# Intruder Deduction for *AC*-like Equational Theories with Homomorphisms

Pascal Lafourcade[1,2], Denis Lugiez[2], and Ralf Treinen[1*]

[1] LSV, ENS de Cachan & CNRS UMR 8643 & INRIA Futurs project SECSI , 94235 Cachan, France, `http://www.lsv.ens-cachan.fr/~{lafourca,treinen}`
[2] LIF, Université Aix-Marseille 1 & CNRS UMR 6166, 13453 Marseille Cedex 13, France, `http://www.cmi.univ-mrs.fr/~lugiez`

**Abstract.** Cryptographic protocols are small programs which involve a high level of concurrency and which are difficult to analyze by hand. The most successful methods to verify such protocols rely on rewriting techniques and automated deduction in order to implement or mimic the process calculus describing the protocol execution.
We focus on the intruder deduction problem, that is the vulnerability to passive attacks, in presence of several variants of $AC$-like axioms (from $AC$ to Abelian groups, including the theory of *exclusive or*) and homomorphism which are the most frequent axioms arising in cryptographic protocols. Solutions are known for the cases of *exclusive or*, of Abelian groups, and of homomorphism alone. In this paper we address the combination of these $AC$-like theories with the law of homomorphism which leads to much more complex decision problems.
We prove decidability of the intruder deduction problem in all cases considered. Our decision procedure is in EXPTIME, except for a restricted case in which we have been able to get a PTIME decision procedure using a property of one-counter and pushdown automata.

## 1 Introduction

Cryptographic protocols are ubiquitous in distributed computing applications. They are employed for instance in internet banking, video on demand services, wireless communication, or secure UNIX services like `ssh` or `scp`. Cryptographic protocols can be described as relatively simple programs which are executed in an untrusted environment. These protocols use cryptographic primitives in order to implement symmetric (shared-key) encryption, and asymmetric (public-key) encryption and signatures.

Verifying protocols is notoriously difficult, and even very simple protocols which look completely harmless may have serious security flaws, as it was dramatically demonstrated by the bug of the Needham-Schroeder protocol found by Lowe [14] using a model-checking tool. It took 17 years since the protocol was published to find the flaw, a so-called *man in the middle attack*. An overview of

* This work was partially supported by the research programs ACI-SI Rossignol, and RNTL PROUVÉ (n° 03 V 360).

*authentication protocols* known a decade ago can be found in [5], a more recent data base of protocols and known flaws is [11]. These protocols are often implemented in small variants which differ from the originally proposed protocol, or are used in combination with other protocols. As a consequence, there is a multitude of verification problems, which raises the need for *automatic* tools.

There are different approaches to modeling cryptographic protocols and analyzing their security properties: process calculi like the *spi-calculus* [1], so-called cryptographic proofs (see, for instance, [2]), and the approach of Dolev and Yao [10] which consists in modeling an attacker by a deduction system. This deduction system specifies how the attacker can obtain new information from previous knowledge, which he has either obtained by silently eavesdropping the communication between honest protocol participants (in case of a *passive* attacker), or by eavesdropping and fraudulently emitting messages, thus provoking honest protocol participants to reply according to the protocol rules (this is the case of a so-called *active* attacker). We call *intruder deduction problem* the question whether a passive eavesdropper can obtain a certain information from knowledge that he observes on the network. The Dolev-Yao approach lends itself to automation since the question whether the intruder can obtain a certain information now reduces to the question whether this information can be deduced using a certain deduction system.

Classically, the verification of cryptographic protocols was based on the so-called *perfect cryptography assumption* which states that it is impossible to obtain any information about an encrypted message without knowing the exact key necessary to decrypt this message. This assumption allowed a separation of verification tasks into proving lower bounds for the cryptanalysis of the cryptographic primitives on the one hand, and verification of a distributed program on the other hand. Unfortunately, this perfect cryptography assumption has proven too idealistic: there are protocols which can be proven secure under the perfect cryptography assumption, but which are in reality insecure since an attacker can use properties of the cryptographic primitives in combination with the protocol rules in order to obtain knowledge of a secret. These properties are typically expressed as equational axioms (so-called algebraic properties), like for instance associativity and commutativity of certain operators. Algebraic properties may be essential for the executability of the protocol, or may just come into play because the cryptographic primitives employed by the protocol happen to satisfy these properties. A recent overview of algebraic properties of cryptographic primitives, their use to mount attacks on protocols, and existing results on verification of cryptographic protocols in presence of equational axioms can be found in [8].

A number of results have been obtained, both for the intruder deduction problem and for the preservation of secrecy under active attacks. We here only mention some results which are of particular relevance to the problems studied in this work: the intruder deduction problem in case of the equational axioms of *exclusive or* is decidable [6] in polynomial time [4], and in case of the equational axioms of Abelian groups is decidable [6][3] in polynomial time [19]. Likewise, the intruder deduction problem is decidable in polynomial time [7] in the case of the equational theory of an homomorphism. Note that the two equational theories of *exclusive or* and of homomorphism model basic properties of important cryptographic primitives:

- *Exclusive or* is a basic building block in many symmetric encryption methods (for instance DES or the more recent AES) or even used directly as an encryption method;
- Homomorphisms are ubiquitous in cryptography, by example the ElGamal encryption method has this property. Note that many protocols combine symmetric and asymmetric encryption.
- Symmetric encryption methods which often work on data blocks of fixed size are in the simplest of cases (the so-called *electronic codebook mode*) homomorphically extended to data streams of arbitrary size.

Some examples of attacks against protocols using the equational theories considered in this paper can be found in [8].

In this paper we investigate the intruder deduction problem in presence of several variants of the equational theory of associativity and commutativity (short $AC$) of a binary operator $\otimes$, plus the homomorphism property of a unary function symbol over the $AC$ operator. The variants of $AC$ which we consider are: pure $AC$, the theory of *exclusive or* (also called $ACUN$), and the theory of Abelian groups. We are furthermore interested in the combination of these $AC$-like theories with a generalization of one homomorphic function to some form of distributivity of the encryption operator over the binary operator $\otimes$. The homomorphism law is then replaced by a law stating that the encryption of the $\otimes$ of two messages is equal to the $\otimes$ of the encryptions of the two messages using the same encryption key. We do not assume that the set of encryption keys is finite. Rather, any term can be used as an encryption key. This can be seen as the extension to an infinite family of homomorphisms, one for each possible encoding key. Our results can be summarized as follows:

1. The intruder deduction problem is decidable. It is NP-complete in case of the theory $AC$ plus homomorphism, and we have an exponential-time upper bound for the equational theory $ACUN$ plus homomorphism and Abelian groups plus homomorphism.
2. The intruder deduction problem is in all three cases decidable in polynomial time if we restrict the class of problems to the so-called *binary* case, that is the case where the set of assumptions and the goal do not contain applications of $\otimes$ to more than two terms.
3. The first two sets of results carry over to the generalization which consists in replacing the homomorphic function by an encryption operation which distributes over $\otimes$.

[3] In fact, the NP-decision procedure in the case of Abelian groups given by [6] can also be improved to *deterministic* polynomial time using the techniques explained in this report.

We follow the approach of [6] and [7] which consists in a generalization of McAllester's *locality* method explained in Section 3.

Plan of the paper: We present in Section 2 the Dolev-Yao model of intruder capacities extended by a rewrite system modulo $AC$ and list the rewrite systems investigated in this paper. In Section 3 we explain the generalization of McAllester's proof technique. We apply this technique in Sections 4, 5 and 6 to obtain decidability and complexity results for the case of *exclusive or* plus homomorphism. We discuss in Section 7 how these results can be transfered to some other related rewrite systems. Finally, we conclude in Section 8.

The full version of this paper with all proofs can be found at [13]. We use standard notation from rewriting. The reader may consult [9, 3] if necessary.

## 2 A Dolev-Yao Model for Rewriting Modulo $AC$

We consider the classic model of deduction rules [10] introduced by Dolev and Yao in order to model the deductive capacities of a passive intruder. We present here an extension of this model where we assume an associative and commutative operator $\otimes$, and an equational theory $E$ which can be exploited by the intruder to mount an attack. Knowledge of the intruder is represented by terms built over a finite signature $\Sigma$ of the form

$$\Sigma = \{\langle\cdot,\cdot\rangle, \{\cdot\}_\cdot, \otimes, f\} \uplus \Sigma_0$$

where $\Sigma_0$ is a set of constant symbols. The term $\langle u, v\rangle$ represents the pair of the two terms $u$ and $v$, and $\{u\}_v$ represents the encryption of the term $u$ by the term $v$. For the sake of simplicity we here only consider symmetric encryption; the results and techniques can be easily transferred to the case of asymmetric encryption.

The equational theory $E$ is represented by a convergent rewrite system $R$ modulo $AC$, that is $R$ is terminating and confluent modulo associativity and commutativity of $\otimes$, and for all terms $t, s \in T(\Sigma)$ we have that $t =_E s$ iff $t{\downarrow_{R/AC}} =_{AC} s{\downarrow_{R/AC}}$.

The deduction system describing the deductive capacities of an intruder is given in Figure 1. This deduction system is composed of the following rules: (A) the intruder may use any term which is in his initial knowledge, (P) the intruder can build a pair of two messages, (UL, UR) he can extract each member of a pair, (C) he can encrypt a message $u$ with a key $v$, (D) if he knows a key $v$ he can decrypt a message encrypted by the same key, (F) he can construct a new term using the function symbol $f$. Since we distinguish a special binary operator $\otimes$ we here furthermore add a family of rules (GX) which allows the intruder to build a new term from an arbitrary number of already known terms by using the (associative) $\otimes$ operator. The need for such a variadic rule (instead of just a binary rule) will become apparent in Section 3.

In fact, this deductive system is equivalent in deductive power to a variant of the system in which terms are not automatically normalized, but in which arbitrary equational proofs are allowed at any moment of the deduction. The

$$
\text{(A)}\quad \frac{u \in T}{T \vdash u{\downarrow_{R/AC}}} \qquad\qquad \text{(UL)}\quad \frac{T \vdash r}{T \vdash u{\downarrow_{R/AC}}} \quad if \langle u, v\rangle = r{\downarrow_{R/AC}}
$$

$$
\text{(P)}\quad \frac{T \vdash u \qquad T \vdash v}{T \vdash \langle u, v\rangle{\downarrow_{R/AC}}} \qquad\qquad \text{(UR)}\quad \frac{T \vdash r}{T \vdash v{\downarrow_{R/AC}}} \quad if \langle u, v\rangle = r{\downarrow_{R/AC}}
$$

$$
\text{(C)}\quad \frac{T \vdash u \qquad T \vdash v}{T \vdash \{u\}_v{\downarrow_{R/AC}}} \qquad\qquad \text{(D)}\quad \frac{T \vdash r \qquad T \vdash v}{T \vdash u{\downarrow_{R/AC}}} \quad if \{u\}_v = r{\downarrow_{R/AC}}
$$

$$
\text{(F)}\quad \frac{T \vdash u}{T \vdash f(u){\downarrow_{R/AC}}} \qquad\qquad \text{(GX)}\quad \frac{T \vdash u_1 \quad \cdots \quad T \vdash u_n}{T \vdash (u_1 \otimes \ldots \otimes u_n){\downarrow_{R/AC}}}
$$

**Fig. 1.** A Dolev-Yao proof system working on normal forms by a rewrite system $R$ modulo $AC$

| (a) $ACh$ | (b) $ACUNh$ | (c) $AGh$ |
| --- | --- | --- |
| $f(x \otimes y) \to f(x) \otimes f(y)$ | $0 \otimes x \to x$<br>$x \otimes x \to 0$<br>$f(0) \to 0$<br>$f(x \otimes y) \to f(x) \otimes f(y)$ | $0 \otimes x \to x$<br>$x \otimes x \to 0$<br>$I(0) \to 0$<br>$I(x \otimes y) \to I(x) \otimes I(y)$<br>$I(I(x)) \to x$<br>$f(I(x)) \to I(f(x))$<br>$f(0) \to 0$<br>$f(x \otimes y) \to f(x) \otimes f(y)$ |

**Fig. 2.** The three rewrite systems modulo $AC$

equivalence of the two proof systems has been shown in [7] without $AC$ axioms; in [13] this has been extended to the case of a rewrite system modulo $AC$.

In the rest of the paper, we will investigate the Dolev-Yao deduction system modulo the rewrite systems presented in Figure 2, which correspond respectively to $AC$ plus homomorphism of $f$ over $\otimes$, the theory of *exclusive or* plus homomorphism of $f$ over $\otimes$, and the theory of Abelian groups plus homomorphism of $f$ over $\otimes$. We will omit the index $R/AC$ and write $\to$ instead of $\to_{R/AC}$.

## 3 Locality and Complexity of Deduction Problems

Our starting point is the locality technique introduced by David McAllester [15]. He considers deduction systems which are represented by finite sets of Horn clauses. He shows that there exists a polynomial-time algorithm to decide the deducibility of a term $w$ from a finite set of terms $T_0$ if the deduction system has the so-called locality property. A deduction system has the *locality property* if any proof of $T_0 \vdash w$ can be transformed into a local proof where a *local proof* is a proof where all the nodes are syntactic subterms of $T_0$ and $w$.

The idea of his proof is as follows: Checking existence of a proof amounts to checking existence of a local proof. Let us call for the moment a *relevant instance* of a deduction rule an instance of a rule where all terms are syntactic subterms of $T_0$ or $w$. Only these relevant instances are needed to construct a local proof.

We say that $w$ is *one-step deducible* from some set $T$, if we can obtain $w$ from $T$ with only one application of a rule of the proof system. To check the existence of a local proof of $T_0 \vdash w$ it is now sufficient to saturate $T_0$ by the one-step deduction relation, where in addition it is sufficient to just consider the relevant instances of the deduction rules.

This approach suffers from two main restrictions:

- The deduction system must be finite.
- The notion of locality is restricted to syntactic subterms.

These restrictions raise a serious problem when we want to work modulo $AC$. If we used only a binary rule (GX) we would have to consider all possible subterms modulo $AC$. Unfortunately, there is in general an exponential number of subterms modulo $AC$ of a given term. The solution proposed in [6], and which we also adopt here, is to use the rule (GX) with an arbitrary number of hypotheses. In this way, we can avoid the exponential number of subterms. However, we are now stuck with an infinite number of rules. Fortunately, we can still obtain an polynomial algorithm by implementing in a clever way the test whether a term $w$ is one-step deducible from a set $T$.

**Definition 1.** *Let $S$ be a function which maps a set of terms to a set of terms. A proof $P$ of $T \vdash w$ is* $S$-local *if all nodes are labeled by some $T \vdash v$, with $v \in S(T \cup \{w\})$. A proof system is* $S$-local *if whenever there is a proof of $T \vdash w$ then there also is some $S$-local proof of $T \vdash w$.*

**Theorem 1.** *Let $S$ be a function mapping a set of terms to a set of terms, and $P$ a proof system. If*

- *the set $S(T)$ can be constructed in time $\mathcal{K}_1$,*
- *$P$ is $S$-local,*
- *one-step deducibility in $P$ is decidable in time $\mathcal{K}_2$,*

*then provability in the proof system $P$ is decidable in time* $\max(\mathcal{K}_1, \mathcal{K}_2)$.

This theorem generalizes McAllester's result because in his case the size of the set of syntactic subterms of the set $T$ is polynomial in the size of $T$, and since one-step deducibility is decidable in polynomial time for a finite proof system. Hence, in McAllester's case, it remained only the $S$-locality to show.

## 4 Proof Transformations

The following definitions and transformations can be applied to the cases $ACh$ and $ACUNh$. The case of $AGh$ requires an extension briefly discussed in Subsection 7.2.

**Definition 2.** *The* size of a proof $P$ *is the number of nodes in* $P$*, denoted by* $|P|$*. A proof* $P$ *of* $T \vdash u$ *is* minimal *if there is no proof* $P'$ *of* $T \vdash u$ *such that* $|P'| < |P|$*.*

**Definition 3.** *Let* $P$ *be a proof of* $T \vdash w$*,* $P$ *is a*

- simple proof *if each node* $T \vdash v$ *occurs at most once on each branch.*
- flat proof *if there is no (GX) rule immediately above another (GX) rule,*
- $\otimes$-lazy proof *if* $P$ *is flat and there is no (GX) rule immediately above an (F) rule in* $P$*,*
- $\otimes$-eager proof *if* $P$ *is flat and if there is at most one (F) rule immediately above a (GX) rule in* $P$*.*

Since two successive (GX) rules can be merged into a single (GX) rule a minimal proof is a flat proof. Obviously any minimal proof is simple. Intuitively, in a $\otimes$-lazy proof the (GX) rule is applied as late as possible, and in a $\otimes$-eager proof the (GX) rule is applied as early as possible.

**Lemma 1.** *If there is a proof of* $T \vdash w$ *then there is also a* $\otimes$*-lazy proof and a* $\otimes$*-eager proof of* $T \vdash w$*.*

*Proof.* Successive (GX) rules can obviously be merged. We can obtain a $\otimes$-lazy proof by applying the following proof transformation rule:

$$\text{(F)}\ \dfrac{\text{(GX)}\ \dfrac{T \vdash x_1 \ldots T \vdash x_n}{T \vdash x_1 \otimes \ldots \otimes x_n}}{T \vdash f(x_1) \otimes \ldots \otimes f(x_n)} \quad \Longrightarrow \quad \text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{T \vdash x_1}{T \vdash f(x_1)} \cdots \text{(F)}\ \dfrac{T \vdash x_n}{T \vdash f(x_n)}}{T \vdash f(x_1) \otimes \ldots \otimes f(x_n)}$$

We obtain a $\otimes$-eager proof by applying the following proof transformation, where the rules $(G_i)$ are all different from $(F)$:

$$\text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{T \vdash x_1}{T \vdash f(x_1)} \cdots \text{(F)}\ \dfrac{T \vdash x_n}{T \vdash f(x_n)} \quad (G_1)\ \dfrac{T \vdash y_1}{T \vdash z_1} \cdots (G_m)\ \dfrac{T \vdash y_m}{T \vdash z_m}}{T \vdash f(x_1) \otimes \ldots \otimes f(x_n) \otimes z_1 \otimes \ldots \otimes z_m}$$

$$\Downarrow$$

$$\text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{\text{(GX)}\ \dfrac{T \vdash x_1 \ldots T \vdash x_n}{T \vdash x_1 \otimes \ldots \otimes x_n}}{T \vdash f(x_1) \otimes \ldots \otimes f(x_n)} \quad (G_1)\ \dfrac{T \vdash y_1}{T \vdash z_1} \cdots (G_m)\ \dfrac{T \vdash y_m}{T \vdash z_m}}{T \vdash f(x_1) \otimes \ldots \otimes f(x_n) \otimes z_1 \otimes \ldots \otimes z_m}$$

## 5 Locality for the Rewrite System *ACUNh*

**Definition 4.** *Let $u$ be a term in normal form, $u$ is* headed with $\otimes$ *if $u$ is of the form $u_1 \otimes \ldots \otimes u_n$ with $n > 1$. Otherwise $u$ is not headed with $\otimes$.*

*We define the function* atoms(u) *as following :*

- *If $u = u_1 \otimes \ldots \otimes u_n$, where each of the $u_i$ is not headed with $\otimes$, then $atoms(u) = \{u_1, \ldots, u_n\}$. The terms $u_i$ are called the* atoms *of $u$.*
- *If $u$ is not headed with $\otimes$, then $atoms(u) = \{u\}$.*

The definition of atoms($T$) generalizes in a natural way to sets of terms $T$ in normal form by $\text{atoms}(T) := \bigcup_{t \in T} \text{atoms}(t)$.

**Definition 5.** *We define for any $T \subseteq T(\Sigma)$ the set $S_T(T)$ as the smallest set which contains $T$, is closed under syntactic subterms, and such that if $f(u_1) \otimes \ldots \otimes f(u_n) \in S_T(T)$ then $u_1 \otimes \ldots \otimes u_n \in S_T(T)$.*

**Lemma 2.** *Let $P$ be a proof which is minimal among all $\otimes$-lazy proofs of $T \vdash w$, and such that the last rule applied in $P$ is of the form $(X)\frac{T \vdash N_1 \ldots T \vdash N_n}{T \vdash w}$, where (X) is one of (UL), (UR), or (D). Then $N_i \in S_T(T)$ for all $i$.*

This has been shown [7] in the setting of *exclusive or* without an homomorphism. The proof is very easily extended (see [13]) to our setting of *ACUNh*.

**Lemma 3.** *Let $P$ be a proof which is minimal among all $\otimes$-lazy proofs of $T \vdash w$, and let $P'$ be a subproof of $P$ with root label $T \vdash N$. If the last rule applied in $P'$ is (P), (C), or (GX) then $N \in S_T(T \cup \{w\})$.*

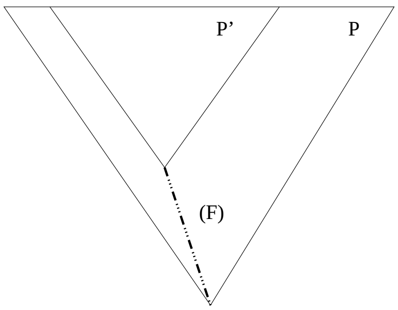
This is a central technical lemma. The proof is given in [13].

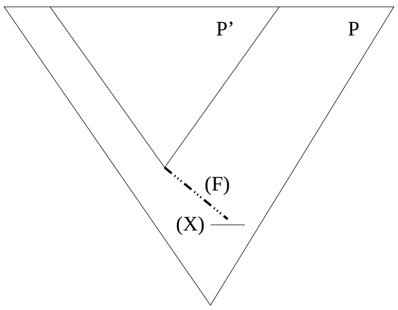
**Lemma 4.** *Let $P$ be a proof which is minimal among all $\otimes$-lazy proofs of $T \vdash w$, and let $P'$ be a subproof of $P$ with root label $T \vdash N$ such that the last rule applied in $P'$ is (F). If either*

1. *all nodes from the root of $P'$ to the root of $P$ are (F),*
2. *or if the first successor not labeled by (F) of the root of $P'$ in $P$ is labeled by a rule different from (GX),*

*then $N \in S_T(T \cup \{w\})$.*

The two cases of the lemma can be illustrated like this:




In the right picture, (X) denotes a rule different from (F) and from (GX). The lemma states that (F) nodes are in $S_T(T \cup \{w\})$ as long as they do not produce an hypothesis of a (GX) rule via a succeeding sequence of (F) nodes. This follows easily from Lemma 2 and Lemma 3 (see [13]).

**Example 1** *The following proof of $T = \{u \otimes v, f(v)\} \vdash f(u)$ is minimal:*

$$\dfrac{\dfrac{\dfrac{u \otimes v \in T}{T \vdash u \otimes v}\,(A)}{T \vdash f(u) \otimes f(v)}\,(F) \qquad \dfrac{f(v) \in T}{T \vdash f(v)}\,(A)}{T \vdash f(u)}\,(GX)$$

*We obtain $S_T(T \cup \{w\}) = \{u, v, u \otimes v, f(u), f(v)\}$. This proof is not $S_T$-local since $f(u) \otimes f(v) \notin S_T(T \cup \{w\})$.*

As can be seen in the above example, the problem in defining $S$-locality for a polynomial-size $S$ is to bound the number of applications of the (F) proof rule when constructing hypotheses to a (GX) rule.

### 5.1 Locality in the Binary Case

In the *binary* case, that is when all terms in $S_T(T \cup \{w\})$ have at most two atoms, we can actually find an upper bound for the number of applications of (F).

**Definition 6.** *A term $t$ is* binary *if every $s \in S_T(t)$ either is not headed with $\otimes$, or is of the form $s_1 \otimes s_2$ where $s_1, s_2$ are not headed with $\otimes$. A set of terms is binary if each of its elements is binary. A proof is binary if each of its nodes is labeled by a sequent $T \vdash w$ where $T$ and $w$ are binary.*

**Proposition 1.** *If $T$ and $w$ are binary then every proof which is minimal among the $\otimes$-lazy proofs of $T \vdash w$ is binary.*

We define for any term $t$ the term $Strip_f(f(t)) = Strip_f(t)$, and $Strip_f(t) = t$ if $t$ does not have root symbol $f$. Furthermore, $\#_f(f(t)) = 1 + \#_f(t)$, and $\#_f(t) = 0$ when $t$ is not headed by $f$. In the binary case we associate a one-counter automaton to the set $S_T(T \cup \{w\})$. The idea is that states of the automaton are terms in $Strip_f(\text{atoms}(S_T(T \cup \{w\})))$, and the counter represents the number of applications of $f$ to a term.

**Definition 7.** *Let $T$ be a set of terms such that every term in $T$ has at most two atoms. We partition $T = T_1 \uplus T_2$ where $T_1$ is the set of terms not headed with $\otimes$, and $T_2$ is the set of terms headed with $\otimes$. The* automaton associated with $T$, *abbreviated $A_T$, is a one-counter automaton without input defined as follows:*

*The set of states $Q_T$ of $A_T$ is*

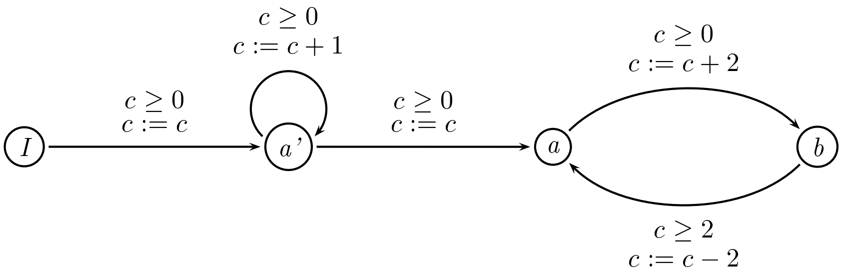
$$\{\text{INIT}\} \cup \{p' \mid p \in Strip_f(T_1)\} \cup \{r \mid r \in Strip_f(T_1) \cup Strip_f(atoms(T_2))\}$$

*where* INIT *is the initial state of $A_T$. The set of transitions is:*

| | *From* | *To* | *Condition* | *Action* |
|---|---|---|---|---|
| $\forall t \in T_1 :$ | INIT | $(Strip_f(t))'$ | $c \geq 0$ | $c := c$ |
| $\forall t \in T_1 :$ | $(Strip_f(t))'$ | $(Strip_f(t))'$ | $c \geq 0$ | $c := c + 1$ |
| $\forall t \in T_1 :$ | $(Strip_f(t))'$ | $Strip_f(t)$ | $c \geq \#_f(t)$ | $c := c$ |
| $\forall t \otimes s \in T_2 :$ | $Strip_f(t)$ | $Strip_f(s)$ | $c \geq \#_f(t)$ | $c := c - \#_f(t) + \#_f(s)$ |

Note that in the last line of the above transition table the statement "$t \otimes s \in T_2$" is to be understood modulo $AC$, such that we obtain from a binary clause a back and a forth transition.

**Example 2** *The automaton $A_T$ for $T = \{a \otimes f^2(b), a\}$ is as follows, where $I$ denotes the initial state:*




One of the two lemmata relating the proof system with our automata construction is:

**Lemma 5.** *Let $T$ be a set of binary terms. For all $t_0, \ldots, t_n \in Strip_f(atoms(T))$ and all natural numbers $c_0, \ldots, c_n$ we have that*

$$A_T \models (t_0, c_0) \to (t_1, c_1) \to \ldots \to (t_n, c_n)$$

*iff there are terms $s_1, \ldots, s_n \in T$ and natural numbers $d_1, \ldots, d_n$ such that:*

1. *for $1 \leq i \leq n$ the term $s_i$ is headed with $\otimes$ and has exactly two atoms, that is $s_i = s_i^1 \otimes s_i^2$*
2. *$\forall 1 \leq i \leq n : f^{d_i}(s_i^1) = f^{c_{i-1}}(t_{i-1})$*
3. *$\forall 1 \leq i \leq n : f^{d_i}(s_i^2) = f^{c_i}(t_i)$*

As a consequence and using the axiom $x \otimes x = 0$, we obtain that

$$\bigoplus_{i=1}^{n} f^{d_i}(s_i)\downarrow = f^{d_1}(s_1^1) \otimes f^{d_n}(s_n^2) = f^{c_0}(t_0) \otimes f^{c_n}(t_n)$$

**Lemma 6.** *Let $A$ be a one-counter automaton and $\pi : (q, c_q) \to^\star (r, c_r)$ a path between the state $q$ with the counter $c_q \geq 0$ and the state $r$ with the counter $c_r \geq 0$. Then there exists a path from $(q, c_q)$ to $(r, c_r)$ such that everywhere along the path the value of the counter is bounded by $p(|A|)$, where $p$ is a polynomial function.*

We believe this lemma to be folklore but were unable to find a proof in the literature. A proof, along with a definition of the polynomial function $p$, is included in the complete version [13]. We can now define:

**Definition 8.** *We define for any finite subset $U$ of $T(\Sigma)$:*

$$S_f(U) = \{f^i(u)\downarrow \mid u \in S_T(U), 0 \leq i \leq p(|A_T|)\}$$

*where the function $p$ is as in Lemma 6.*

Note that the size of $S_f(U)$ is polynomial in the size of $U$. Combining Lemmata 2 through 6 we obtain:

**Lemma 7.** *Let $T \subseteq T(\Sigma)$ and $w \in T(\Sigma)$ be binary, and let $P$ be a proof which is minimal among all $\otimes$-lazy proofs of $T \vdash w$. All nodes of $P$ are in $S_f(T \cup \{w\})$.*

### 5.2 Locality in the General Case

**Definition 9.** *We define for any finite subset $U$ of $T(\Sigma)$:*

$$S_{\otimes}(U) = \{u_1 \otimes \ldots \otimes u_n | u_1, \ldots, u_n \in S_T(U)\}$$

Note that the size of $S_{\otimes}(T)$ is exponential in the size of $T$.

**Lemma 8.** *Let $M \subseteq T(\Sigma)$, $t_0 \in T(\Sigma)$, and $t_1, \ldots, t_n \in S_T(M)$.*
*If $(t_0 \otimes t_1 \otimes \ldots \otimes t_n)\downarrow \in S_{\otimes}(M)$ then $t_0 \in S_{\otimes}(M)$.*

The easy proof can be found in [13]. This lemma, together with the previous lemmata, is the key for proving the following lemma which states that any proof which is minimal among the $\otimes$-eager proofs of $T \vdash w$ contains only nodes in $S_{\otimes}(T \cup \{w\})$.

**Lemma 9.** *The Dolev-Yao proof system in case of ACUNh is $S_{\otimes}$-local.*

## 6 One-Step Deducibility in Case of *ACUNh*

We follow the well-known method for solving unification problems modulo $AC$-like theories [16]. We only show how to decide one-step deducibility for the family of rules (GX), since checking one-step deducibility for the remaining deduction rules is straightforward. We transform the problem of testing one-step deducibility into the satisfiability of a system of linear Diophantine equations.

Let $t \in T(\Sigma)$ and $u \in T(\Sigma)$ not headed with $\otimes$. We denote by $\delta(u,t)$ the number of occurrences of $u$ in atoms($t$) (which is, in the case *ACUNh*, either 0 or 1).

**Definition 10.** *Let $s \in T(\Sigma)$ and $T = \{t_1, \ldots, t_n\}$ be a finite subset of $T(\Sigma)$. Let atoms($T \cup \{s\}$) $= \{a_1, \ldots, a_m\}$. The equation system $D(T,s)$ over the variables $x_1, \ldots, x_n$ is*

$$D(T,s) := \bigwedge_{i=1}^{m} \sum_{j=1}^{n} \delta(a_i, t_j) * x_j = \delta(a_i, s)$$

**Example 3** *Let $T = \{a_1 \otimes a_2 \otimes a_3, a_1 \otimes a_4, a_2 \otimes a_4\}$ and $s = a_1 \otimes a_2$, where all the $a_i$ are not headed with $\otimes$. We introduce numerical variables $x_1, x_2, x_3$, that is one numerical variable for each element of $T$ :*

$$\begin{array}{ll} x_1 & \text{for } a_1 \otimes a_2 \otimes a_3 \\ x_2 & \text{for } a_1 \otimes a_4 \\ x_3 & \text{for } a_2 \otimes a_4 \end{array}$$

*For every atom $a_i$ we create an equation. This yields the following equation system:*

$$\begin{cases} a_1 : x_1 + x_2 = 1 \\ a_2 : x_1 + x_3 = 1 \\ a_3 : x_1 = 0 \\ a_4 : x_2 + x_3 = 0 \end{cases}$$

**Lemma 10.** *Let $s \in T(\Sigma)$ and $T$ a finite subset of $T(\Sigma)$. Then $s$ is deducible with one application of a rule (GX) from $T$ if and only if $D(T, s)$ is solvable over $\mathbb{Z}/2\mathbb{Z}$.*

Since satisfiability of a system of linear Diophantine equations over $\mathbb{Z}/2\mathbb{Z}$ is in PTIME [12], we obtain from Lemma 10, Theorem 1, and Lemma 9 that:

**Theorem 2.** *The question whether $T \vdash w$ is deducible form $T$ in case of the rewrite system ACUNh is decidable in EXPTIME.*

In the binary case we obtain from Lemma 10, Theorem 1, Lemma 7, and Proposition 1 that:

**Theorem 3.** *The question whether $T \vdash w$ is deducible form $T$ in case of the rewrite system ACUNh, where $T$ and $w$ are binary, is decidable in PTIME.*

## 7 Variants and Extensions

### 7.1 The Rewrite System *ACh*

The case of the rewrite system *ACh* is much simpler than the case *ACUNh* since with *ACh* it is not possible that terms are canceled out when applying the constructor $\otimes$. Hence we do not get the difficulty seen in Example 1.

**Lemma 11.** *The extended Dolev-Yao proof system in case of ACh is $S_T$-local.*

The downside is that, in order to decide one-step deducibility, we now have to solve linear Diophantine equation systems over $\mathbb{N}$. This problem is in general NP-complete [17]. Furthermore, it is quite easy to reduce satisfiability of linear Diophantine equations over $\mathbb{N}$ to the intruder deduction problem modulo *ACh*.

An exception is again the binary case, where one-step deducibility is decidable in polynomial time (which is trivial to prove in this case). We hence obtain:

**Theorem 4.** *The problem whether $T \vdash w$ in case of the rewrite system ACh is NP-complete, and decidable in PTIME if we restrict the problem to the binary case.*

### 7.2 The Rewrite System *AGh*

The case of the rewrite system *AGh* is very similar to the case of *ACUNh*. The lemmata and techniques can be adapted easily when we change the definitions of $S_T$, $S_f$, and $S_\otimes$ and require now in addition that they are closed under application of the inversion function and subsequent normalization of the term.

We can test one-step deducibility essentially as in Section 6. The major difference is that we now have to check our equation system $D(T, s)$ for satisfiability in $\mathbb{Z}$, which again is in PTIME [18].

**Theorem 5.** *The problem whether $T \vdash w$ in case of the rewrite system AGh is decidable in EXPTIME, and decidable in PTIME if we restrict the problem to the binary case.*

### 7.3 Extension to an Encryption Operation which is Homomorphic over $\otimes$

This extension consists of replacing, in the three rewrite systems given at the end of Section 2, the rewrite rule

$$f(x \otimes y) \to f(x) \otimes f(y)$$

by the new rule

$$\{x \otimes y\}_z \to \{x\}_z \otimes \{y\}_z$$

On a technical level, this introduces the additional difficulty that we can now decompose in certain cases a sum built by $\otimes$, as for instance

$$\text{(D)} \quad \frac{T \vdash \{a\}_k \otimes \{b\}_k \otimes \{c\}_k \qquad T \vdash k}{T \vdash a \otimes b \otimes c}$$

However, we obtain for this extension lemmata and results which are analogous to the ones in the previous sections. The construction of the automaton for the binary case explained in Section 5 has now to be generalized since we now have an a priori infinite family of homomorphisms. In the case of Section 5 one counter was enough to count the number of applications of the homomorphic function $f$. In the extended case, we have to represent the sequence of encryption keys used in a stack of encryption operations, which can now be done with a pushdown automaton. We can find a lemma analogous to Lemma 6 also for the class of pushdown automata. The only remaining difficulty is to show that the stack alphabet, which consists of the encryption keys used in a minimal and $\otimes$-lazy proof, is finite. This is not obvious since we may use any term as an encryption key. However, we obtain easily by the Lemmata which correspond to Lemmata 2, 3, and 4 that:

**Lemma 12.** *Let $P$ be a proof which is minimal among the $\otimes$-lazy proofs of $T \vdash w$. All the encryption keys used in the proof $P$ are in $S_T(T \cup \{w\})$.*

As a consequence, the Theorems 2, 3, 4, and 5 still hold for this extension.

## 8 Conclusion

A summary of the results obtained on the complexity of the intruder deduction problem modulo $AC$-like equational theories with homomorphism is given in the following table. The results for homomorphism only (without $AC$ axioms) have been shown in a different paper [7] and are here cited only for completeness.

*Complexity of the intruder deduction problem*

| | Binary case | General case |
|---|---|---|
| $h$ | *PTIME* [7] | |
| $ACh$ | *PTIME* | *NP*-Complete |
| $ACUNh$ | *PTIME* | *EXPTIME* |
| $AGh$ | *PTIME* | *EXPTIME* |

The reason for the high complexity in the general case is a different one for the different equational theories considered, as shown in the following table:

*Complexity in the general case*

| | Computation of subterms | One step deducibility | General deducibility |
|---|---|---|---|
| $h$ | *PTIME* [7] | *PTIME* [7] | *PTIME* [7] |
| $ACh$ | *PTIME* | *NP-Complete* | *NP-Complete* |
| $ACUNh$ | *EXPTIME* | *PTIME* | *EXPTIME* |
| $AGh$ | *EXPTIME* | *PTIME* | *EXPTIME* |

As future work, we plan to investigate the case of an active intruder. We can yet observe that it has been shown in [8] that decidability of unification modulo an equational theory $E$ is a necessary condition for the decidability of the security of a protocol for a bounded number of sessions and in presence of this equational theory $E$. Since unification modulo $AC$ plus homomorphism is known undecidable [16], the security against active attackers is undecidable at least for this equational theory as well.

**Acknowledgments** We are grateful for the numerous useful remarks and hints we obtained from our colleagues at LSV, in particular (in alphabetic order) Mathieu Baudet, Hubert Comon-Lundh, Stéphanie Delaune, Jean Goubault-Larrecq, Florent Jacquemard, Claudine Picaronny, and Philippe Schnoebelen. Philippe provided us with a first proof of Lemma 6.

## References

[1] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, Jan. 1999.

[2] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *Proc. 1st IFIP International Conference on Theoretical Computer Science (IFIP–TCS)*, volume 1872 of *LNCS*, pages 3–22. Springer-Verlag, 2000.

[3] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[4] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with XOR. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 261–270, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.

[5] J. Clark and J. Jacob. A survey of authentication protocol literature. `http://www.cs.york.ac.uk/~jac/papers/drareviewps.ps`, 1997.

[6] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 271–280, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.

[7] H. Comon-Lundh and R. Treinen. Easy intruder deductions. In N. Dershowitz, editor, *Verification: Theory & Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday*, volume 2772 of *LNCS*, pages 225–242. Springer-Verlag, 2003.

[8] V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. Research Report LSV-04-15, LSV, ENS de Cachan, Sept. 2004. Available at `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rapports-year-2004-list.php`.

[9] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B - Formal Models and Semantics, chapter 6, pages 243–320. Elsevier Science Publishers and The MIT Press, 1990.

[10] D. Dolev and A. Yao. On the security of public-key protocols. In *Transactions on Information Theory*, volume 29, pages 198–208. IEEE Comp. Soc. Press, 1983.

[11] F. Jacquemard. Security protocols open repository. Available at `http://www.lsv.ens-cachan.fr/spore/index.html`.

[12] E. Kaltofen, M. S. Krishnamoorthy, and B. D. Saunders. Fast parallel computation of hermite and smith forms of polynomial matrices. *SIAM J. Algebraic Discrete Methods*, 8(4):683–690, 1987.

[13] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for *ac*-like equational theories with homomorphisms. Research Report LSV-04-16, LSV, ENS de Cachan, Nov. 2004. Available at `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rapports-year-2004-list.php`.

[14] G. Lowe. An attack on the Needham-Schroeder public key authentication protocol. *Information Processing Letters*, 56(3):131–133, 1995.

[15] D. A. McAllester. Automatic recognition of tractability in inference relations. *JACM*, 40(2):284–303, April 1993.

[16] P. Narendran. Solving linear equations over polynomial semirings. In *Proc. of 11th Annual Symposium on Logic in Computer Science (LICS)*, pages 466–472, July 1996.

[17] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[18] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, 1986.

[19] M. Turuani. Personal communication, 2003.