# Reducing Protocol Analysis with XOR to the XOR-free Case in the Horn Theory Based Approach

**Ralf Küsters** · **Tomasz Truderung**

**Abstract** In the Horn theory based approach for cryptographic protocol analysis, cryptographic protocols and (Dolev-Yao) intruders are modeled by Horn theories and security analysis boils down to solving the derivation problem for Horn theories. This approach and the tools based on this approach, including ProVerif, have been very successful in the automatic analysis of cryptographic protocols. However, dealing with the algebraic properties of operators, such as the exclusive OR (XOR), which are frequently used in cryptographic protocols has been problematic. In particular, ProVerif cannot deal with XOR.

In this paper, we show how to reduce the derivation problem for Horn theories with XOR to the XOR-free case. Our reduction works for an expressive class of Horn theories. A large class of intruder capabilities and protocols that employ the XOR operator can be modeled by these theories. Our reduction allows us to carry out protocol analysis using tools, such as ProVerif, that cannot deal with XOR, but are very efficient in the XOR-free case. We implemented our reduction and, in combination with ProVerif, used it for the fully automatic analysis of several protocols that employ the XOR operator. Among others, our analysis revealed a new attack on an IBM security module.

**Keywords** Security Protocols · XOR · Automated verification

## 1 Introduction

In the Horn theory based approach for cryptographic protocol analysis, cryptographic protocols and the so-called Dolev-Yao intruder are modeled by Horn theories. The security analysis, including the analysis of secrecy and authentication properties, then essentially boils down to solving the derivation problem for Horn theories, i.e., the question whether a

This is an extended version of a paper that first appeared at CCS 2008 [KT08a]. This work was partially supported by the DFG under Grant KU 1434/4-2, the SNF under Grant 200021-116596, and the Polish Ministry of Science and Education under Grant 3 T11C 042 30.

Ralf Küsters,
University of Trier, Germany
E-mail: kuesters@uni-trier.de

Tomasz Truderung
University of Trier, Germany; on leave from Wrocław University, Poland
E-mail: truderung@uni-trier.de

certain fact is derivable from the Horn theory. This kind of analysis takes into account that an unbounded number of protocol sessions may run concurrently. While the derivation problem is undecidable in general, there are very successful automatic analysis tools, ProVerif [Bla01] being one of the most prominent ones among them, that work well in practice.

However, dealing with the algebraic properties of operators, such as the exclusive OR (XOR), which are frequently used in cryptographic protocols, has been problematic in the Horn theory approach. While ProVerif has been extended to deal with certain algebraic properties in [BAF08], associative operators, which in particular include XOR, are still out of the scope. Even though there exist some decidability results for the derivation problem in certain classes of Horn theories with XOR [CLC03, VSS05, CKS07], the decision procedures have not led to practical implementations yet, except for the very specific setting in [CKS07] (see the related work).

The goal of this work is therefore to come up with a practical approach that allows for the automatic analysis of a wide range of cryptographic protocols with XOR, in a setting with an unbounded number of protocol sessions and no bounds on the size of messages. Our approach is to reduce this problem to the one without XOR, i.e., to the simpler case without algebraic properties. This simpler problem can then be solved by tools, such as ProVerif, that a priori cannot deal with XOR, but are very efficient in solving the XOR-free case. More precisely, the contribution of this paper is as follows.

**Contribution of this Paper.** We consider an expressive class of (unary) Horn theories, called $\oplus$-linear. A Horn theory is $\oplus$-linear, if for every Horn clause in this theory, except for the clause that models the intruder's ability to apply the XOR operator ($I(x), I(y) \rightarrow I(x \oplus y)$), the terms that occur in these clauses are $\oplus$-linear. A term is $\oplus$-linear if for every subterm of the form $t \oplus t'$ in this term, it is true that $t$ or $t'$ does not contain variables. We do not put any other restriction on the Horn theories. In particular, our approach will allow us to deal with all cryptographic protocols and intruder capabilities that can be modeled as $\oplus$-linear Horn theories. Note that if a Horn clause does not contain the symbol $\oplus$, then it is $\oplus$-linear by definition.

We show that the derivation problem for $\oplus$-linear Horn theories with XOR can be reduced to a purely syntactic derivation problem, i.e., a derivation problem where the algebraic properties of XOR do not have to be considered anymore. Now, the syntactic derivation problem can be solved by highly efficient tools, such as ProVerif, which cannot deal with XOR.

Using ProVerif, we apply our two step approach—first reduce the problem, then run ProVerif on the result of the reduction—to the analysis of several cryptographic protocols that use the XOR operator in an essential way. The experimental results demonstrate that our approach is practical. In one case, we found a new attack on a protocol. We point the reader to [KT08b] for our implementation and the specifications of the protocols that we analyzed.

We note that a potential alternative to our approach is to perform unification modulo XOR instead of syntactic unification in a resolution algorithm such as the one employed by ProVerif. Whether or not this approach is practical is an open problem. The main difficulty is that unification modulo XOR is much more inefficient than syntactic unification; it is NP-complete rather than linear and, in general, there does not exist a (single) most general unifier.

**Related Work.** In [CLC03, VSS05, SV09], classes of Horn theories (security protocols) are identified for which the derivation problem modulo XOR is shown to be decidable. These classes are orthogonal to the one studied in this paper. While $\oplus$-linearity is not required, other restrictions are put on the Horn clauses, in particular linearity on the occurrence of vari-

ables. The classes in [CLC03, VSS05] do, for example, not contain the Recursive Authentication and the SK3 protocol, which, however, we can model. To the best of our knowledge, the decision procedures proposed in [CLC03, VSS05, SV09] have not been implemented. The procedure proposed in [CLC03] has non-elementary runtime.

In [Ste05, CKS07, CDS07], the IBM 4758 CCA API, which we also consider in our experiments, has been analyzed. Notably, in [CKS07] a decision procedure, along with an implementation, is presented for the automatic analysis of a class of security protocols which contains the IBM 4758 CCA API. However, the protocol class and the decision procedure is especially tailored to the IBM 4758 CCA API. The only primitives that can be handled are the XOR operator and symmetric encryption. All other primitives, such as pairing, public-key encryption, and hashing, are out of the scope of the method in [CKS07]. The specification of the IBM 4758 CCA API in [CKS07] is hard coded in a C implementation.

In [BAF08], it is described how the basic resolution algorithm used in ProVerif can be extended to handle some equational theories. However, as already mentioned in that work, associative operators, such as XOR, are out of the scope of this extension.

In [CLD05], the so-called finite variant property has been studied for XOR and other operators. It has been used (implicitly or explicitly) in other works [CLS03, CLC03], and also plays a role in our work.

In [CKRT03, CLS03, KT07], decision procedures for protocol analysis with XOR w.r.t. a *bounded* (rather than an unbounded) number of sessions are presented. The notion of $\oplus$-linearity that we use is taken from the work in [KT07]. That work also contains some reduction argument. However, our work is different to [KT07] in several respects: First, of course, our approach is for an *unbounded* number of sessions, but it is not guaranteed to terminate. Second, the class of protocols (and intruder capabilities) we can model in our setting is much more general than the one in [KT07]. Third, the reduction presented in [KT07] heavily depends on the bounded session assumption; the argument would not work in our setting. Fourth, the reduction presented in [KT07] does not seem to be practical.

**Structure of this Paper.** In the following section, we recall the Horn theory approach and introduce a running example. The main technical contribution of this paper is presented in Sections 3 and 4. Our implementation and experimental results are discussed in Section 5. We conclude in Section 6. Some proofs are postponed to the appendix.

## 2 Preliminaries

In this section, we introduce Horn theories modulo the XOR operator and illustrate how these theories are used to model the so-called Dolev-Yao intruder and cryptographic protocols by a running example.

### 2.1 Horn theories

Let $\Sigma$ be a finite signature, i.e., a finite set of function symbols with associated arities, and $V$ be a set of variables. The set of terms over $\Sigma$ and $V$ is defined as usual. By $\mathrm{var}(t)$ we denote the set of variables that occur in the term $t$. We assume $\Sigma$ to contain the binary function symbol $\oplus$ (*exclusive OR*), as well as a constant 0. To model cryptographic protocols, $\Sigma$ typically also contains constants (*atomic messages*), such as principal names, nonces, and keys, unary function symbols, such as $\mathsf{hash}(\cdot)$ (*hashing*) and $\mathsf{pub}(\cdot)$ (*public key*), and binary function symbols, such as $\langle\cdot,\cdot\rangle$ (*pairing*), $\{\cdot\}_\cdot$ (*symmetric encryption*), and $\{\!|\cdot|\!\}_\cdot$ (*public key*

*encryption*). The signature $\Sigma$ may also contain any other free function symbol, such as various kinds of signatures and MACs. We only require that the corresponding intruder rules are $\oplus$-linear (see Section 3), which, for example, rules that do not contain the symbol $\oplus$ are.

*Ground terms*, i.e. terms without variables, are called *messages*. For a unary predicate $q$ and a (ground) term $t$ we call $q(t)$ a *(ground) atom*. A *substitution* is a finite set of pairs of the form $\sigma = \{t_1/x_1, \ldots, t_n/x_n\}$, where $t_1, \ldots, t_n$ are terms and $x_1, \ldots, x_n$ are variables. The set $\mathrm{dom}(\sigma) = \{x_1, \ldots, x_n\}$ is called the domain of $\sigma$. We define $\sigma(x) = x$ if $x \notin \mathrm{dom}(\sigma)$. The application $t\sigma$ of $\sigma$ to a term/atom/set of terms $t$ is defined as usual.

We call a term *standard* if its top-symbol is not $\oplus$; otherwise, it is called *non-standard*. For example, the term $\langle a, b \oplus a \rangle$ is standard, while $b \oplus a$ is non-standard. A term is $\oplus$*-free* if it does not contain the XOR operator.

The notion of a *subterm* is defined as usual. For example, $a$ and $x \oplus y$ are subterms of $\langle a \oplus \{(x \oplus y) \oplus z\}_y, b \rangle$, but $y \oplus z$ is not.

A non-standard subterm $s$ of $t$ is called *complete*, if either $s = t$ or $s$ occurs in $t$ as a direct subterm of some standard term. For instance, for $t = \langle a \oplus \{(x \oplus y) \oplus z\}_y, b \rangle$, the terms $a \oplus \{(x \oplus y) \oplus z\}_y$ and $(x \oplus y) \oplus z$ are complete non-standard subterms of $t$, but $x \oplus y$ is not.

To model the algebraic properties of the exclusive OR (XOR), we consider the congruence relation $\sim$ on terms induced by the following equational theory (see, e.g., [CLS03, CKRT03]):

$$x \oplus y = y \oplus x \qquad (x \oplus y) \oplus z = x \oplus (y \oplus z) \tag{1}$$

$$x \oplus x = 0 \qquad x \oplus 0 = x \tag{2}$$

For example, we have that $t_{ex} = a \oplus b \oplus \{0\}_k \oplus b \oplus \{c \oplus c\}_k \oplus c \sim a \oplus c$. Note that due to the associativity of $\oplus$ we often omit brackets and simply write $a \oplus b \oplus c$ instead of $(a \oplus b) \oplus c$ or $a \oplus (b \oplus c)$.

For atoms $q(t)$ and $q'(t')$, we write $q(t) \sim q'(t')$ if $q = q'$ and $t \sim t'$. We say that two terms are *equivalent modulo AC*, where AC stands for associativity and commutativity, if they are equivalent modulo (1). A term is $\oplus$*-reduced* if modulo AC, the equations (2), when interpreted as reductions from left to right, cannot be applied. Clearly, every term can be turned into $\oplus$-reduced form and this form is uniquely determined modulo AC. For example, both $a \oplus c$ and $c \oplus a$ are $\oplus$-reduced forms of $t_{ex}$.

A *Horn theory* $T$ is a finite set of *Horn clauses* of the form $a_1, \ldots, a_n \to a_0$, where $a_i$ is an atom for every $i \in \{0, \ldots, n\}$. We assume that the variables that occur on the right-hand side of a Horn clause also occur on the left-hand side. If $n = 0$, i.e., the left-hand side of the clause is always true, we call the Horn clause $a_0$ a *fact*. Note that, by the above assumption, every fact is ground.

Given a Horn theory $T$ and a ground atom $a$, we say that *$a$ can syntactically be derived from $T$* (written $T \vdash a$) if there exists a *derivation* for $a$ from $T$, i.e., there exists a sequence $\pi = b_1, \ldots, b_l$ of ground atoms such that $b_l = a$ and for every $i \in \{1, \ldots, l\}$ there exists a substitution $\sigma$ and a Horn clause $a_1, \ldots, a_n \to a_0$ in $T$ such that $a_0\sigma = b_i$ and for every $j \in \{1, \ldots, n\}$ there exists $k \in \{1, \ldots, i-1\}$ with $a_j\sigma = b_k$. In what follows, we refer to $b_i$ by $\pi(i)$ and to $b_1, \ldots, b_i$ by $\pi_{\leq i}$. The *length* $l$ of a derivation $\pi$ is referred to by $|\pi|$.

We call a sequence $b_1, \ldots, b_l$ of ground atoms an *incomplete syntactic derivation of $a$ from $T$* if $b_l = a$ and $T \cup \{b_1, \ldots, b_{i-1}\} \vdash b_i$ for every $i \in \{1, \ldots, l\}$ (note that this is equivalent to $T \vdash b_i$, for every $i \in \{1, \ldots, l\}$).

Similarly, we write $T \vdash_\oplus a$ if there exists a *derivation of $a$ from $T$ modulo XOR*, i.e., there exists a sequence $b_1, \ldots, b_l$ of ground atoms such that $b_l \sim a$ and for every $i \in \{1, \ldots, l\}$ there exists a substitution $\sigma$ and a Horn clause $a_1, \ldots, a_n \to a_0$ in $T$ such that $a_0\sigma \sim b_i$ and

$$\begin{aligned}
\mathrm{I}(x) &\to \mathrm{I}(\mathsf{hash}(x)) & \mathrm{I}(x),\mathrm{I}(y) &\to \mathrm{I}(\langle x,y\rangle)\\
\mathrm{I}(\langle x,y\rangle) &\to \mathrm{I}(x) & \mathrm{I}(\langle x,y\rangle) &\to \mathrm{I}(y)\\
\mathrm{I}(x),\mathrm{I}(y) &\to \mathrm{I}(\{x\}_y), & \mathrm{I}(\{x\}_y),\mathrm{I}(y) &\to \mathrm{I}(x)\\
\mathrm{I}(x),\mathrm{I}(\mathsf{pub}(y)) &\to \mathrm{I}(\{\!|x|\!\}_{\mathsf{pub}(y)}), & \mathrm{I}(\{\!|x|\!\}_{\mathsf{pub}(y)}),\mathrm{I}(y) &\to \mathrm{I}(x)\\
\mathrm{I}(x),\mathrm{I}(y) &\to \mathrm{I}(x\oplus y)
\end{aligned}$$

**Fig. 1** Intruder Rules.

for every $j \in \{1,\ldots,n\}$ there exists $k \in \{1,\ldots,i-1\}$ with $a_j\sigma \sim b_k$. *Incomplete derivations modulo XOR* are defined analogously to the syntactic case.

Given $T$ and $a$, we call the problem of deciding whether $T \vdash a$ ($T \vdash_\oplus a$) is true, the *deduction problem (modulo XOR)*. In case $T$ models a protocol and the intruder (as described below), the fact that $T \vdash_\oplus a$, with $a = \mathrm{I}(t)$, is *not* true means that the term $t$ is secret, i.e., the intruder cannot get hold of $t$ even when running an unbounded number of sessions of the protocol, with no bound on the size of messages, and using algebraic properties of the XOR operator.

## 2.2 Modeling Protocols by Horn theories

Following [Bla01], we now illustrate how Horn theories can be used to analyze cryptographic protocols, where, however, we take the XOR operator into account. As mentioned in the introduction, the Horn theory approach allows us to analyze secrecy properties of protocols w.r.t. an unbounded number of sessions and with no bound on the message size in a fully automatic and sound way. However, the algorithms are not guaranteed to terminate and may produce false attacks.

A Horn theory for modeling protocols and the (Dolev-Yao) intruder uses only the predicate I. The fact $\mathrm{I}(t)$ means that the intruder may be able to obtain the message $t$. The fundamental property is that if $\mathrm{I}(t)$ cannot be derived from the set of clauses, then the protocol preserves the secrecy of $t$. The Horn theory consists of three sets of Horn clauses: the initial intruder facts, the intruder rules, and the protocol rules. The set of *initial intruder facts* represents the initial intruder knowledge, such as names of principals and their public keys. The clauses in this set are facts, e.g., $\mathrm{I}(a)$ (the intruder knows the name $a$) and $\mathrm{I}(\mathsf{pub}(sk_a))$ (the intruder knows the public key of $a$, with $sk_a$ being the corresponding private key). The set of *intruder rules* represents the intruders ability to derive new messages. For the cryptographic primitives mentioned above, the set of intruder rules consists of the clauses depicted in Figure 1. The last clause in this figure will be called the $\oplus$-*rule*. It allows the intruder to perform the XOR operation on arbitrary messages. The set of *protocol rules* represents the actions performed in the actual protocol. The $i$-th protocol step of a principal is described by a clause of the form $\mathrm{I}(r_1),\ldots,\mathrm{I}(r_i) \to \mathrm{I}(s_i)$ where the terms $r_j$, $j \in \{1,\ldots,i\}$, describe the (patterns of) messages the principal has received in the previous $i-1$ steps plus the (pattern of the) message in the $i$-th step. The term $\mathrm{I}(s_i)$ is the (pattern of) the $i$-th output message of the principal. Given a protocol $P$, we denote by $T_P$ the Horn theory that comprises all three sets mentioned above.

Below we illustrate the above by a simple example protocol, which we will use as a running example throughout this paper. Applications of our approach to more complex pro-

tocols are presented in Section 5.2. We emphasize that the kind of Horn theories outlined above are only an example of how protocols and intruders can be modeled. As already mentioned in the introduction, our methods applies to all $\oplus$-linear Horn theories.

### 2.3 Running example

We consider a protocol that was proposed in [CKRT03]. It is a variant of the Needham-Schroeder-Lowe protocol in which XOR is employed. The informal description of the protocol, which we denote by $P_{NSL_\oplus}$, is as follows:

$$\begin{array}{lll} 1. & A \to B: & \{\!|\langle N, A\rangle|\!\}_{\mathsf{pub}(sk_B)} \\ 2. & B \to A: & \{\!|\langle M, N \oplus B\rangle|\!\}_{\mathsf{pub}(sk_A)} \\ 3. & A \to B: & \{\!|M|\!\}_{\mathsf{pub}(sk_B)} \end{array}$$

where $A$, $B$ are participant names and $N$, $M$ are nonces generated by $A$ and $B$, respectively. As noted in [CKRT03], this protocol is insecure; a similar attack as the one on the original Needham-Schroeder protocol can be mounted, where, however, now the algebraic properties of XOR are exploited.

To illustrate how this protocol can be modeled in terms of Horn theories, let $\mathsf{P}$ be a set of participant names and $\mathsf{H} \subseteq \mathsf{P}$ be the set of names of the honest participants. As proved in [CLC04], for the secrecy property it suffices to consider the case $\mathsf{P} = \{\mathsf{a}, \mathsf{b}\}$ and $\mathsf{H} = \{\mathsf{a}\}$. In the following, $sk_a$, for $a \in \mathsf{P}$, denotes the private key of $a$, $n(a,b)$ denotes the nonce sent by $a \in \mathsf{P}$ to $b \in \mathsf{P}$ in message 1., and $m(b,a)$ denotes the nonce generated by $b$ and sent to $a$ in message 2.

The initial intruder knowledge is the following set of facts:

$$\{\mathrm{I}(a) \mid a \in \mathsf{P}\} \cup \{\mathrm{I}(\mathsf{pub}(sk_a)) \mid a \in \mathsf{P}\} \cup \{\mathrm{I}(sk_a) \mid a \in \mathsf{P} \setminus \mathsf{H}\}$$

The intruder rules are those depicted in Figure 1. The first step of the protocol performed by an honest principal is modeled by the facts:

$$\mathrm{I}(\{\!|\langle n(a,b), a\rangle|\!\}_{\mathsf{pub}(sk_b)})$$

for $a \in \mathsf{H}$, $b \in \mathsf{P}$. Note that it is not necessary to model messages sent by dishonest principals, since these are taken care of by the actions that can be performed by the intruder.

The second step of the protocol performed by an honest principal is modeled by the clauses:

$$\mathrm{I}(\{\!|\langle x, a\rangle|\!\}_{\mathsf{pub}(sk_b)}) \to \mathrm{I}(\{\!|\langle m(b,a), x \oplus b\rangle|\!\}_{\mathsf{pub}(sk_a)}) \tag{3}$$

for $b \in \mathsf{H}$, $a \in \mathsf{P}$. The third step of the protocol performed by an honest principal is modeled by the clauses:

$$\mathrm{I}(\{\!|\langle y, n(a,b) \oplus b\rangle|\!\}_{\mathsf{pub}(sk_a)}) \to \mathrm{I}(\{\!|y|\!\}_{\mathsf{pub}(sk_b)}) \tag{4}$$

for $a \in \mathsf{H}$, $b \in \mathsf{P}$. The set of Horn clauses defined above is denoted by $T_{P_{NSL_\oplus}}$. It is not hard to verify that we have $T_{P_{NSL_\oplus}} \vdash_\oplus m(b,a)$ for every $a, b \in \mathsf{H}$. In fact, secrecy of the nonces sent by an honest responder to an honest initiator is not guaranteed by this protocol [CKRT03].

## 3 Dominated Derivations

In Section 4, we show how to reduce the deduction problem modulo XOR to the one without XOR for $\oplus$-linear Horn theories, introduced below. This reduction allows us to reduce the problem of checking secrecy for protocols that use XOR to the case of protocols that do not use XOR. The latter problem can then be solved by tools that cannot deal with XOR, such as ProVerif. The class of protocol and intruder capabilities that we can handle this way is quite large: It contains all protocol and intruder rules that are $\oplus$-linear.

In this section, we prove a proposition that will be the key to the reduction. Before we state the proposition, we need to introduce $\oplus$-linear Horn theories and some further terminology.

A term is $\oplus$-*linear* if for each of its non-standard subterms of the form $t_1 \oplus \cdots \oplus t_n$, where $t_1, \ldots, t_n$ are standard, all the terms $t_1, \ldots, t_n$ but one are ground and $\oplus$-free. For example, if $x, y, z$ are variables and $a, b$ constants, the term $t^1_{ex} = \langle a, a \oplus \langle x, y \rangle \rangle$ is $\oplus$-linear, while the terms $t^2_{ex} = \langle a, a \oplus \langle x, y \rangle \oplus z \rangle$ and $t^3_{ex} = a \oplus \langle a, a \oplus b \rangle \oplus z$ are not. We note that the results presented in this paper also hold if $\oplus$-freeness is not required, and hence, for example, $t^3_{ex}$ would be considered to be $\oplus$-linear. However, requiring $\oplus$-freeness slightly simplifies the proofs and does not seem to restrict the applicability of our approach in practice.

A Horn clause is called $\oplus$-linear if each term occurring in the clause is $\oplus$-linear. A Horn theory is $\oplus$-linear if each clause in this theory, except for the $\oplus$-rule (see Fig. 1), is $\oplus$-linear. In particular, given a protocol $P$, the induced theory $T_P$ is $\oplus$-linear if the sets of protocol and intruder rules, except for the $\oplus$-rule, are. A derivation is $\oplus$-linear if all terms occurring in the derivation are.

Our running example is an example of a protocol with an $\oplus$-linear Horn theory (note that, in (3) and (4), $b$ is a constant); other examples are mentioned in Section 5.2. Also, many intruder rules are $\oplus$-linear. In particular, all those that do not contain the XOR symbol. For example, in addition to the cryptographic primitives mentioned in Figure 1, other primitives, such as various kinds of signatures, encryption with prefix properties, and MACs have $\oplus$-linear intruder rules.

Besides $\oplus$-linearity, we also need a more fine-grained notion: $\mathsf{C}$-domination. Let $\mathsf{C}$ be a finite set of ground and $\oplus$-free terms. Let $\mathsf{C}^{\oplus} = \{t \mid \text{there exist } c_1, \ldots, c_n \in \mathsf{C} \text{ such that } t \sim c_1 \oplus \cdots \oplus c_n\}$ be the $\oplus$-closure of $\mathsf{C}$. Finally, let $\tilde{\mathsf{C}} = \{t \mid t \sim t' \text{ for some } t' \in \mathsf{C}\}$. Note that $0 \in \mathsf{C}^{\oplus}$, $\mathsf{C} \subseteq \mathsf{C}^{\oplus}$, and $\tilde{\mathsf{C}} \subseteq \mathsf{C}^{\oplus}$.

Now, a term is $\mathsf{C}$-*dominated* if for each of its non-standard subterms of the form $t_1 \oplus \cdots \oplus t_n$, where $t_1, \ldots, t_n$ are standard, all the terms $t_1, \ldots, t_n$ but one belong to $\mathsf{C}$. For example, the term $t^1_{ex}$ from above is $\{a\}$-dominated, but it is not $\{b\}$-dominated. The terms $t^2_{ex}$ and $t^3_{ex}$ are not $\{a\}$-dominated.

A Horn clause is $\mathsf{C}$-dominated, if the terms occurring in this clause are $\mathsf{C}$-dominated. Finally, a Horn theory $T$ is $\mathsf{C}$-dominated if each clause in $T$, except for the $\oplus$-rule, is $\mathsf{C}$-dominated. For example, we have that the Horn theory $T_{P_{NSL_{\oplus}}}$ of our running example is $\{\mathsf{a}, \mathsf{b}\}$-dominated (recall that $\mathsf{P} = \{\mathsf{a}, \mathsf{b}\}$). A derivation is $\mathsf{C}$-dominated if all terms occurring in the derivation are.

There is an obvious connection between $\oplus$-linearity and $\mathsf{C}$-domination:

**Lemma 1** *For every $\oplus$-linear term/Horn theory/derivation there exists a finite set $\mathsf{C}$ of ground, $\oplus$-free terms such that the term/Horn theory/derivation is $\mathsf{C}$-dominated.*

The set $\mathsf{C}$ mentioned in the lemma could be chosen to be the set of all ground, $\oplus$-free terms occurring in the term/Horn theory/derivation. However, $\mathsf{C}$ should be chosen as small

as possible in order to make the reduction presented in Section 4 more efficient. This is in fact the main motivation for introducing the notion of C-domination.

C-dominated terms can also be characterized in terms of what we call bad terms. We call a non-standard term $t$ *bad* (w.r.t. C), if $t \sim c \oplus t_1 \oplus \ldots \oplus t_n$ for $n > 1$, $c \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct standard terms $t_1, \ldots, t_n \notin \tilde{\mathsf{C}}$, where $t$ and $t'$ are $\oplus$-*distinct* if $t \not\sim t'$. A non-standard term which is not bad is called *good*. The following lemma is easy to prove.

**Lemma 2** *A* C*-dominated term does not contain bad subterms. An* $\oplus$*-reduced term without bad subterms is* C*-dominated.*

Note that if a term without bad subterms is not $\oplus$-reduced, it might not be C-dominated. For example, for $\mathsf{C} = \{a\}$, the term $x \oplus x \oplus a$ does not contain bad subterms, and yet, it is not C-dominated.

The following proposition is the main result of this section and it is the key to our reduction. The proposition states that C-dominated Horn theories always allow for C-dominated derivations. Because of Lemma 1, the proposition applies to all $\oplus$-linear Horn theories.

**Proposition 1** *Let $T$ be a* C*-dominated Horn theory and $b$ be a* C*-dominated fact. If $T \vdash_{\oplus} b$, then there exists a* C*-dominated derivation modulo XOR for $b$ from $T$. Moreover, the substitutions applied in this derivation are* C*-dominated too.*

Before we present the proof of this proposition, we introduce some terminology, which is also used in subsequent sections, and sketch the idea of the proof. We write $t \simeq_{\mathsf{C}} t'$ if $t' \sim c \oplus t$ (or equivalently, $c \oplus t' \sim t$), for some $c \in \mathsf{C}^{\oplus}$.

For the rest of this section we fix a derivation $\pi$ modulo XOR for $b$ from $T$. W.l.o.g. we can assume that each term occurring in $\pi$ is $\oplus$-reduced and that each term in a substitution applied in $\pi$ is $\oplus$-reduced as well.

The key definitions for the proof of Proposition 1 are the following ones:

**Definition 1** For a standard term $t$, the set C, and the derivation $\pi$, we define the *type of $t$ (w.r.t. $\pi$ and* C*)*, written $\tilde{t}$, to be an $\oplus$-reduced element $c$ of $\mathsf{C}^{\oplus}$ such that $\pi(i) \sim \mathrm{I}(c \oplus t)$ for some $i$, and for each $j < i$, it is not true that $\pi(j) \sim \mathrm{I}(c' \oplus t)$ for some $c' \in \mathsf{C}^{\oplus}$. If such an $i$ does not exist, we say that the type of $t$ is undefined.

Note that the type of a term is uniquely determined modulo AC and that equivalent terms (w.r.t. $\sim$) have equivalent types.

In the following definition, we define an operator which replaces standard terms in bad terms which are not in $\tilde{\mathsf{C}}$ by their types. This turns a bad term into a good one. To define the operator, we use the following notation. We write $\varphi_{\oplus}[x_1, \ldots, x_n]$ for a term which is built only from $\oplus$, standard elements of $\tilde{\mathsf{C}}$, and the pairwise distinct variables $x_1, \ldots, x_n$ such that each $x_i$ occurs exactly once in $\varphi_{\oplus}[x_1, \ldots, x_n]$. An example is $\varphi^{ex}_{\oplus}[x_1, x_2, x_3] = ((x_1 \oplus x_2) \oplus (a \oplus x_3))$, where $a \in \tilde{\mathsf{C}}$. For messages $t_1, \ldots, t_n$, we write $\varphi_{\oplus}[t_1, \ldots, t_n]$ for the message obtained from $\varphi_{\oplus}[x_1, \ldots, x_n]$ by replacing every $x_i$ by $t_i$, for every $i \in \{1, \ldots, n\}$. Note that each non-standard term can be expressed in the form $\varphi_{\oplus}[t_1, \ldots, t_n]$ for some $\varphi_{\oplus}$ as above and standard terms $t_1, \ldots, t_n \notin \tilde{\mathsf{C}}$.

**Definition 2** Let C and $\pi$ be as above. For a message $t$, we determine the message $\Delta(t)$ as follows: $\Delta(t)$ is computed by substituting, in a top-down manner, every complete bad subterm of $t$ of the form $\varphi_{\oplus}[t_1, \ldots, t_n]$, for some $\varphi_{\oplus}$ as above and standard terms $t_1, \ldots, t_n \notin \tilde{\mathsf{C}}$, by $\varphi_{\oplus}[\tilde{t}_1, \ldots, \tilde{t}_n]$. $\Delta(t)$ is undefined, if one of the $\tilde{t}_i$, $i \in \{1, \ldots, n\}$, are undefined.

We emphasize that after replacing $\varphi_{\oplus}[t_1,\ldots,t_n]$ by $\varphi_{\oplus}[\tilde{t}_1,\ldots,\tilde{t}_n]$, we continue to apply $\Delta$ in a top-down manner to all proper subterms of $\varphi_{\oplus}[\tilde{t}_1,\ldots,\tilde{t}_n]$. This may be necessary, if some subterm $s \in \tilde{\mathsf{C}}$ of $\varphi_{\oplus}[t_1,\ldots,t_n]$ contains bad terms, which may happen if $s$ is not $\oplus$-reduced.

We will show, in Lemma 8, that if $t$ occurs in $\pi$, then $\Delta(t)$ is defined. Note also that $\Delta$ is defined with respect to the given $\pi$ and $\mathsf{C}$.

Now, the main idea behind the proof of Proposition 1 is to apply $\Delta(\cdot)$ to $\pi$. We then show that (i) $\Delta(\pi)$ is an incomplete $\mathsf{C}$-dominated derivation modulo XOR for $b$ from $T$ and (ii) to obtain a complete derivation only $\mathsf{C}$-dominated terms are needed. The details of the proof are presented next, by a series of lemmas.

**Proof of Proposition 1.** The following lemma gathers important properties of $\Delta$ (see the appendix for the proof).

**Lemma 3** *In the following statements, we always assume that $\Delta$ is defined on the terms that we apply this mapping to.*

(a) *For each term $t$, $\Delta(t)$ (if defined) does not contain bad subterms.*
(b) *If $t$ is $\oplus$-reduced, then $\Delta(t)$ is $\mathsf{C}$-dominated.*
(c) *If $t$ is $\mathsf{C}$-dominated, then $\Delta(t) = t$. In particular, $\Delta(c) = c$, for every $\oplus$-reduced term $c \in \mathsf{C}^{\oplus}$, and $\Delta(\tilde{s}) = \tilde{s}$, for every standard term $s$ for which the type $\tilde{s}$ is defined.*
(d) *$\Delta(c \oplus t) = \Delta(c) \oplus \Delta(t)$, for $c \in \mathsf{C}^{\oplus}$.*
(e) *$\Delta(s\theta) \sim s\Delta(\theta)$, for a $\mathsf{C}$-dominated term $s$ and a substitution $\theta$.*
(f) *Let $s$ and $t$ be terms such that $s \sim t$. Then, $\Delta(s) \sim \Delta(t)$.*

The proof of the following lemma can easily be obtained by structural induction on $s$:

**Lemma 4** *Let $s$ and $t$ be messages such that $s$ is $\oplus$-reduced, $s$ contains a complete bad subterm $s'$, and $s \sim t$. Then, there exists a complete bad subterm $t'$ of $t$ such that $t' \sim s'$.*

The following lemma says that when substituting variables in a $\mathsf{C}$-dominated term, then bad terms that might be introduced by a $\oplus$-reduced substitution cannot be canceled out. The proof of this lemma can be found in the appendix.

**Lemma 5** *Let $r\theta \sim t$, for a term $t$, an $\oplus$-reduced substitution $\theta$, and a $\mathsf{C}$-dominated term $r$. Then, for each complete bad subterm $r'$ of $r\theta$ there exists a complete bad subterm $t'$ of $t$ such that $t' \sim r'$.*

We can now show (see the appendix for the proof) that if an instance of a $\mathsf{C}$-dominated term contains a complete bad subterm, then this subterm (up to $\simeq_{\mathsf{C}}$) must be part of the substitution with which the instance was obtained.

**Lemma 6** *Let $\theta$ be a ground substitution and $s$ be a $\mathsf{C}$-dominated term. Assume that $t$ is a complete bad subterm of $s\theta$. Then, there exists a variable $x$ and a complete bad subterm $t'$ of $\theta(x)$ such that $t' \simeq_{\mathsf{C}} t$.*

The following lemma says that if an instance of a $\mathsf{C}$-dominated Horn clause contains a complete bad subterm on its right-hand side, then this subterm (up to $\simeq_{\mathsf{C}}$) already occurs on the left-hand side.

**Lemma 7** *Assume that $p_1(r_1),\ldots,p_n(r_n) \to p_0(s)$ is a $\mathsf{C}$-dominated Horn clause, $\theta$ is an $\oplus$-reduced ground substitution, $w,u_1,\ldots,u_n$ are $\oplus$-reduced messages such that $w \sim s\theta$ and $u_i \sim r_i\theta$, for $i \in \{1,\ldots,n\}$. If $w'$ is a complete bad subterm of $w$, then there exists a complete bad subterm $u'$ of $u_i$, for some $i \in \{1,\ldots,n\}$, such that $u' \simeq_{\mathsf{C}} w'$.*

*Proof* Suppose that $w'$ is a complete bad subterm of $w$. Because $w \sim s\theta$ and $w$ is $\oplus$-reduced, by Lemma 4, there exists a complete bad subterm $t$ of $s\theta$ with $w' \sim t$. By Lemma 6, there exists a variable $x \in \mathrm{var}(s)$ and a complete bad subterm $t'$ of $\theta(x)$ with $t' \simeq_{\mathsf{C}} t$. Because $x$, as a variable of $s$, has to occur also in $r_i$ for some $i \in \{1,\ldots,n\}$, the term $t'$ is a (not necessarily complete) subterm of $r_i\theta$. Since $r_i$ is $\mathsf{C}$-dominated, there exists a complete subterm $r'$ of $r_i\theta$ with $r' \simeq_{\mathsf{C}} t'$. Now, recall that $t' \simeq_{\mathsf{C}} t$ and $t \sim w'$. It follows that $r' \simeq_{\mathsf{C}} w'$. Furthermore, since $w'$ is bad, so is $r'$. Now, by Lemma 5, there exists a complete bad subterm $u'$ of $u_i$ such that $u' \sim r'$. It follows that $u' \simeq_{\mathsf{C}} w'$. □

The following lemma connects bad terms that occur in a derivation with the types of their subterms.

**Lemma 8** *For every $n \geq 1$, if $\pi(i) \sim \mathrm{I}(c \oplus t_1 \oplus \cdots \oplus t_n)$, for $c \in \mathsf{C}^{\oplus}$ and pairwise $\oplus$-distinct standard terms $t_1,\ldots,t_n \notin \tilde{\mathsf{C}}$, then, for each $k \in \{1,\ldots,n\}$, there exists $j \leq i$ such that $\pi(j) \sim \mathrm{I}(\tilde{t}_k \oplus t_k)$ (hence $\tilde{t}_k$ is defined).*

*Proof* If $n = 1$, then $\mathrm{I}(\tilde{t}_1 \oplus t_1)$ belongs to $\pi_{\leq i}$, by the definition of types.

Now, suppose that $n > 1$. In that case we will show, by induction on $i$, something more than what is claimed in the lemma: If $t$ with $t \sim c \oplus t_1 \oplus \cdots \oplus t_n$, $c \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct standard terms $t_i \notin \tilde{\mathsf{C}}$, occurs as a complete bad subterm in $\pi(i)$, then, for each $k \in \{1,\ldots,n\}$, there exists $j \leq i$ such that $\pi(j) \sim \mathrm{I}(\tilde{t}_k \oplus t_k)$.

Suppose that $t$, as above, occurs as a complete bad subterm in $\pi(i)$.

If there exists $t'$ such that $t' \simeq_{\mathsf{C}} t$ and $t'$ occurs in $\pi_{<i}$ as a complete subterm, then we are trivially done by the induction hypothesis. (Note that $t'$ is bad since $t$ is.) So, suppose that such a $t'$ does not occur in $\pi_{<i}$ as a complete subterm. By Lemma 7, $\pi(i)$ cannot be obtained by a $\mathsf{C}$-dominated Horn clause. Thus, $\pi(i)$ is obtained by the $\oplus$-rule, which means that $\pi(i) = \mathrm{I}(u)$ with $u \sim s \oplus r$ for some $\mathrm{I}(s)$ and $\mathrm{I}(r)$ occurring in $\pi_{<i}$. We may assume that $s \sim d \oplus s_1 \oplus \cdots \oplus s_p$, with $d \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct, $\oplus$-reduced standard terms $s_1,\ldots,s_p \notin \tilde{\mathsf{C}}$, and $r \sim e \oplus r_1 \oplus \cdots \oplus r_q$, with $e \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct, $\oplus$-reduced standard terms $r_1,\ldots,r_q \notin \tilde{\mathsf{C}}$.

According to our assumption, neither $s$ nor $r$ contains a complete subterm $t'$ with $t' \simeq_{\mathsf{C}} t$. In particular, neither $s$ nor $r$ contains $t'$ with $t' \sim t$. So, since $\pi(i) \sim \mathrm{I}(s \oplus r)$ contains $t$ as a complete subterm, it must be the case that $t \sim s \oplus r$. Now, with $t \sim c \oplus t_1 \oplus \ldots \oplus t_n$, as above, and $k \in \{1,\ldots,n\}$ it follows that either $s_l \sim t_k$ or $r_l \sim t_k$, for some $l$. Suppose that the former case holds (the argument is similar for the latter case). If $p > 1$ (and thus $s$ is a bad term), then, by the induction hypothesis, we know that there exists $j < i$ such that $\pi(j) \sim \mathrm{I}(\tilde{s}_l \oplus s_l)$. Since $t_k \sim s_l$, we have that $\tilde{t}_k \sim \tilde{s}_l$, and hence, $\pi(j) \sim \mathrm{I}(\tilde{t}_k \oplus t_k)$. Otherwise, $s \sim d \oplus t_k$, and hence, by the definition of types, there exists $j < i$ with $\pi(j) \sim \mathrm{I}(\tilde{t}_k \oplus t_k)$. □

The following lemma is the key in proving that $\Delta(\pi)$ is an incomplete derivation modulo XOR.

**Lemma 9** *For every $i \leq |\pi|$, if $\mathrm{I}(c \oplus t_1 \oplus \cdots \oplus t_n)$, for some $c \in \mathsf{C}^{\oplus}$ and pairwise $\oplus$-distinct standard terms $t_1,\ldots,t_n \notin \tilde{\mathsf{C}}$, belongs to $\pi_{<i}$, then there is a derivation for $\mathrm{I}(c \oplus \tilde{t}_1 \oplus \cdots \oplus \tilde{t}_n)$ from $T \cup \Delta(\pi_{<i})$ modulo XOR.*

*Proof* Suppose that $\mathrm{I}(c \oplus t_1 \oplus \cdots \oplus t_n)$, as above, belongs to $\pi_{<i}$. First, note that, by Lemma 8, the types $\tilde{t}_1,\ldots,\tilde{t}_n$ are defined.

If $n = 0$ or $n > 1$, then, by the definition of $\Delta$, we have that $\mathrm{I}(c \oplus \tilde{t}_1 \oplus \cdots \oplus \tilde{t}_n) \sim \mathrm{I}(\Delta(c \oplus t_1 \oplus \cdots \oplus t_n))$, and hence, $\mathrm{I}(c \oplus \tilde{t}_1 \oplus \cdots \oplus \tilde{t}_n)$ can be derived from $\Delta(\pi_{<i})$. (Note that we have assumed that $\pi$ is $\oplus$-reduced and therefore so is $c$, which, by Lemma 3-(c), yields $\Delta(c) = c$).

So suppose that $n = 1$. Since we have $\mathrm{I}(c \oplus t_1)$ in $\pi_{<i}$, by the definition of types, we also have $\mathrm{I}(\tilde{t}_1 \oplus t_1)$ in $\pi_{<i}$. Thus, by Lemma 3-(c) and (d), $\mathrm{I}(c \oplus \Delta(t_1))$ and $\mathrm{I}(\tilde{t}_1 \oplus \Delta(t_1))$ are in $\Delta(\pi_{<i})$. From these one obtains $\mathrm{I}(c \oplus \tilde{t}_1)$ by applying the $\oplus$-rule. □

Now, we can finish the proof of Proposition 1. Recall that we assume that $\pi$ is $\oplus$-reduced and that in this derivation we use only $\oplus$-reduced substitutions.

First, note that, because $\pi$ is $\oplus$-reduced, by Lemma 3-(b), $\Delta(\pi)$ is $\mathsf{C}$-dominated. We will now show (*): For each $i \in \{1, \ldots, |\pi|\}$, $\Delta(\pi(i))$ can be derived from $T \cup \Delta(\pi_{<i})$ modulo XOR by using only $\mathsf{C}$-dominated terms. This then completes the proof of Proposition 1.

To prove (*), we consider two cases. It is easy to check that in both cases the applied substitutions are $\mathsf{C}$-dominated.

*Case 1.* $\pi(i)$ is obtained from $\pi_{<i}$ using a $\mathsf{C}$-dominated Horn clause $R = (p_1(s_1), \ldots, p_n(s_n) \to p_0(s_0))$ of $T$: Then there exists a $\oplus$-reduced substitution $\theta$ such that $\pi(i) \sim p_0(s_0\theta)$ and the atoms $p_1(s_1\theta), \ldots, p_n(s_n\theta)$ occur in $\pi_{<i}$ modulo XOR. Thus, by Lemma 3-(f), $p_1(\Delta(s_1\theta)), \ldots, p_n(\Delta(s_n\theta))$ occur in $\Delta(\pi_{<i})$ modulo XOR. Now, by Lemma 3-(e), we have that $\Delta(s_i\theta) \sim s_i(\Delta\theta)$, for every $i \in \{0, \ldots, n\}$. Thus, by applying $R$ with the substitution $\Delta(\theta)$, we obtain $\Delta(\pi(i)) \sim p_0(\Delta(s_0\theta)) \sim p_0(s_0(\Delta(\theta)))$.

*Case 2.* $\pi(i)$ is obtained by the $\oplus$-rule: Hence, there are two atoms $\mathrm{I}(s)$ and $\mathrm{I}(r)$ in $\pi_{<i}$ such that $\pi(i) \sim \mathrm{I}(s \oplus r)$. We may assume that $s \sim c \oplus s_1 \oplus \cdots \oplus s_m$, with $c \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct, $\oplus$-reduced standard terms $s_1, \ldots, s_m \notin \tilde{\mathsf{C}}$, and $r \sim d \oplus r_1 \oplus \cdots \oplus r_l$, with $d \in \mathsf{C}^{\oplus}$, and pairwise $\oplus$-distinct, $\oplus$-reduced standard terms $r_1, \ldots, r_l \notin \tilde{\mathsf{C}}$.

We define $\{t_1, \ldots, t_n\}$ as the set of those elements $s_i$ for which there is no $r_j$ with $s_i \sim r_j$ and, analogously, those elements $r_j$ for which there is no $s_i$ with $s_i \sim r_j$. Then, $\pi(i) \sim \mathrm{I}(s \oplus r) \sim I(c \oplus d \oplus t_1 \oplus \cdots \oplus t_n)$. By Lemma 9, we know that $\mathrm{I}(c \oplus \tilde{s}_1 \oplus \cdots \oplus \tilde{s}_m)$ and $\mathrm{I}(d \oplus \tilde{r}_1 \oplus \cdots \oplus \tilde{r}_l)$ can be derived from $T \cup \Delta(\pi_{<i})$ modulo XOR. Hence, $\mathrm{I}(t')$ with $t' = c \oplus d \oplus \tilde{t}_1 \oplus \cdots \oplus \tilde{t}_n$ can be derived from $T \cup \Delta(\pi_{<i})$ as well, by applying the $\oplus$-rule. Here we use that, for all terms $w, z$, if $w \sim z$, then $\tilde{w} \sim \tilde{z}$. Now, let us consider two cases:

(a) $n = 0$ or $n > 1$: In this case, we have that $\Delta(\pi(i)) \sim \mathrm{I}(t')$, and hence, $\Delta(\pi(i))$ can be derived from $\Delta(\pi_{<i})$.

(b) $n = 1$: Because $\mathrm{I}(c \oplus s_1 \oplus \cdots \oplus s_m)$ and $\mathrm{I}(d \oplus r_1 \oplus \cdots \oplus r_l)$ occur in $\pi_{<i}$ modulo XOR, by Lemma 8, $\mathrm{I}(\tilde{t}_1 \oplus t_1)$ occurs in $\pi_{<i}$ modulo XOR as well. Thus, by Lemma 3-(f), $\mathrm{I}(\tilde{t}_1 \oplus \Delta(t_1))$ occurs in $\Delta(\pi_{<i})$ modulo XOR. Now, because $\mathrm{I}(t')$, with $t' = c \oplus d \oplus \tilde{t}_1$, can be derived from $\Delta(\pi_{<i})$ modulo XOR, so can $\mathrm{I}(c \oplus d \oplus \Delta(t_1)) \sim \Delta(\pi(i))$. □

## 4 The Reduction

In this section, we show how the deduction problem modulo XOR can be reduced to the deduction problem without XOR for $\mathsf{C}$-dominated theories. More precisely, for a $\mathsf{C}$-dominated theory $T$, we show how to effectively construct a Horn theory $T^+$ such that a ($\mathsf{C}$-dominated) fact can be derived from $T$ modulo XOR if and only if it can be derived from $T^+$ in a syntactic derivation, where XOR is considered to be a function symbol without any algebraic properties. As mentioned, the syntactic deduction problem, and hence, the problem of checking secrecy for cryptographic protocols w.r.t. an unbounded number of sessions, can then be solved by tools, such as ProVerif, which cannot deal with the algebraic properties of XOR.

In the remainder of this section, let $T$ be a $\mathsf{C}$-dominated theory. In what follows, we will first define the reduction function, which turns $T$ into $T^+$, and state the main result (Section 4.1), namely soundness and completeness of the reduction. Before proving this result, in Section 4.3, we illustrate the reduction function by our running example (Section 4.2).

## 4.1 The Reduction Function

The reduction function uses an operator $\ulcorner\cdot\urcorner$, which turns terms into what we call a normal form, and a set $\Sigma(t)$ of substitutions associated with the term $t$. We first define this operator and the set $\Sigma(t)$. The operator $\ulcorner\cdot\urcorner$ is defined w.r.t. a linear ordering $<_{\mathsf{C}}$ on $\mathsf{C}$, which we fix once and for all.

**Definition 3** For a $\mathsf{C}$-dominated term $t$, we define the *normal form* of $t$ (w.r.t. $\mathsf{C}$), denoted by $\ulcorner t\urcorner$, recursively as follows:

- If $t$ is a variable, then $\ulcorner t\urcorner = t$.
- If $t = f(t_1,\ldots,t_n)$ is a standard term, then $\ulcorner t\urcorner = f(\ulcorner t_1\urcorner,\ldots,\ulcorner t_n\urcorner)$.
- If $t \in \mathsf{C}^{\oplus}$ is non-standard and $t \sim c_1 \oplus \cdots \oplus c_n$, for some pairwise $\oplus$-distinct $c_1,\ldots,c_n \in \mathsf{C}$, such that $c_1 <_{\mathsf{C}} \cdots <_{\mathsf{C}} c_n$, then $\ulcorner t\urcorner = c_1 \oplus (c_2 \oplus (\cdots \oplus c_n)\cdots)$ (where $\ulcorner t\urcorner = 0$ if $n = 0$).
- If $t$ is non-standard and $t \sim c \oplus t'$, for some $c \in \mathsf{C}^{\oplus}$, $c \not\sim 0$, and standard $t'$ not in $\tilde{\mathsf{C}}$, then $\ulcorner t\urcorner = \ulcorner c\urcorner \oplus \ulcorner t'\urcorner$.

We say that a term $t$ is in *normal form*, if $t = \ulcorner t\urcorner$. A substitution $\theta$ is in normal form, if $\theta(x)$ is in normal form for each variable $x$ in the domain of $\theta$.

It is easy to see that $\ulcorner t\urcorner = \ulcorner s\urcorner$ for $\mathsf{C}$-dominated terms $t$ and $s$ if and only if $t \sim s$, and that $\ulcorner t\urcorner$ is $\oplus$-reduced for any $t$. By $\mathsf{C}^{\oplus}_{\mathsf{norm}}$, we denote the set $\{\ulcorner c\urcorner \mid c \in \mathsf{C}^{\oplus}\}$. Clearly, this set is finite and computable in exponential time in the size of $\mathsf{C}$.

To define the set $\Sigma(t)$ of substitutions, we need the notion of fragile subterms. For a $\mathsf{C}$-dominated term $t$, the set of *fragile subterms of* $t$, denoted by $\mathscr{F}(t)$, is $\mathscr{F}(t) = \{s \mid s$ is a non-ground, standard term which occurs as a subterm of $t$ of the form $t' \oplus s$ or $s \oplus t'$ for some $t'\}$. For example, $\mathscr{F}((a \oplus \langle x,b\rangle) \oplus b) = \{\langle x,b\rangle\}$.

We are now ready to define the (finite and effectively computable) set $\Sigma(t)$ of substitutions for a $\mathsf{C}$-dominated term $t$. Before defining this set, let us look at its main property: For every $\mathsf{C}$-dominated, ground substitution $\theta$ in normal form, there exists a substitution $\sigma \in \Sigma(t)$ and a substitution $\theta'$ such that $\ulcorner t\theta\urcorner = (\ulcorner t\sigma\urcorner)\theta'$. In other words, the substitutions in $\Sigma(t)$ yield all relevant instances of $t$. All ground, normalized instances are syntactic instances of those instances. This resembles the finite variant property of XOR [CLD05] mentioned in the introduction. However, our construction of $\Sigma(t)$ is tailored and optimized towards $\mathsf{C}$-dominated terms and substitutions. More importantly, we obtain a stronger property in the sense that the equality—$\ulcorner t\theta\urcorner = (\ulcorner t\sigma\urcorner)\theta'$— is a *syntactic* equality, not just equality modulo AC; the notion of $\mathsf{C}$-domination, which we introduced here, is crucial in order to obtain this property. Having syntactic equality is important to get rid of algebraic properties completely, which is the goal of our reduction.

**Definition 4** Let $t$ be a $\mathsf{C}$-dominated term. We define a family of substitutions $\Sigma(t)$ as follows. The domain of every substitution in $\Sigma(t)$ is the set of all variables which occur in some $s \in \mathscr{F}(t)$. Now, we define $\sigma$ to belong to $\Sigma$, if for each $x \in \mathrm{dom}(\sigma)$ one of the following cases holds:

(i) $\sigma(x) = x$,

$$\ulcorner r_1\sigma\urcorner,\ldots,\ulcorner r_n\sigma\urcorner \to \ulcorner r_0\sigma\urcorner \quad \text{for each } \mathsf{C}\text{-dominated rule } r_1,\ldots,r_n \to r_0 \text{ of } T \text{ and each } \sigma \in \Sigma(\langle r_0,\ldots,r_n\rangle). \tag{5}$$

$$\mathrm{I}(c),\mathrm{I}(c') \to \mathrm{I}(\ulcorner c \oplus c'\urcorner) \quad \text{for each } c,c' \in \mathsf{C}^{\oplus}_{\mathsf{norm}} \tag{6}$$

$$\mathrm{I}(c),\mathrm{I}(x) \to \mathrm{I}(c \oplus x) \quad \text{for each } c \in \mathsf{C}^{\oplus}_{\mathsf{norm}} \tag{7}$$

$$\mathrm{I}(c),\mathrm{I}(c' \oplus x) \to \mathrm{I}(\ulcorner c \oplus c'\urcorner \oplus x) \quad \text{for each } c,c' \in \mathsf{C}^{\oplus}_{\mathsf{norm}} \tag{8}$$

$$\mathrm{I}(c \oplus x),\mathrm{I}(c' \oplus x) \to \mathrm{I}(\ulcorner c \oplus c'\urcorner) \quad \text{for each } c,c' \in \mathsf{C}^{\oplus}_{\mathsf{norm}} \tag{9}$$

**Fig. 2** Rules of the theory $T^+$. We use the convention that $I(0 \oplus x)$ stands for $I(x)$.

(ii) $x \in \mathscr{F}(t)$ and $\sigma(x) = c \oplus x$, for some $c \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$, $c \neq 0$,
(iii) there exists $s \in \mathscr{F}(t)$ with $x \in \mathrm{var}(s)$ and a $\mathsf{C}$-dominated substitution $\theta$ in normal form such that $s\theta \in \mathsf{C}^{\oplus}$ and $\sigma(x) = \theta(x)$.

To illustrate the definition and the property mentioned above, consider, as an example, $t = c \oplus x$ and the substitution $\theta(x) = d \oplus m$, with $d \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$ and a $\mathsf{C}$-dominated, standard term $m \notin \mathsf{C}^{\oplus}_{\mathsf{norm}}$ in normal form. We want to come up with a substitution $\sigma \in \Sigma(t)$ and a substitution $\theta'$ such that $\ulcorner t\sigma\urcorner\theta' = \ulcorner t\theta\urcorner$: We can choose $\sigma(x) = d \oplus x$, according to (ii), and $\theta'(x) = m$. We obtain $\ulcorner t\theta\urcorner = \ulcorner c \oplus d\urcorner \oplus m = (\ulcorner t\sigma\urcorner)\theta'$. If $\theta(x)$ were $d \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$, then we could take $\sigma(x) = d$, according to (iii).

We can show the following lemma (see the appendix for the proof):

**Lemma 10** *For a $\mathsf{C}$-dominated term $t$, the set $\Sigma(t)$ can be computed in exponential time in the size of $t$ and $\mathsf{C}$.*

We are now ready to define the reduction function which turns $T$ into $T^+$. The Horn theory $T^+$ is given in Fig. 2. With the results shown above, it is clear that $T^+$ can be constructed in exponential time from $T$. The Horn clauses in (6)–(9) simulate the $\oplus$-rule in case the terms we consider are $\mathsf{C}$-dominated. The other rules in $T$ are simulated by the rules in (5), which are constructed in such a way that they allow us to produce messages in normal form for input messages in normal form.

We can now state the main theorem of this paper. This theorem says that a message (a secret) can be derived from $T$ using derivations modulo XOR if and only if it can be derived from $T^+$ using only syntactic derivations, i.e., no algebraic properties of XOR are taken into account. This allows us to reduce the problem of verifying secrecy for cryptographic protocols with XOR, to the XOR-free case. The latter problem can then be handled by tools, such as ProVerif, which otherwise could not deal with XOR.

**Theorem 1** *For a $\mathsf{C}$-dominated Horn theory $T$ and a $\mathsf{C}$-dominated message $b$ in normal form, we have: $T \vdash_{\oplus} b$ if and only if $T^+ \vdash b$.*

Before we prove this theorem, we illustrate the reduction by our running example.

### 4.2 Example

Consider the Horn theory $T_{P_{NSL_{\oplus}}}$ of our running example. As mentioned in Section 3, this Horn theory is $\mathsf{C}$-dominated for $\mathsf{C} = \{a,b\}$. In what follows, we illustrate how $T^+_{P_{NSL_{\oplus}}}$ looks like, where the elements of $\mathsf{C}$ are ordered as $a <_{\mathsf{C}} b$.

First, consider the instances of Horn clauses of $T_{P_{NSL_\oplus}}$ given by (5). Only the Horn clauses in (3) have fragile subterms. All other Horn clauses have only one instance in $T^+_{P_{NSL_\oplus}}$: the clause itself. This is because for such Horn clauses $\Sigma(\cdot)$ contains only one substitution, the identity. The Horn clause in (3) has one fragile subterm, namely $x$. Hence, the domain of every substitution in the corresponding $\Sigma$-set is $\{x\}$, and according to Definition 4, this set contains the following eight substitutions: item (i) gives $\sigma_1 = \{x/x\}$; item (ii) gives $\sigma_2 = \{a \oplus x/x\}$, $\sigma_3 = \{b \oplus x/x\}$, and $\sigma_4 = \{(a \oplus b) \oplus x/x\}$; item (iii) gives $\sigma_5 = \{0/x\}$, $\sigma_6 = \{a/x\}$, $\sigma_7 = \{b/x\}$, and $\sigma_8 = \{a \oplus b/x\}$. For each of these substitutions we obtain an instance of (3). For example, $\sigma_4$ yields

$$\mathrm{I}(\{\!|\langle (a \oplus b) \oplus x, a\rangle|\!\}_{\mathsf{pub}(sk_b)}) \to \mathrm{I}(\{\!|\langle m(b,a), a \oplus x\rangle|\!\}_{\mathsf{pub}(sk_a)}).$$

Now, consider the Horn clauses induced by (6)–(9). For example, the set of Horn clauses (8) contains among others: $\mathrm{I}(a \oplus b), \mathrm{I}(b \oplus x) \to \mathrm{I}(a \oplus x)$ and $\mathrm{I}(b), \mathrm{I}(a \oplus x) \to \mathrm{I}((a \oplus b) \oplus x)$.

### 4.3 Proof of Theorem 1

In what follows, let $T$ be a $\mathsf{C}$-dominated Horn theory and $b$ be a $\mathsf{C}$-dominated message in normal form. Note that $\ulcorner b \urcorner = b$. The following lemma proves that our reduction is sound, i.e., that $T^+ \vdash b$ implies $T \vdash_\oplus b$.

**Lemma 11** *If $\pi$ is a syntactic derivation for $b$ from $T^+$, then $\pi$ is a derivation for $b$ from $T$ modulo XOR.*

*Proof* Let $\pi$ be a syntactic derivation for $b$ from $T^+$. To prove the lemma it suffices to prove that each $\pi(i)$ can be obtained by a derivation modulo XOR from $T$ and $\pi_{<i}$. If $\pi(i)$ is obtained from $\pi(j) = \mathrm{I}(t)$ and $\pi(k) = \mathrm{I}(s)$ for $j,k < i$, using one of the Horn clauses (6)–(9), then we can apply the $\oplus$-rule with $\pi(j)$ and $\pi(k)$ to obtain $\mathrm{I}(t \oplus s) \sim \pi(i)$.

Now, suppose that $\pi(i)$ is obtained using a Horn clause in (5) of the form $\ulcorner r_1\sigma\urcorner, \ldots, \ulcorner r_n\sigma\urcorner \to \ulcorner r_0\sigma\urcorner$ for some Horn clause $(r_1, \ldots, r_n \to r_0) \in T$ and some $\sigma \in \Sigma(\langle r_0, \ldots, r_n\rangle)$. Note that $\ulcorner t \urcorner \sim t$ and, if $t \sim t'$, then $t\sigma \sim t'\sigma$ for all terms $t, t'$ and substitutions $\sigma$. So, there exists a substitution $\theta$ such that $\pi(i) = \ulcorner r_0\sigma\urcorner\theta \sim (r_0\sigma)\theta = r_0(\sigma\theta)$ and, for each $k \in \{1, \ldots, n\}$, there exists $j < i$ such that $\pi(j) = \ulcorner r_k\sigma\urcorner\theta \sim (r_k\sigma)\theta = r_k(\sigma\theta)$. Therefore, we can use the rule $r_1, \ldots, r_n \to r_0$ with the substitution $\sigma\theta$ to obtain $r_0(\sigma\theta) = (r_0\sigma)\theta \sim \pi(i)$. □

To prove the completeness of our reduction, i.e., that $T \vdash_\oplus b$ implies $T^+ \vdash b$, we first prove the property of $\Sigma(t)$ mentioned before Definition 4. For this, we need the following definition.

**Definition 5** Let $t$ be a $\mathsf{C}$-dominated term and $\theta$ be a $\mathsf{C}$-dominated, ground substitution in normal form with $\mathrm{dom}(\theta) = \mathrm{var}(t)$. Let $\sigma = \sigma(t, \theta)$ be the substitution defined as follows. The domain of $\sigma$ is the set of all variables that occur in some $s \in \mathscr{F}(t)$. Let $x$ be such a variable. We define $\sigma(x)$ according to the following conditions, which have decreasing priority:

(a) If there exists $s \in \mathscr{F}(t)$ with $x \in \mathrm{var}(s)$ such that $s\theta \in \mathsf{C}^\oplus_{\mathsf{norm}}$, then $\sigma(x) = \theta(x)$.
(b) Otherwise, if $x \in \mathscr{F}(t)$ and $\theta(x) = c \oplus s'$, for a ground term $c \in \mathsf{C}^\oplus$ and some standard term $s'$ not in $\tilde{\mathsf{C}}$, then $\sigma(x) = c \oplus x$. (Note that $c \neq 0$ since $\theta(x)$ is in normal form.)
(c) Otherwise, $\sigma(x) = x$. (Note that in this case we know that $\theta(x)$ is some standard term not in $\tilde{\mathsf{C}}$ if $x \in \mathscr{F}(t)$.)

Equipped with this definition, we can show (see the appendix) the property of $\Sigma(t)$ mentioned before Definition 4.

**Lemma 12** *Let $t$ be a $\mathsf{C}$-dominated term and $\theta$ be a $\mathsf{C}$-dominated, ground substitution in normal form with $\mathrm{dom}(\theta) = \mathrm{var}(t)$. Then, $\sigma = \sigma(t,\theta) \in \Sigma(t)$ and there exists a substitution $\theta'$ such that $\theta = \sigma\theta'$, i.e., $\theta(x) = \sigma(x)\theta'$ for every $x \in \mathrm{dom}(\theta)$, and $\ulcorner t'\theta \urcorner = \ulcorner t'\sigma \urcorner\theta'$ for every subterm $t'$ of $t$.*

We can now show the completeness of our reduction.

**Lemma 13** *If $\pi$ is a $\mathsf{C}$-dominated derivation for $b$ from $T$ modulo XOR obtained using $\mathsf{C}$-dominated substitutions, then $\ulcorner \pi \urcorner$ is a syntactic derivation for $b$ from $T^+$.*

*Proof* We show that every $\ulcorner \pi(i) \urcorner$ can be derived syntactically from $T^+$ and $\ulcorner \pi_{<i} \urcorner$. Two cases are distinguished:

**Case 1:** $\pi(i)$ is obtained from $\pi(j) = \mathrm{I}(t)$ and $\pi(k) = \mathrm{I}(s)$, for $j,k < i$, using the $\oplus$-rule. In that case $\pi(i) = \mathrm{I}(r)$ with $r \sim t \oplus s$. By assumption, $t$, $s$ and $r$ are $\mathsf{C}$-dominated. Hence, $\ulcorner t \urcorner$ and $\ulcorner s \urcorner$ are either normalized standard terms not in $\mathsf{C}^{\oplus}$, terms in $\mathsf{C}^{\oplus}_{\mathsf{norm}}$, or terms of the form $c \oplus u$ for $c \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$ and a normalized standard term $u \notin \mathsf{C}^{\oplus}$. However, it is not the case that $\ulcorner t \urcorner = u$ or $\ulcorner t \urcorner = c \oplus u$ and $\ulcorner s \urcorner = u'$ or $\ulcorner s \urcorner = c' \oplus u'$ with $u, u' \notin \mathsf{C}^{\oplus}$ and $u \neq u'$ since otherwise $r$ would not be $\mathsf{C}$-dominated. Now, it is easy to see that $\oplus$-rule can be simulated by one of the Horn clauses (6)–(9).

**Case 2:** $\pi(i)$ is obtained using some $\mathsf{C}$-dominated rule $(r_1, \ldots, r_n \to r_0) \in T$ and a ground substitution $\theta$. Since $\pi$ is a derivation modulo XOR, we may assume that $\theta$ is in normal form. We have that $\pi(i) \sim r_0\theta$ and there exist $j_1, \ldots, j_n < i$ such that $\pi(j_k) \sim r_k\theta$, for all $k \in \{1, \ldots, n\}$.

Let $\sigma = \sigma(\langle r_0, \ldots, r_n \rangle, \theta)$ and let $\theta'$ be as specified in Lemma 12. By Lemma 12, $\sigma \in \Sigma(\langle r_0, \ldots, r_n \rangle)$. Now, to obtain $\ulcorner \pi(i) \urcorner$, we can use the rule $\rho = (\ulcorner r_1\sigma \urcorner, \ldots, \ulcorner r_n\sigma \urcorner \to \ulcorner r_0\sigma \urcorner) \in T^+$ with the substitution $\theta'$. In fact, by Lemma 12, we have that $\ulcorner r_k\sigma \urcorner\theta' = \ulcorner r_k\theta \urcorner = \ulcorner \pi(j_k) \urcorner$ for all $k \in \{0, \ldots, n\}$, where $j_0 = i$. (Recall that for $\mathsf{C}$-dominated terms $s$ and $t$ with $s \sim t$, we have that $\ulcorner s \urcorner = \ulcorner t \urcorner$.) $\square$

Now, from the above lemma and Proposition 1 it immediately follows that $T \vdash_{\oplus} b$ implies $T^+ \vdash b$.

## 5 Implementation and Experimental Results

We have implemented our reduction, and together with ProVerif, tested it on a set of protocols which employ the XOR operator (see [KT08b] for the implementation). In this section, we report on our implementation and the experimental results.

### 5.1 Implementation

We have implemented our reduction function in SWI Prolog (version 5.6.14). Our implementation essentially takes a Horn theory as input. More precisely, the input consists of (1) a declaration of all the functor symbols used in the protocol and by the intruder, (2) the initial intruder facts as well as the protocol and intruder rules, except for the $\oplus$-rule, which

| protocol | correct | reduction time | ProVerif time |
|---|---|---|---|
| $\mathsf{NSL}_{\oplus}$ | no | 0.02s | 0.006s |
| $\mathsf{NSL}_{\oplus}$-fix | yes | 0.04s | 0.09s |
| SK3 | yes | 0.05s | 0.3s |
| RA | no | 0.05s | 0.17s |
| RA-fix | yes | 0.05s | 0.27s |
| CCA-0 | no | 0.15s | 109s |
| CCA-1A | yes | 0.06s | 0.7s |
| CCA-1B | yes | 0.07s | 1.3s |
| CCA-2B | yes | 0.14s | 7.1s |
| CCA-2C | yes | 0.15s | 58.0s |
| CCA-2E | yes | 0.07s | 1.42s |

**Fig. 3** Experimental Results.

is assumed implicitly, (3) a statement which defines a secrecy goal. Moreover, options that are handed over to ProVerif may be added.

Our implementation then first checks whether the given Horn theory, say $T$, (part (2) of the input) is $\oplus$-linear. If it is not, an error message is returned. If it is, a set $\mathsf{C}$ (of minimal size) is computed such that the Horn theory is $\mathsf{C}$-dominated. Recall that such a set always exists if the Horn theory is $\oplus$-linear. It is important to keep $\mathsf{C}$ as small as possible in order for the reduction to be more efficient. Once $\mathsf{C}$ is computed, the reduction function as described in Section 4, with some optimizations tailored towards ProVerif (see below), is applied to $T$, i.e., $T^+$ is computed. Now, $T^+$ together with the rest of the original input is passed on to ProVerif. This tool then does the rest of the work, i.e., it checks the goals for $T^+$. This is possible since, due the reduction, the XOR operator in $T^+$ can now be considered to be an operator without any algebraic properties.

Our implementation does not follow the construction of the reduction function described in Section 4 precisely in order to produce an output that is optimized for ProVerif (but still equivalent): a) While terms of the form $c \oplus t$, with $c \in \mathsf{C}^{\oplus}$, $t \notin \mathsf{C}^{\oplus}$ are represented by $\mathtt{xor}(\mathtt{c},\mathtt{t})$, terms $a \oplus b \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$ are represented by $\mathtt{xx}(\mathtt{a},\mathtt{b})$. This representation prevents some unnecessary unifications between terms. However, it is easy to see that with this representation, the proofs of soundness and completeness of our reduction still go through. The basic reason is that terms in $\mathsf{C}^{\oplus}_{\mathsf{norm}}$ can be seen as constants. b) For the Horn clauses (6)–(9) in Figure 2, we do not produce copies for every choice of $c, c' \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$. Instead, we use a more compact representation by introducing auxiliary predicate symbols. For example, the family of Horn clauses in (8) is represented as follows: $\mathtt{xtab}(\mathtt{x},\mathtt{y},\mathtt{z}), \mathtt{I}(\mathtt{y}), \mathtt{I}(\mathtt{xor}(\mathtt{x},\mathtt{t})) \rightarrow \mathtt{I}(\mathtt{xor}(\mathtt{z},\mathtt{t}))$, where the facts $\mathtt{xtab}(\mathtt{c},\mathtt{c}',\ulcorner \mathtt{c} \oplus \mathtt{c}' \urcorner)$ for every $c, c' \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$ are added to the Horn theory given to ProVerif.

## 5.2 Experiments

We applied our method to a set of ($\oplus$-linear) protocols. The results, obtained by running our implementation on a 2,4 Ghz Intel CoreTM 2 Duo E6700 processor with 2GB RAM, are depicted in Figure 3, where we list both the time of the reduction and the time ProVerif needed for the analysis of the output of the reduction. We note that except for certain versions of the CCA protocol, the other protocols listed in Figure 3 are out of the scope of the implementation in [CKS07], the only other implementation that we know of for cryptographic protocol

$$I(x),\ I(\{k\}_{\mathrm{KM}\oplus\mathrm{DATA}}) \rightarrow I(\{x\}_k) \qquad (\textit{Encipher})$$

$$I(\{x\}_k),\ I(\{k\}_{\mathrm{KM}\oplus\mathrm{DATA}}) \rightarrow I(x) \qquad (\textit{Decipher})$$

$$I(\{k\}_{\mathrm{KM}\oplus type}),\ I(type),\ I(\{kek\}_{\mathrm{KM}\oplus\mathrm{EXP}}) \rightarrow I(\{k\}_{kek\oplus type}) \qquad (\textit{KeyExport})$$

$$I(\{k\}_{kek\oplus type}),\ I(type),\ I(\{kek\}_{\mathrm{KM}\oplus\mathrm{IMP}}) \rightarrow I(\{k\}_{\mathrm{KM}\oplus type}) \qquad (\textit{KeyImport})$$

$$I(k1),\ I(type) \rightarrow I(\{k1\}_{\mathrm{KM}\oplus\mathrm{KP}\oplus type}) \qquad (\textit{KeyPartImp-First})$$

$$I(k2),\ I(\{x\}_{\mathrm{KM}\oplus\mathrm{KP}\oplus type}),\ I(type) \rightarrow I(\{x\oplus k2\}_{\mathrm{KM}\oplus\mathrm{KP}\oplus type}) \qquad (\textit{KeyPartImp-Middle})$$

$$I(k3),\ I(\{y\}_{\mathrm{KM}\oplus\mathrm{KP}\oplus type}),\ I(type) \rightarrow I(\{y\oplus k3\}_{\mathrm{KM}\oplus type}) \qquad (\textit{KeyPartImp-Last})$$

$$I(\{k\}_{kek_1\oplus type}),\ I(type),\ I(\{kek_1\}_{\mathrm{KM}\oplus\mathrm{IMP}}),\ I(\{kek_2\}_{\mathrm{KM}\oplus\mathrm{EXP}}) \rightarrow I(\{k\}_{kek_2\oplus type}) \qquad (\textit{KeyTranslate})$$

**Fig. 4** CCA API , where KM denotes a constant (the key master stored in the cryptographic coprocessor), *type* is a constant that ranges over the constants in $\{\mathrm{DATA}, \mathrm{IMP}, \mathrm{EXP}, \mathrm{PIN}\}$, and all other symbols $x$, $y$, $k$, ... are variables.

analysis w.r.t. an unbounded number of sessions that takes XOR into account. As mentioned in the introduction, the method in [CKS07] is especially tailored to the CCA protocol. It can only deal with symmetric encryption and the XOR operator, but, for example, cannot deal with protocols that use public-key encryption or pairing. Let us discuss the protocols and settings that we analyzed in more detail.

By $\mathsf{NSL}_{\oplus}$ we denote our running example. Since there is an attack on this protocol, we also propose a fix $\mathsf{NSL}_{\oplus}$-fix in which the message $\{\!|\langle M, N\oplus B\rangle|\!\}_{\mathsf{pub}(sk_A)}$ is replaced by $\{\!|\langle M, h(\langle N, M\rangle)\oplus B\rangle|\!\}_{\mathsf{pub}(sk_A)}$ for a hash function $h(\cdot)$.

The ($\oplus$-linear) protocol SK3 [SR96] is a key distribution protocol for smart cards, which uses the XOR operator. RA denotes an ($\oplus$-linear) group protocol for key distribution [BO97]. Since there is a known attack on this protocol, we proposed a fix: a message $k_{A,B}\oplus h(\langle \mathsf{key}(A), N\rangle)$ sent by the key distribution server to $A$ is replaced by $k_{A,B}\oplus h(\langle \mathsf{key}(A), \langle N, B\rangle\rangle)$.

CCA stands for Common Cryptographic Architecture (CCA) API [IBM03] as implemented on the hardware security module IBM 4758 (an IBM cryptographic coprocessor). The CCA API is used in ATMs and mainframe computers of many banks to carry out PIN verification requests. It accepts a set of commands, which can be seen as receive-send-actions, and hence, as cryptographic protocols. The only key stored in the security module is the master key KM. All other keys are kept outside of the module in the form $\{k\}_{\mathrm{KM}\oplus type}$, where $type \in \{\mathrm{DATA}, \mathrm{IMP}, \mathrm{EXP}, \mathrm{PIN}\}$ denotes the type of the key, modeled as a constant.

In Figure 4, we model the most important commands of the CCA API (see also [CKS07]) in terms of Horn clauses. (*Encipher*) and (*Decipher*) are used to encrypt/decrypt data by data keys. (*KeyExport*) is used to export a key to another security module by encrypting it under a key-encryption-key, with (*KeyImport*) being the corresponding import command. The problem is to make the same key-encryption-key available in different security modules. This is done by a secret sharing scheme using the commands (*KeyPartImp-First*)–(*KeyPartImp-Last*), where KP is a type (a constant) which stands for "key part", *kek* is obtained as $k1\oplus k2\oplus k3$, and each $ki$, $i\in\{1,2,3,\}$, is supposed to be known by only one individual. (*KeyTranslate*) is used to encrypt a key under a different key-encryption-key.

We note that two of the Horn clauses in Figure 4, namely (*KeyPartImp-Middle*) and (*KeyPartImp-Last*), are not $\oplus$-linear. Fortunately, we can deal with these rules: For the scenarios of interest, which are motivated by certain access control regulations, only a subset of the rules in Figure 4 and/or certain instances of these rules, plus some initial intruder knowledge, need to be taken into account. For each such scenario it is possible to obtain an *equivalent* Horn theory with only $\oplus$-linear rules by a standard unfolding technique together with straightforward simplifications. To illustrate the idea, assume that a scenario contains only the clauses (*KeyPartImp-First*) and (*KeyPartImp-Last*), plus some other (non-critical) clauses. In such a scenario the only way to resolve (*KeyPartImp-Last*) might be with (*KeyPartImp-First*) on the atom $I(\{y\}_{\mathrm{KM}\oplus\mathrm{KP}\oplus type})$, resulting in the clause

$$I(k3),\ I(k1),\ I(type),\ I(type) \rightarrow I(\{k1 \oplus k3\}_{\mathrm{KM}\oplus type}).$$

This clause can be simplified to the equivalent, $\oplus$-linear clause

$$I(x),\ I(type) \rightarrow I(\{x\}_{\mathrm{KM}\oplus type}).$$

which we add to our Horn theory. Now, the non $\oplus$-linear clause (*KeyPartImp-Last*) can be removed resulting in an equivalent $\oplus$-linear Horn theory. For more complex scenarios, these kinds of resolutions and simplifications are applied exhaustively (which is possible by hand) and then the non $\oplus$-linear clauses can be removed.

There are several known attacks on the CCA API, which concern the key-part-import process. One attack is by Bond [Bon01]. As a result of this attack the intruder is able to obtain PINs for each account number by performing data encryption on the security module. A stronger attack was found by IBM and is presented in [Clu03] where the intruder can obtain a PIN derivation key, and hence, can obtain PINs even without interacting with the security module. However, the IBM attack depends on key conjuring [CKS07], and hence, is harder to carry out. Using our implementation (together with ProVerif) and the configuration denoted by CCA-0 in Figure 3, we found a new attack which achieves the same as the IBM attack, but is more efficient as it does not depend on key conjuring (see Section 5.3 for details).

In response to the attacks reported in [Bon01], IBM proposed two recommendations described below.

**Recommendation 1.** As mentioned, the attacks exploit problems in the key-part-import process. To prevent these problems, one IBM recommendation is to replace this part by a public-key setting. However, as shown in [CKS07], further access control mechanisms are needed, which essentially restrict the kind of commands certain roles may perform. Two cases, which correspond to two different roles, are considered and are denoted CCA-1A and CCA-1B in Figure 3. We note that the Horn theories that correspond to these cases are $\oplus$-linear, and hence, our tool can be applied directly, no changes are necessary (not even the transformations mentioned above). Since public-key encryption (and pairing) cannot be directly handled by the tool presented by Cortier et al. [CKS07], Cortier et al. had to modify the protocol in an ad hoc way, which is not guaranteed to yield an equivalent protocol. This is also why the runtimes of the tools cannot be compared directly.

**Recommendation 2.** Here additional access control mechanisms are assumed which ensure that no single role is able to mount an attack. We analyzed exactly the same subsets of commands as the ones in [CKS07]. These cases are denoted CCA-2B, -2C, and -2E in Figure 3, following the notation in [CKS07] . The runtimes obtained in [CKS07] are comparable to ours: 333s for CCA-2B, 58s for -2C, and 0.03s for -2E.

### 5.3 Our Attack on CCA

As we noted before, our tool found an attack on the CCA API which—according to our knowledge—has not been discovered before. This attack uses the same assumptions as Bond's attack in terms of the role played by the intruder and his knowledge. As in the IBM attack, we use the fact that 0 is the default value for the constant DATA.

Our attack does not use key conjuring and, hence, is easier to carry out than the IBM attack. As a result of the attack, the intruder obtains a PIN derivation key in clear, like in the IBM attack, and hence, can compute PINs from bank account numbers without interacting with the security module.

In the attack we assume that a new key-encryption-key *kek* needs to be imported, using the three-part key import commands (*KeyPartImp-First*)–(*KeyPartImp-Last*), which means that $kek = k1 \oplus k2 \oplus k3$, where $k1$, $k2$, $k3$ are the shares known to three different individuals. The key *kek* is then used to import a new PIN-derivation key *pdk* to the security module, in the form

$$\{pdk\}_{kek \oplus \mathrm{PIN}}. \tag{10}$$

We assume that this message can be seen by the attacker and that the attacker is the third participant of the process of importing *kek*, which means that he can perform (*KeyPartImp-Last*), knows the value $k3$, and obtains the message

$$\{k1 \oplus k2\}_{\mathrm{KM} \oplus \mathrm{KP} \oplus \mathrm{IMP}}. \tag{11}$$

Now we describe the steps of the attack. After the intruder receives (11), he uses (*KeyPartImp-Last*) with $k3 \oplus \mathrm{PIN}$ instead of $k3$. In this way he obtains

$$\{kek \oplus \mathrm{PIN}\}_{\mathrm{KM} \oplus \mathrm{IMP}}. \tag{A1}$$

The intruder uses the same command again, this time with $k3 \oplus \mathrm{PIN} \oplus \mathrm{EXP}$, obtaining:

$$\{kek \oplus \mathrm{PIN} \oplus \mathrm{EXP}\}_{\mathrm{KM} \oplus \mathrm{IMP}}. \tag{A2}$$

Next, when *pdk* is imported, the intruder uses (*KeyImport*) twice: The first time with input (A1), (10), and $type = \mathrm{DATA} = 0$, which results in the message

$$\{pdk\}_{\mathrm{KM} \oplus \mathrm{DATA}}. \tag{A3}$$

The second time the command (*KeyImport*) is used with input (A2), (10), and $type = \mathrm{EXP}$, which gives the message

$$\{pdk\}_{\mathrm{KM} \oplus \mathrm{EXP}}. \tag{A4}$$

Now, using (*KeyExport*) with input (A3), (A4), and $type = \mathrm{DATA} = 0$, the attacker obtains

$$\{pdk\}_{pdk \oplus \mathrm{DATA}} = \{pdk\}_{pdk}. \tag{A5}$$

Finally, using (*Decipher*) with input (A5) and (A3), the attacker obtains the clear value of *pdk*, which can be then used to obtain the PIN for any account number: Given an account number, the corresponding PIN is derived by encrypting the account number under *pdk*.

## 6 Conclusion and Future Work

In this paper, we showed how to reduce the derivation problem for an expressive class of Horn theories with XOR, namely $\oplus$-linear Horn theories, to a purely syntactic derivation problem, where the algebraic properties of XOR can be ignored. Using this reduction, protocol analysis for protocols that use the XOR operator can be reduced to a simpler problem where the algebraic properties of XOR can be ignored. In particular, this allowed us to apply ProVerif, which cannot deal with XOR, for the analysis of protocols that use XOR. Our experimental results demonstrated that our approach can be applied in practice. Altogether, in this paper we presented the first practical method for the automatic analysis of protocols that use the XOR operator where the analysis is w.r.t. an unbounded number of protocol sessions.

We note that the general approach presented in this paper—reducing the derivation problem to a purely syntactic derivation problem and then applying tools to solve the syntactic derivation problem—has already been successfully adapted in [KT09] to deal with another important operator, namely Diffie-Hellman-Exponentiation. However, the reduction proposed in [KT09] is very different to the one presented here.

So far, the efficiency of our reduction very much depends on the number of elements in the set $\mathsf{C}$. It would be desirable to obtain a reduction where the size of $\mathsf{C}$ is a less critical factor.

Another natural direction for future work is the following. ProVerif can deal with two kinds of protocol specifications: (i) specifications expressed as Horn theories and (ii) specifications expressed in process calculus (which are then automatically translated into Horn theories by ProVerif). While so far we only make use of the first specification method, it would be desirable to also support the second.

In this work, we concentrated on secrecy properties. These properties can, for the following reason, be handled especially well in our setting: The arguments of XOR in protocols specified as Horn theories are often nonces, which can be represented as terms of the form $n(a,b)$, where $a$ and $b$ are participant names (see, e.g., Section 2.3). As mentioned in Section 2.3, in [CLC04] it was shown that to analyze secrecy properties it suffices to consider a single honest participant and a single dishonest participant. Hence, the number of nonces of the form $n(a,b)$ can be bounded by a (small) constant, which often yields $\oplus$-linear Horn theories. We note that in translations from processes to Horn theories as done by ProVerif nonces are represented as functions with variable parameters, such as variables for session identifiers. These typically yield non-$\oplus$-linear Horn theories. However, for secrecy properties, the translations can be simplified, as just explained, leading to $\oplus$-linear Horn theories.

It would be interesting to extend our approach to other security properties, such a authentication properties and observational equivalence. Unfortunately, for these properties the translation from processes to Horn theories as done by ProVerif cannot be simplified as easily as in the case of secrecy properties. While in [KT08a] we obtained preliminary results for (weak) authentication properties, these results do not solve the above problem.

## References

[BAF08] Bruno Blanchet, Martín Abadi, and Cédric Fournet. Automated verification of selected equivalences for security protocols. *Journal of Logic and Algebraic Programming*, 75(1):3–51, 2008.

[Bla01] B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW-14)*, pages 82–96. IEEE Computer Society, 2001.

[BO97] J.A. Bull and D.J. Otway. The authentication protocol. Technical Report DRA/CIS3/PROJ/CORBA/SC/1/CSM/436-04/03, Defence Research Agency, Malvern, UK, 1997.

[Bon01] Mike Bond. Attacks on cryptoprocessor transaction sets. In *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop*, volume 2162 of *Lecture Notes in Computer Science*, pages 220–234. Springer, 2001.

[CDS07] V. Cortier, S. Delaune, and G. Steel. A formal theory of key conjuring. In *20th IEEE Computer Security Foundations Symposium (CSF'07)*, pages 79–93. IEEE Comp. Soc. Press, 2007.

[CKRT03] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR. In *Proceedings of the Eighteenth Annual IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 261–270. IEEE Computer Society Press, 2003.

[CKS07] V. Cortier, G. Keighren, and G. Steel. Automatic Analysis of the Security of XOR-Based Key Management Schemes. In *Proceedings of the 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2007)*, volume 4424 of *Lecture Notes in Computer Science*, pages 538–552. Springer, 2007.

[CLC03] H. Comon-Lundh and V. Cortier. New Decidability Results for Fragments of First-order Logic and Application to Cryptographic Protocols. In *Proceedings of the 14th Internatioinal Conference on Rewriting Techniques and Applications (RTA 2003)*, volume 2706 of *Lecture Notes in Computer Science*, pages 148–164. Springer, 2003.

[CLC04] Hubert Comon-Lundh and Véronique Cortier. Security properties: two agents are sufficient. *Sci. Comput. Program.*, 50(1-3):51–71, 2004.

[CLD05] Hubert Comon-Lundh and Stéphanie Delaune. The Finite Variant Property: How to Get Rid of Some Algebraic Properties. In *Term Rewriting and Applications, 16th International Conference, RTA 2005*, volume 3467 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2005.

[CLS03] H. Comon-Lundh and V. Shmatikov. Intruder Deductions, Constraint Solving and Insecurity Decision in Presence of Exclusive OR. In *Proceedings of the Eighteenth Annual IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 271–280. IEEE, Computer Society Press, 2003.

[Clu03] J. Clulow. The design and analysis of cryptographic APIs for security devices, 2003. Master's thesis, University of Natal, Durban.

[IBM03] CCA Basic Services Reference and Guide: CCA Basic Services Reference and Guide, 2003. Available at http://www-03.ibm.com/security/cryptocards/pdfs/bs327.pdf.

[KT07] R. Küsters and T. Truderung. On the Automatic Analysis of Recursive Security Protocols with XOR. In W. Thomas and P. Weil, editors, *Proceedings of the 24th Symposium on Theoretical Aspects of Computer Science (STACS 2007)*, volume 4393 of *Lecture Notes in Computer Science*, pages 646–657. Springer, 2007.

[KT08a] R. Küsters and T. Truderung. Reducing Protocol Analysis with XOR to the XOR-free Case in the Horn Theory Based Approach. In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008)*, pages 129 –138. ACM Press, 2008.

[KT08b] Ralf Küsters and Tomasz Truderung. Reducing Protocol Analysis with XOR to the XOR-free Case in the Horn Theory Based Approach. Implementation, 2008. Available at http://infsec.uni-trier.de/software/KuestersTruderung-XORPROVERIF-2008.zip.

[KT09] Ralf Küsters and Tomasz Truderung. Using ProVerif to Analyze Protocols with Diffie-Hellman Exponentiation. In *Proceedings of the 22th IEEE Computer Security Foundations Symposium (CSF 2009)*, pages 157–171. IEEE Computer Society, 2009.

[SR96] V. Shoup and A. Rubin. Session Key Distribution Using Smart Cards. In *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques*, volume 1070 of *Lecture Notes in Computer Science*, pages 321–331. Springer, 1996.

[Ste05] Graham Steel. Deduction with XOR Constraints in Security API Modelling. In *Proceedings of the 20th International Conference on Automated Deduction (CADE 2005)*, volume 3632 of *Lecture Notes in Computer Science*, pages 322–336. Springer, 2005.

[SV09] Helmut Seidl and Kumar Neeraj Verma. Flat and One-Variable Clauses for Single Blind Copying Protocols: the XOR Case. In Ralf Treinen, editor, *Proceedings of the 20th International Conference on Rewriting Techniques and Applications (RTA 2009)*, volume 5595 of *Lecture Notes in Computer Science*, pages 118–132. Springer, 2009.

[VSS05] K.N. Verma, H. Seidl, and T. Schwentick. On the Complexity of Equational Horn Clauses. In *Proceedings of the 20th International Conference on Automated Deduction (CADE 2005)*, volume 3328 of *Lecture Notes in Computer Science*, pages 337–352. Springer, 2005.

## A Proofs for Section 3

Throughout this section, we use the following relationship between terms: $t \mathrel{\widehat{=}}_{AC} t'$ iff $t$ and $t'$ coincide modulo AC, but where standard subterms coincide syntactically. For example, $(a \oplus \langle a \oplus b, b\rangle) \oplus b \mathrel{\widehat{=}}_{AC} (a \oplus b) \oplus \langle a \oplus b, b\rangle \not\mathrel{\widehat{=}}_{AC} (a \oplus b) \oplus \langle b \oplus a, b\rangle$.

Proof of Lemma 3

*Proof* Statement (a) is a direct consequence of the fact that the term $\varphi_{\oplus}[\tilde{t}_1, \ldots, \tilde{t}_n]$ in the definition of $\Delta$ is good.

For statement (b) first observe that if $t$ is $\oplus$-reduced, then so is $\Delta(t)$. (Recall that $\tilde{t}_i$ is $\oplus$-reduced, by the definition of a type). Now, together with (a) and Lemma 2, this implies (b).

The first part of (c), follows immediately from Lemma 2. For the other statements note that an $\oplus$-reduced element $c$ of $\mathsf{C}^{\oplus}$ is of the form $c_1 \oplus \ldots \oplus c_n$ with $\oplus$-free terms $c_1, \ldots, c_n \in \mathsf{C}$. Hence, $c$ is $\mathsf{C}$-dominated. The same argument applies to $\tilde{s}$, as $\tilde{s}$ is an $\oplus$-reduced element of $\mathsf{C}^{\oplus}$.

The observation used to show (d) is that, for $c \in \mathsf{C}^{\oplus}$, the term $c \oplus t$ is bad if and only if $t$ is bad.

To prove (e), we proceed by structural induction on $s$ and consider the following cases:

- $s = x$ *is a variable*: Then, clearly, $\Delta(s\theta) = s\Delta(\theta)$.
- $s$ *is a standard term*: Then one can easily complete the proof using the induction hypothesis.
- $s \mathrel{\widehat{=}}_{AC} c \oplus s'$*, for some* $c \in \mathsf{C}^{\oplus}$ *and some standard* $s' \notin \tilde{\mathsf{C}}$:
  Then, $\Delta(s\theta) = \Delta(c \oplus s'\theta)$ which, by Lemma 3-(d), is equal to $\Delta(c) \oplus \Delta(s'\theta) = c \oplus \Delta(s'\theta)$, where for the last equality we use Lemma 3-(c) and the fact that $c$, as a subterm of $s$, is $\mathsf{C}$-dominated. Hence, by the induction hypothesis, we have $\Delta(s\theta) = c \oplus s'\Delta(\theta) = (c \oplus s')\Delta(\theta)$.

For the proof of (f), we will use the following definition: Let, for a standard term $r$, $\delta(r) = \tilde{r}$, if $r \notin \tilde{\mathsf{C}}$, and $\delta(r) = \Delta(t)$, otherwise. For a non-standard term $r = r_1 \oplus \cdots \oplus r_k$ with standard $r_1, \ldots, r_k$, let $\delta(r) = \delta(r_1) \oplus \ldots \oplus \delta(r_n)$.

Suppose that $s \sim t$. We proceed by induction on the size of $s$ and $t$. If $s$ and $t$ are both good, then the proof can be easily completed using the induction hypothesis. Otherwise, it must be true that both $t$ and $s$ are bad terms. So, let us assume that this is the case. We consider two sub-cases:

(a) $s \sim t \sim 0$. Then $s \mathrel{\widehat{=}}_{AC} c_1 \oplus d_1 \oplus \cdots \oplus c_m \oplus d_m \oplus s_1 \oplus r_1 \oplus \ldots \oplus s_n \oplus r_n$ with $c_i \sim d_i$ and $s_i \sim r_i$, where $c_i, d_i$ are standard terms in $\tilde{\mathsf{C}}$ and $s_i, r_i$ are standard terms not in $\tilde{\mathsf{C}}$. We have

$$\Delta(s) \mathrel{\widehat{=}}_{AC} \Delta(c_1) \oplus \Delta(d_1) \oplus \cdots \oplus \Delta(c_m) \oplus \Delta(d_m) \oplus \tilde{s}_1 \oplus \tilde{r}_1 \oplus \ldots \oplus \tilde{s}_n \oplus \tilde{r}_n.$$

For each $i \in \{1, \ldots, m\}$, by the induction hypothesis, we have $\Delta(c_i) \sim \Delta(d_i)$. By the definition of types, we also have $\tilde{s}_i \sim \tilde{r}_i$, for each $i \in \{1, \ldots, n\}$. Therefore, $\Delta(s) \sim 0$. Analogously, we obtain $\Delta(t) \sim 0$, and hence, $\Delta(s) \sim \Delta(t)$.

(b) $s \sim t \not\sim 0$. Then we can represent $s$ as $s \mathrel{\widehat{=}}_{AC} c_1 \oplus \ldots \oplus c_m \oplus s_1 \oplus \ldots \oplus s_n \oplus s'$ where $c_1, \ldots, c_m$ are pairwise $\oplus$-distinct standard terms in $\tilde{\mathsf{C}}$, $s_1, \ldots, s_n$ are pairwise $\oplus$-distinct

standard terms not in $\tilde{\mathsf{C}}$, and $s' \sim 0$. Analogously, we can represent $t$ in the form $t \mathrel{\widehat{=}}_{AC} d_1 \oplus \ldots \oplus d_m \oplus t_1 \oplus \ldots \oplus t_n \oplus t'$ with $c_i \sim d_i$ and $s_i \sim t_i$. We have

$$\Delta(s) \;=\; \delta(s) \;=\; \Delta(c_1) \oplus \ldots \oplus \Delta(c_m) \oplus \tilde{s_1} \oplus \ldots \oplus \tilde{s_n} \oplus \delta(s')$$

and, similarly

$$\Delta(t) \;=\; \delta(t) \;=\; \Delta(d_1) \oplus \ldots \oplus \Delta(d_m) \oplus \tilde{t_1} \oplus \ldots \oplus \tilde{t_n} \oplus \delta(t').$$

Similarly to the case (a), we obtain $\delta(s') \sim 0$ and $\delta(t') \sim 0$. Moreover, by the induction hypothesis $\Delta(c_i) \sim \Delta(d_i)$. By the definition of types, we have $\tilde{s_i} \sim \tilde{t_i}$. It follows that $\Delta(s) \sim \Delta(t)$.

Proof of Lemma 5.

Assume that $r'$ is a complete bad subterm of $r\theta$. We proceed by structural induction on $r$ and consider the following cases:

- $r = x$ *is a variable*: Because $\theta$ is $\oplus$-reduced, so is $\theta(x)$. So, since $r'$ is a subterm of $\theta(x)$ and $\theta(x) \sim t$, Lemma 4 implies that there exists a complete bad subterm $t'$ of $t$ with $t' \sim r'$.
- $r = f(r_1,\ldots,r_n)$, *for* $f \neq \oplus$*:* In this case, $t \mathrel{\widehat{=}}_{AC} t' \oplus f(t_1,\ldots,t_n)$ with $t_i \sim r_i\theta$, for every $i \in \{1,\ldots,n\}$, and $t' \sim 0$. Note that, by the definition of $\mathrel{\widehat{=}}_{AC}$, the term $f(t_1,\ldots,t_n)$ is a subterm of $t$. Since $r\theta$ is good, $r'$ is a subterm of $r_i\theta$ for some $i \in \{1,\ldots,n\}$. By the induction hypothesis, there exists a complete bad subterm $t'$ of $t_i$ (and thus, of $t$) with $t' \sim r'$.
- $r = c$, *for* $c \in \mathsf{C}^{\oplus}$*:* We have that $r\theta = r$. Since $r$ is $\mathsf{C}$-dominated it follows that $c$ does not contain bad subterms. Hence, nothing is to show.
- $r \mathrel{\widehat{=}}_{AC} c \oplus r''$ with $c \in \mathsf{C}^{\oplus}$ and $r'' \notin \tilde{\mathsf{C}}$ standard, but not a variable: The case $r' = r\theta$ cannot occur since this term is not a bad term. Since $r$ is $\mathsf{C}$-dominated, $c$ does not contain a bad subterm. Therefore, $r'$ must be a subterm of $r''\theta$.
  Let $s \sim r''\theta$, for some $\oplus$-reduced term $s$. So, we have $t \sim c \oplus s$. Since $r''$ is a proper subterm of $r$, it is $\mathsf{C}$-dominated. Hence, from the fact that $r'$ is a complete bad subterm of $r''\theta$ it follows by the induction hypothesis that there exists a complete bad subterm $t'$ of $s$ with $r' \sim t'$. The equivalence $t \sim c \oplus s$ implies $c \oplus t \sim s$. By Lemma 4, it follows that there exists a bad subterm $t''$ of $c \oplus t$ with $t'' \sim t'$. We may assume that $c$ is of the form $c_1 \oplus \ldots \oplus c_k$ for standard terms $c_i$, $i \in \{1,\ldots,k\}$. By definition of $\mathrel{\widehat{=}}_{AC}$, every $c_i$ is a subterm of $r$, and hence, $\mathsf{C}$-dominated. It follows that $c$ does not contain a bad subterm. Thus, $t''$ is a subterm of $t$. Since $t'' \sim t' \sim r'$, we are done.
- $r \mathrel{\widehat{=}}_{AC} c \oplus x$, for $c \in \mathsf{C}^{\oplus}$ and a variable $x$: We may assume as in the previous case that $c$ is of the form $c_1 \oplus \cdots \oplus c_k$, where $c_1,\ldots,c_k$ are standard and $\mathsf{C}$-dominated. In particular, $c$ does not contain a bad subterm. Assume also that $\theta(x) \mathrel{\widehat{=}}_{AC} c'_1 \oplus \cdots \oplus c'_m \oplus t_1 \oplus \cdots \oplus t_n$ with $m,n \geq 0$, pairwise $\oplus$-distinct, $\oplus$-reduced standard terms $c'_1,\ldots,c'_m \in \mathsf{C}$, and pairwise $\oplus$-distinct, $\oplus$-reduced standard terms $t_1,\ldots,t_n \notin \tilde{\mathsf{C}}$. (Recall that $\theta$ is $\oplus$-reduced.) First assume that $r' = r\theta$, which implies that $n > 1$. Then we can set $t' = t$ since $t' = t \sim r\theta = r'$. Otherwise, since $c$ does not contain a bad subterm, $r'$ is a complete bad subterm of some $s = c'_j$ (for some $j \in \{1,\ldots,m\}$) or some $s = t_i$ (for some $i \in \{1,\ldots,n\}$). In any case, the term $s$ does not coincide with any of $c_1,\ldots,c_k$, because these terms do not contain bad subterms. Hence, $s$ is equivalent to some subterm $t''$ of $t$. Moreover, note that $s$ is $\oplus$-reduced. Therefore, because $r'$ is a complete bad subterm of $s$, by Lemma 4, there exists a complete bad subterm $t'$ of $t''$, and thus of $t$, such that $t' \sim r'$. $\square$

Proof of Lemma 6.

We proceed by structural induction on $s$:

- $s = x$ *is a variable*: We can take $t' = t$.
- $s$ *is standard*: Then $s \neq t$, and thus, for one of the direct subterms $s'$ of $s$, $s'\theta$ has to contain $t$ as a complete bad subterm. By the induction hypothesis, there exists a variable $x \in \mathrm{var}(s') \subseteq \mathrm{var}(s)$ such that $\theta(x)$ contains a complete bad subterm $t'$ with $t' \simeq_{\mathsf{C}} t$.
- $s \in \mathsf{C}^{\oplus}$: This case is not possible, since $s = s\theta$ is $\mathsf{C}$-dominated, and hence, cannot contain a bad subterm.
- $s \mathrel{\widehat{=}}_{AC} c \oplus s'$, *where* $c \in \mathsf{C}^{\oplus}$ *and* $s' \notin \tilde{\mathsf{C}}$ *is standard, but not a variable*: Then, $t \neq s\theta$ since $s\theta$ is a good term. Moreover, $c$ is $\mathsf{C}$-dominated (since it belongs to $s$) and therefore cannot have $t$ as a subterm. Hence, $t$ must be a subterm of $s'\theta$ and we can use the induction hypothesis.
- $s \mathrel{\widehat{=}}_{AC} c \oplus x$, *for* $c \in \mathsf{C}^{\oplus}$ *and a variable* $x$: If $t = s\theta$, we can take $t' = \theta(x)$ (note that $t' \simeq_{\mathsf{C}} t$). Otherwise, since $c$ is $\mathsf{C}$-dominated, and therefore does not contain complete bad subterms, it follows that $t$ is a subterm of $\theta(x)$. In this case we can take $t' = t$. □

## B Proofs for Section 4

Proof of Lemma 10.

We start by showing that matching of $\mathsf{C}$-dominated terms modulo XOR yields a uniquely determined matcher modulo XOR, if any, and this matcher can be computed in polynomial time.

For this purpose, we first extend the notion of a normal form, and hence, the operator $\ulcorner\cdot\urcorner$, which up to now was only defined on $\mathsf{C}$-dominated terms, to all terms. We fix some linear ordering $<_t$ on terms. Given a term $t$, the normal form $\ulcorner t \urcorner$ of $t$ is obtained by first computing the $\oplus$-reduced form $t'$ of $t$, which is uniquely determined modulo AC. Then, we consider all complete, non-standard subterms $s$ of $t'$ in a bottom-up manner. These terms are of the form $c \oplus t_1 \oplus \cdots \oplus t_n$, where $c \in \mathsf{C}^{\oplus}$ and $t_1, \ldots, t_n \notin \tilde{\mathsf{C}}$ are standard terms. We order the terms $t_1, \ldots, t_n$ according to $<_t$, resulting in $t_{i_1} <_t \cdots <_t t_{i_n}$ for indices $i_1, \ldots, i_n \in \{1, \ldots, n\}$ with $\{i_1, \ldots, i_n\} = \{1, \ldots, n\}$. Now, we replace $c \oplus t_1 \oplus \cdots \oplus t_n$ by $\ulcorner c \urcorner \oplus (t_{i_1} \oplus (t_{i_2} \oplus (\cdots \oplus t_{i_n}) \cdots)))$.

*Claim 1.* Let $s$ be a $\mathsf{C}$-dominated term and $t$ be a ground term. Then, the matcher of $s$ against $t$ is uniquely determined modulo XOR, i.e., if $s\theta \sim t$ and $s\theta' \sim t$ for substitutions $\theta$ and $\theta'$, then $\theta(x) \sim \theta'(x)$ for every $x \in \mathrm{var}(s)$. Moreover, the matcher of $s$ against $t$ can be computed in polynomial time in the size of $s$ and $t$.

*Proof (of Claim 1)* We show how to compute the unique matcher (modulo XOR) of $s$ against $t$. The computed matcher will be in normal form. First, for substitutions $\sigma_1$ and $\sigma_2$ we define $\sigma_1 \sqcup \sigma_2$ as $\sigma_1 \cup \sigma_2$ if for each $x \in \mathrm{dom}(\sigma_1) \cap \mathrm{dom}(\sigma_2)$ we have that $\sigma_1(x) = \sigma_2(x)$. Otherwise, $\sigma_1 \sqcup \sigma_2$ is undefined.

We may assume that both $s$ and $t$ are in normal form. Now, we obtain the matcher $\sigma$ of $s$ against $t$ recursively as follows. We consider the following cases:

1. $s = x$ is a variable: Then $\sigma = \{t/x\}$.
2. $s$ is a ground term: Then $\sigma = \emptyset$ if $s = t$. Otherwise, the matcher does not exist. (We can consider syntactic equality here, since $s$ and $t$ are in normal form.)

3. $s = c \oplus s'$, for a ground term $c \in \mathsf{C}^{\oplus}$ and a non-ground, standard term $s'$: Then $\sigma$ is the matcher of $s'$ against the term $\ulcorner c \oplus t \urcorner$. (Clearly, if $\sigma$ is the matcher of $s'$ against $c \oplus t$, then $\sigma$ is also the matcher of $c \oplus s'$ against $t$.)
4. $s = f(s_1, \ldots, s_n)$ is non-ground with $f \neq \oplus$: If $t = f(t_1, \ldots, t_n)$, we take $\sigma = \sigma_1 \sqcup \cdots \sqcup \sigma_n$, where $\sigma_i$, for $i \in \{1, \ldots, n\}$, is the matcher of $s_i$ against $t_i$. Otherwise, if such a $\sigma$ does not exist, the matcher does not exist.

It is easy to see that because $s$ is assumed to be $C$-dominated and in normal form, the cases considered above are exhaustive. Also, the algorithm computes a matcher of $s$ against $t$, if it exists. Finally, it is easy to verify that matchers are uniquely determined modulo XOR. This completes the proof of the claim.

Now, we are ready to prove Lemma 10: The domain of every substitution in $\Sigma(t)$ is polynomial, since it is a subset of $\mathrm{var}(t)$. Hence, it suffices to show that for every variable in the domain there are only exponentially many possible values and these values can be computed effectively. This is clear for the case (i) and (ii) in Definition 4, as the size of $\mathsf{C}^{\oplus}_{\mathsf{norm}}$ is bounded exponentially in the size of $\mathsf{C}$.

As for the case (iii), let $s, x$ and $\theta$ be as in this case. Note that $s$ is $\mathsf{C}$-dominated. Hence, by our claim and since $\theta$ is in normal form, $\theta$ is the unique matcher of $s$ against some $c \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$. Because $\theta$ can be computed from $s$ and $c$ in polynomial time and, moreover, both $s$ and $c$ range over exponentially bounded sets (in fact, the size of $\mathscr{F}(t)$ is polynomial and the size of $\mathsf{C}^{\oplus}_{\mathsf{norm}}$ is exponential in the size of $t$ and $\mathsf{C}$), the lemma follows.

Proof of Lemma 12.

Let $t$ and $\theta$ be like in the lemma. By Definition 5, it is easy to see that $\sigma = \sigma(t, \theta) \in \Sigma(t)$. It is also easy to see that there exists $\theta'$ such that $\theta = \sigma\theta'$ and the domain of $\theta'$ is the set of all variables that occur in some $\sigma(x)$ for $x \in \mathrm{dom}(\sigma)$. Note that $\theta'$ is uniquely determined. Let $t'$ be a subterm of $t$. We need to show that $\ulcorner t'\theta \urcorner = \ulcorner t'\sigma \urcorner\theta'$. We proceed by structural induction on $t'$.

Obviously, the claim is true if $t'$ is ground. So, in what follows, we assume that $t'$ is non-ground.

First, suppose that $t' = x \in \mathrm{var}(t)$: We distinguish the following cases:

(a) If $\sigma(x)$ was defined according to Definition 5, (a), then $\sigma(x) = \theta(x)$. It follows that $\ulcorner x\theta \urcorner = \ulcorner x\sigma \urcorner\theta'$.
(b) Otherwise, if $\sigma(x)$ was defined according to Definition 5, (b), then $x \in \mathscr{F}(t)$, $\theta(x) = c \oplus s'$, for $c \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$ and some normalized standard term $s'$ not in $\tilde{\mathsf{C}}$, and $\sigma(x) = c \oplus x$. It follows that $\theta'(x) = s'$ and $\ulcorner x\sigma \urcorner\theta' = \ulcorner c \oplus x \urcorner\theta' = (c \oplus x)\theta' = c \oplus s' = \ulcorner c \oplus s' \urcorner = \ulcorner x\theta \urcorner$.
(c) Otherwise, if $\sigma(x)$ was defined according to Definition 5, (c), then $\sigma(x) = x$ and $\theta'(x) = \theta(x)$. Since $\theta(x)$ is normalized, it follows that $\ulcorner x\theta \urcorner = \ulcorner x\sigma \urcorner\theta'$.

Second, suppose that $t' = f(t_1, \ldots, t_n)$, for $f \neq \oplus$: By the induction hypothesis, it follows that $\ulcorner t'\theta \urcorner = f(\ulcorner t_1\theta \urcorner, \ldots, \ulcorner t_n\theta \urcorner) = f(\ulcorner t_1\sigma \urcorner\theta', \ldots, \ulcorner t_n\sigma \urcorner\theta') = \ulcorner t'\sigma \urcorner\theta'$.

Now, suppose that $t' \sim c \oplus x$, for a ground term $c \in \mathsf{C}^{\oplus}$ (note that $x \in \mathscr{F}(t)$): We distinguish the following cases:

(a) If $\sigma(x)$ was defined according to Definition 5, (a), then $\sigma(x) = \theta(x)$. It follows that $\ulcorner t'\theta \urcorner = \ulcorner c \oplus x\theta \urcorner = \ulcorner c \oplus x\sigma \urcorner$ which is ground and thus equal to $\ulcorner c \oplus x\sigma \urcorner\theta' = \ulcorner t'\sigma \urcorner\theta'$. Hence, we conclude that $\ulcorner t'\theta \urcorner = \ulcorner t'\sigma \urcorner\theta'$.
(b) Otherwise, if $\sigma(x)$ was defined according to Definition 5, (b), then $x \in \mathscr{F}(t)$, $\theta(x) = c' \oplus s'$, for $c' \in \mathsf{C}^{\oplus}_{\mathsf{norm}}$ and some normalized standard term $s'$ not in $\tilde{\mathsf{C}}$, and $\sigma(x) = c' \oplus x$.

It follows that $\theta'(x) = s'$ and $\ulcorner t'\sigma\urcorner\theta' = \ulcorner c \oplus c' \oplus x\urcorner\theta' = \ulcorner c \oplus c'\urcorner \oplus x\theta' = \ulcorner c \oplus c'\urcorner \oplus s' = \ulcorner c \oplus c' \oplus s'\urcorner = \ulcorner t'\theta\urcorner$.

(c) Otherwise, if $\sigma(x)$ was defined according to Definition 5, (c), then $\sigma(x) = x$ and $\theta'(x) = \theta(x)$. Since $x \in \mathscr{F}(t)$ and items (a) and (b) of Definition 5 do not hold, $\theta(x) = \theta'(x)$ is a normalized standard term not in $\tilde{\mathsf{C}}$. It follows that $\ulcorner t'\theta\urcorner = \ulcorner c \oplus \theta(x)\urcorner = \ulcorner c\urcorner \oplus \theta(x) = \ulcorner c\urcorner \oplus \theta'(x) = \ulcorner c \oplus x\urcorner\theta' = \ulcorner (c \oplus x)\sigma\urcorner\theta' = \ulcorner t'\sigma\urcorner\theta'$.

Finally, suppose that $t' \sim c \oplus s$, for a ground term $c \in \mathsf{C}^{\oplus}$ and a (non-ground) $\mathsf{C}$-dominated, standard subterm $s$ of $t'$ with $s \notin \tilde{\mathsf{C}}$ and $s \notin \mathrm{var}(t)$: Note that $s \in \mathscr{F}(t)$. We distinguish the following cases:

(a) If $s\theta \in \mathsf{C}^{\oplus}$, then $\sigma(x)$, for all $x \in \mathrm{var}(s)$, was defined according to Definition 5-(a). Hence, $\sigma(x) = \theta(x)$ for all $x \in \mathrm{var}(s)$, and thus $s\sigma = s\theta$. Moreover, $s\sigma$ is ground, because $\theta$ is assumed to be ground. Therefore $\ulcorner t'\theta\urcorner = \ulcorner c \oplus s\theta\urcorner = \ulcorner c \oplus s\sigma\urcorner = \ulcorner c \oplus s\sigma\urcorner\theta' = \ulcorner t'\sigma\urcorner\theta'$.

(b) Otherwise, if $s\theta \notin \mathsf{C}^{\oplus}$, by the induction hypothesis it follows that $\ulcorner s\theta\urcorner = \ulcorner s\sigma\urcorner\theta'$. We also know that $s\sigma$ is not in $\tilde{\mathsf{C}}$ (otherwise, since $\theta = \sigma\theta'$, the term $s\theta$ would also be in $\tilde{\mathsf{C}}$). Moreover, since $s\theta \notin \tilde{\mathsf{C}}$, we obtain that $\ulcorner t'\theta\urcorner = \ulcorner c\urcorner \oplus \ulcorner s\theta\urcorner = \ulcorner c\urcorner \oplus \ulcorner s\sigma\urcorner\theta' = \ulcorner (c \oplus s)\sigma\urcorner\theta' = \ulcorner t'\sigma\urcorner\theta'$. □