formal methods in security protocols*, Trento, Italy, 1999.

[9] M. Fiore and M. Abadi. Computing symbolic models for verifying cryptographic protocols. In *Proc. 14th IEEE Computer Security Foundations Workshop*, pages 160–173, 2001.

[10] D. McAllester. Automatic recognition of tractability in inference relations. *J. ACM*, 40(2):284–303, 1993.

[11] C. Meadows and P. Narendran. A unification algorithm for the group Diffie-Hellman protocol. In *Proc. Workshop on Issues in Theory of Security*, 2002.

[12] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[13] J. Millen and V. Shmatikov. Constraint solving for bounded process cryptographic protocol analysis. In *Proc. 8th ACM Conference on Computer and Communications Security (CCS '01)*, pages 166–175, 2001.

[14] P. Narendran, Q. Guo, and D. Wolfram. Unification and matching modulo nilpotence. In *Proc. CADE-13*, volume 1104 of *LNCS*, pages 261–274, 1996.

[15] L. Paulson. Mechanized proofs for a recursive authentication protocol. In *Proc. 10th IEEE Computer Security Foundations Workshop*, pages 84–95, 1997.

[16] M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proc. 14th IEEE Computer Security Foundations Workshop*, pages 174–190, 2001.

[17] P. Ryan and S. Schneider. An attack on a recursive authentication protocol: A cautionary tale. *Information Processing Letters*, 65(1):7–10, 1998.

[18] M. Schmidt-Schauß. Unification in a combination of arbitrary disjoint equational theories. *J. Symbolic Computation*, 8(1/2):51–99, 1989.

[19] M. Steiner, G. Tsudik, and M. Waidner. Diffie-Hellman key distribution extended to group communication. In *Proc. 3rd ACM Conference on Computer and Communications Security (CCS '96)*, pages 31–37, 1996.