P. Lafourcade, D. Lugiez,
and R. Treinen

Intruder Deduction for the Equational Theory of Exclusive-or with Distributive Encryption.

Research Report LSV-05-19
October 2005

Laboratoire
Spécification
et
Vérification

CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Intruder Deduction for the Equational Theory of *Exclusive-or* with Distributive Encryption[*]

Pascal Lafourcade
LIF, Université Aix-Marseille 1 & CNRS UMR 6166
LSV, ENS de Cachan & CNRS UMR 8643 & INRIA Futurs project SECSI

Denis Lugiez
LIF, Université Aix-Marseille 1 & CNRS UMR 6166

Ralf Treinen
LSV, ENS de Cachan & CNRS UMR 8643 & INRIA Futurs project SECSI

October 5, 2005

Abstract

Cryptographic protocols are small programs which involve a high level of concurrency and which are difficult to analyze by hand. The most successful methods to verify such protocols are based on rewriting techniques and automated deduction in order to implement or mimic the process calculus describing the execution of a protocol.

We are interested in the intruder deduction problem, that is the vulnerability to passive attacks, in presence of the theory of an encryption operator which distributes over the *exclusive-or*. This equational theory describes very common properties of cryptographic primitives. Solutions to the intruder deduction problem modulo an equational theory are known for the cases of *exclusive-or*, of Abelian groups, of a homomorphism symbol alone, and of combinations of these theories. In this paper we consider the case where the encryption distributes over *exclusive-or*. The interaction of the distributive law of the encryption with the cancellation law of *exclusive-or* leads to a much more complex decision problem. We prove decidability of the intruder deduction problem for an encryption which distributes over *exclusive-or* with an EXPTIME procedure and we give a PTIME decision procedure relying on prefix rewrite systems for a restricted case, the *binary* case.

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

1 Introduction

Cryptographic protocols are ubiquitous in distributed computing applications. They are employed for instance in internet banking, video on demand services, wireless communication, or secure UNIX services like `ssh` or `scp`. Cryptographic protocols can be described as relatively simple programs which are executed in an untrusted environment.

Verifying protocols is notoriously difficult, and even very simple protocols which look completely harmless may have serious security holes, as it was demonstrated by the flaw of the Needham-Schroeder protocol found by Lowe [Low95] using a model-checking tool. It took 17 years since the protocol was published to find an attack, a so-called *man in the middle attack*. An overview of *authentication protocols* known a decade ago can be found in [CJ97], a more recent data base of protocols and known flaws is [Jac].

There are different approaches to modeling cryptographic protocols and analyzing their security properties: process calculi like the *spi-calculus* [AG99], so-called cryptographic proofs (see, for instance, [AR00]), and the approach of Dolev and Yao [DY83] which consists in modeling an attacker by a deduction system. This deduction system specifies how the attacker can obtain new information from previous knowledge, which he has either obtained by eavesdropping the communication between honest protocol participants (in case of a *passive* attacker), or by eavesdropping and fraudulently emitting messages, thus provoking honest protocol participants to reply according to the protocol rules (this is the case of a so-called *active* attacker). We call *intruder deduction problem* the question whether a passive eavesdropper can obtain a certain information from messages that he observes on the network.

Algebraic properties. Classically, the verification of cryptographic protocols was based on the so-called *perfect cryptography assumption* which states that it is impossible to obtain any information about an encrypted message without knowing the exact key necessary to decrypt this message. Unfortunately, this perfect cryptography assumption has been proven too idealistic: There are protocols which can be proven secure under the perfect cryptography assumption, but which are in reality insecure since an attacker can use properties of the cryptographic primitives in combination with the protocol rules to learn some secret informations. These properties are typically expressed as equational axioms (so-called algebraic properties). The executability of cryptographic methods like DES or the more recent AES heavily relies on the algebraic properties of the *exclusive-or* operation (*exclusive-or* also written $\oplus$: associativity-commutativity, existence of a unit element 0 and nilpotence $x \oplus x = 0$). Algebraic properties which are not used explicitly in the protocol can also be exploited by an attacker to mount an attack, see [CDL05] for an overview of the verification of cryptographic protocols in presence of algebraic properties. Many results have already been obtained in this area, both for the intruder deduction problem and for the preservation of secrecy under active attacks. The intruder deduction problem in the case of the equational axioms of *exclusive-or* and of Abelian groups is decidable in polynomial time [CLS03, CKRT03]. Likewise, the intruder deduction problem is decidable in polynomial time [CLT03] in the case of the equational theory of a homomorphism. In [CR05], the authors provide an algorithm that combines decision procedures for the active intruder in the case of *disjoint* equational theories.

Our Contribution. In this paper we investigate the intruder deduction problem for cryptographic protocols that use an encryption operation which distributes over the binary *exclusive-or* operator $\oplus$, that is the encryption of a sum $\oplus$ is the sum of the encryptions. We do not assume that the set of encryption keys is finite. Rather, any term can be used as an encryption key. Contrary to the aforementioned combination algorithm, the equational theories of the $\oplus$ operation and of the encryption operation are not disjoint, therefore we must design an algorithm dedicated to this case. Our results can be summarized as follows:

(i) We can decide the intruder deduction problem in exponential time in the general case.

(ii) We can decide the intruder deduction problem in polynomial time in the so-called *binary* case, i.e. when the set of assumptions and the goal do not contain more than two consecutive applications of $\oplus$.

Our solution is similar to the approach of [CLS03, CLT03]. It generalizes McAllester's *locality* method explained in Section 4 and relies on proof transformations to establish a locality theorem. The improved complexity result in the binary case is obtained by a new approach based on prefix word rewriting.

Plan of the paper: We present in Section 2 the usual notions needed in the rest of the paper. In Section 3 we introduce the Dolev-Yao model of intruder capacities extended by a rewrite system modulo AC and present the rewrite system investigated in this paper. In Section 4 we explain the generalization of McAllester's proof technique. In the following two sections we provide the two main ingredients which allow us to obtain an EXPTIME algorithm: We show in Section 5 that the set of instances of our proof system is decidable in polynomial time, and we show in Section 6 a proof normalization result which is the technical core of our paper. We discuss in Section 7 the restriction to the binary case and give a decision procedure in PTIME using prefix rewrite systems. Finally, we conclude in Section 8.

2 Preliminaries

We summarize some basic notations used in this paper, see [DJ90, BN98] for an overview of rewriting.

We denote as usual with U^* the set of all finite sequences of symbols from U.

Let Σ be a signature. $T(\Sigma, X)$ denotes the set of terms over the signature Σ and the set of variables X, that is the smallest set such that:

1. $X \subseteq T(\Sigma, X)$;
2. if $t_1, \ldots, t_n \in T(\Sigma, X)$, and $f \in \Sigma$ has arity $n \geq 0$, then $f(t_1, \ldots, t_n) \in T(\Sigma, X)$.

We abbreviate $T(\Sigma, \emptyset)$ as $T(\Sigma)$; elements of $T(\Sigma)$ are called Σ-*ground terms*. The set of variables occurring in a term t is denoted by $\mathcal{V}(t)$.

The *set of occurrences* of a term t is defined recursively as $\mathcal{O}(f(t_1, \ldots, t_n)) = \{\epsilon\} \cup \bigcup_{i=1\ldots n} i \cdot \mathcal{O}(t_i)$. For instance, $\mathcal{O}(f(a, g(b, x))) = \{\epsilon, 1, 2, 21, 22\}$. The *size* $|t|$ of a term t is defined as its number of occurrences, that is $|t| =$ *cardinality*$(\mathcal{O}(t))$. We extend the notion of size to a set of terms T by $|T| = \Sigma_{t \in T} |T|$. If $o \in \mathcal{O}(t)$ then the *subterm of* t *at position* o is defined recursively by:

- $t \mid_\epsilon = t$
- $f(t_1, \ldots, t_n) \mid_{j \cdot o} = t_j \mid_o$

We call a term r a *subterm* of a term t if r is a subterm of t at some position of t. If t and s are terms and $o \in \mathcal{O}(t)$ then the *grafting* of s onto t at position o is defined recursively as:

- $t[\epsilon \leftarrow s] = s$
- $f(t_1, \ldots, t_n)[j \cdot o \leftarrow s] = f(t_1, \ldots, t_{j-1}, t_j[o \leftarrow s], t_{j+1}, \ldots, t_n)$

For instance, $f(a, g(b, x))[22 \leftarrow h(c)] = f(a, g(b, h(c)))$.

A Σ-*equation* is a pair $(l, r) \in T(\Sigma, X)$, commonly written as $l = r$. The relation $=_E$ generated by a set of Σ equations E is the smallest congruence on $T(\Sigma)$ that contains all ground instances of all equations in E.

A Σ-*rewriting system* R is a finite set of so-called *rewriting rules* $l \rightarrow r$ where $l \in T(\Sigma, X)$ and $r \in T(\Sigma, \mathcal{V}(l))$. A term $t \in T(\Sigma, X)$ *rewrites* to s in one step by R if there is a rewriting rule $l \rightarrow r$ in R, an occurrence o and a substitution σ such that $t \mid_o = l\sigma$ and $s = t[o \leftarrow r\sigma]$. If the occurrence o is the empty string, that is if rewriting takes places at the root of the tree, then t *prefix-rewrites* in one step to s, noted $t \mapsto s$. We write $\rightarrow^*$ for the reflexive and transitive closure of $\rightarrow$, and $\mapsto^*$ for the reflexive and transitive closure of $\mapsto$. A term t is in *normal form* if there is no term s with $t \rightarrow s$. If $t \rightarrow^* s$ and s is a normal form then we say that s is a *normal form of* t, and write $s = t\downarrow$.

A term rewriting system is called *convergent* if it is:

- *strongly terminating*, that is if there is no infinite sequence of the form $t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow \cdots$.
- *locally confluent*, that is if $t \rightarrow s_1$ and $t \rightarrow s_2$ then there exists a term r with $s_1 \rightarrow^* r$ and $s_2 \rightarrow^* r$.

By a well known result (see, e.g., [DJ90]), every convergent rewrite system is *confluent*, that is if $t \rightarrow^* s_1$ and $t \rightarrow^* s_2$ then there exists a term r with $s_1 \rightarrow^* r$ and $s_2 \rightarrow^* r$. As a consequence, in a convergent rewrite system every term has a unique normal form

By R/S we denote the so-called *class rewrite system* composed of a set $R = \{l_i \to r_i\}$ of rewrite rules and a set $S = \{u_i = v_i\}$ of equations. Generalizing the notion of term rewriting, we say that s rewrites to t *modulo* S, denoted $s \to_{R/S} t$, if $s =_S u[l\sigma]_p$ and $u[r\sigma]_p =_S t$, for some context u, position p in u, rule $l \to r$ in R, and substitution σ.

Let T be a set of terms, the mapping $S : T \to T$ is idempotent if for every $X \subseteq T$: $S(S(X)) = S(X)$. The mapping S is monotone if for all $X, Y \subseteq T$: if $X \subseteq Y$ then $S(X) \subseteq S(Y)$. S is transitive if for all $X, Y, Z \subseteq T$, $X \subseteq S(Y)$ and $Y \subseteq S(Z)$ implies $X \subseteq S(Z)$.

Proposition 1 *Let S be a mapping from sets of terms to sets of term. If the mapping S is idempotent and monotone then it is transitive.*

Proof: straightforward. □

3 A Dolev-Yao Model for Rewriting Modulo AC

We consider the classic model of deduction rules [DY83] introduced by Dolev and Yao in order to model the deductive capacities of a passive intruder. We present here an extension of this model where we assume an associative and commutative operator $\oplus$, and an equational theory E which can be exploited by the intruder to mount an attack. Knowledge of the intruder is represented by terms built over a finite signature $\Sigma = \{\langle\cdot,\cdot\rangle, \{\cdot\}_\cdot, \oplus\} \uplus \Sigma_0$, where Σ_0 is a set of constant symbols. The term $\langle u, v\rangle$ represents the pairing of the two terms u and v, and $\{u\}_v$ represents the encryption of the term u by the term v. For the sake of simplicity we here only consider symmetric encryption.

The equational theory E is represented by a convergent rewrite system R modulo AC, that is R is terminating and confluent modulo associativity and commutativity of $\oplus$, and for all terms $t, s \in T(\Sigma)$ we have that $t =_E s$ if and only if $t\downarrow_{R/AC} =_{AC} s\downarrow_{R/AC}$. The deduction system describing the deductive capacities of an intruder is given in Figure 1. This deduction system is composed of the following rules: (A) the intruder may use any term which is in his initial knowledge, (P) the intruder can build a pair of two messages, (UL, UR) he can

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

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

$$(C)\frac{T \vdash u \qquad T \vdash v}{T \vdash \{u\}_v\downarrow_{R/AC}} \qquad (D)\frac{T \vdash r \qquad T \vdash v}{T \vdash u\downarrow_{R/AC}} \quad \begin{array}{l}\textit{if } r =_E \{u\}_v \\ \textit{and } u\downarrow_{R/AC} = u\end{array}$$

$$(GX)\frac{T \vdash u_1 \qquad \ldots \qquad T \vdash u_n}{T \vdash u_1 \oplus \ldots \oplus u_n\downarrow_{R/AC}}$$

Figure 1: A Dolev-Yao proof system working on normal forms by a rewrite system R modulo AC

extract each member of a pair, (C) he can encrypt a message u with a key v, (D) if he knows a key v he can decrypt a message encrypted by v. Sometimes, we shall annotate the rules (C) and (D) by the key that they use, yielding rules (C_v) and (D_v). Because of the algebraic properties of the $\oplus$ operator, we add a family of rules (GX) which allows the intruder to build a new term from an arbitrary number of already known terms by using the $\oplus$ operator. The need for such a variadic rule (instead of just a binary rule) will become apparent in Section 4.

In fact, this deductive system is equivalent in deductive power to a variant of the system in which terms are not automatically normalized, but in which arbitrary equational proofs are allowed at any moment of the deduction. The equivalence of the two proof systems has been shown in [CLT03] without AC axioms; in [LLT04] this has been extended to the case of a rewrite system modulo AC.

From now on, we assume that the set T consists only of normalized terms.

We will investigate the Dolev-Yao deduction system modulo the following rewrite system, which corresponds to the theory XDE (*eXclusive-or* with a Distributive Encryption over $\oplus$): $0 \oplus x \to x$; $x \oplus x \to 0$; $\{0\}_z \to 0$; $\{x \oplus y\}_z \to \{x\}_z \oplus \{y\}_z$. We will omit the index R/AC and write $\downarrow$

4 Locality and Complexity of the Intruder Deduction Problem

4.1 A Generalization of Locality

Our starting point is the locality technique introduced by McAllester [McA93]. He considers deduction systems which are represented by finite sets of Horn clauses. He shows that there exists a polynomial-time algorithm to decide the deducibility of a term w from a finite set of terms T if the deduction system has the so-called *locality property*. A deduction system has the *locality property* if any proof can be transformed into a *local proof*, that is a proof where all nodes are syntactic subterms of $T \cup \{w\}$. The idea of the proof is to check existence of a local proof by a saturation algorithm which computes all syntactic subterms of $T \cup \{w\}$ that are deducible from T.

An abstract version of this algorithm is presented in Figure 2 where S is a function which maps a set of terms to a set of terms such that $S(T)$ is the set of subterms of T (the set of *syntactic* subterms in McAllester's original algorithm). In this algorithm we denote the one-step deduction relation by $\vdash^{=1}$, where we say that w is *one-step deducible* from T if we can obtain w from T with only one application of a rule of the proof system.

There are two main restrictions in McAllester's approach: the deduction system must be *finite* and the notion of locality is restricted to *syntactic subterms*. These restrictions raise a serious problem when we are working modulo AC, as it is already pointed out in [CLS03]. If we used the rule (GX) only in its binary form then we would have to consider all possible subterms modulo AC. Unfortunately, there is in general an exponential number of subterms modulo AC of

```
Input: $T, w$
$Sub \leftarrow S(T, w)$;
repeat
  $T_p \leftarrow T$;
  foreach $t \in Sub$ do
    if $T_p \vdash^{=1} t$ then $T \leftarrow T \cup \{t\}$ fi
  od
until $T_p = T$
```

Figure 2: Checking existence of an S-local proof.

a given term. The solution proposed in [CLS03], and which we also adopt here, is to use the rule (GX) with an arbitrary number of hypotheses. However, we are now stuck with an infinite number of rules. Fortunately, we can implement the test in the loop in McAllester's algorithm in a clever way that allows to get a more efficient procedure.

In the rest of the paper we denote $T \cup \{w\}$ by T, w.

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

The following theorem generalizes McAllester's result.

Theorem 1 *Let S be a function mapping a set of terms to a set of terms, and P a proof system. Let T be a set of terms, let w be a term and let n be $|T, w|$. If:*

1. *one-step deducibility of $S \vdash u$ in P is decidable in time $g(|S, u|)$ for any term u and set of terms S,*
2. *the set $S(T, w)$ can be constructed in time $f(n)$,*
3. *P is S-local,*

then provability of $T \vdash w$ in the proof system P is decidable in time $f(n) + f(n) * f(n) * g(f(n))$ (non-deterministic if one of (2), (1) is non-deterministic).

Proof: By S-locality of the proof system, provability of $T \vdash w$ is equivalent to existence of an S-local proof for $T \vdash w$. Existence of an S-local proof of $T \vdash w$ is checked by the algorithm of Figure 2 and the computation of *Sub* takes time $f(n)$. As a consequence, the cardinality of *Sub* is bounded by $f(n)$. Hence, the number of iterations of the outer loop is bounded by $f(n)$, and for each iteration of the outer loop the number of iterations of the inner loop is also bounded by $f(n)$. Since the size of T is bounded by $f(n)$ the conditional instruction can be performed in time $g(f(n))$. □

Therefore the roadmap to prove deducibility in our more general setting is:

(i) show that one-step deducibility can be tested in time $f(n)$, for some complexity measure f,

(ii) define a notion of subterms which can be computed in time $g(n)$, for some complexity measure g,

(iii) show locality with respect to this notion of subterms.

We first show that one-step deducibility is decidable in polynomial time for the equational theory XDE. Then we define a notion of subterms (Definition 5) which yields an exponential set and which enables us to prove a locality theorem (Theorem 3), yielding the decidability of the intruder deduction problem. In Section 7, we shall define a polynomial notion of subterms in the binary case, which allows to get a polynomial-time complexity in this case.

5 One-Step Deducibility

Our method is inspired by the method used in [Nar96] to solve an unification problem modulo AC-like theories.

Definition 2 *Let u be a term in normal form, u is* headed with $\oplus$ *if u is of the form $u_1 \oplus \ldots \oplus u_n$ with $n > 1$. Otherwise u is* not headed with $\oplus$. *We define the function atoms(u) as follows:*

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

Example 1 *$t_1 = u \oplus \langle v, w \rangle$ is headed with $\oplus$, but $t_2 = \langle u, v \oplus w \rangle$ is not, hence atoms$(t_2) = \{t_2\}$ and atoms$(t_1) = \{u, \langle v, w \rangle\}$.*

The definition of atoms(T) is generalized to sets of terms T in normal form by setting atoms$(T) := \bigcup_{t \in T}$ atoms(t). According to the definition, the function atoms is monotone and idempotent.

Theorem 2 *Let T be a set of terms and w be a term. The question whether w is one-step deducible from T with any of the rules of Figure 1 is decidable in polynomial time.*

Proof: We show how to decide one-step deducibility only for the family of rules (GX). Checking one-step deducibility for the other deduction rules of Figure 1 is straightforward since it is a simple matching problem for a finite number of patterns.

With the following transformation the problem of testing one-step deducibility for the rule (GX) is equivalent to the solvability of a system of linear Diophantine equations over $\mathbb{Z}/2\mathbb{Z}$.

- Input:

  - A finite set of terms $T = \{t_0, \dots, t_n\}$
  - A term w

- Output
  - A system $D(T, w)$ of linear Diophantine equations over the variables $X = \{x_0, \dots, x_n\}$ such that $T \vdash w$ if and only if $D(T, w)$ is solvable in $\mathbb{Z}/2\mathbb{Z}$.

- Algorithm
  - To each t_i we associate the variable x_i for $i = 0, \dots, n$.
  - Let $A = \{a_1, \dots, a_m\}$ be the set of atoms of T, w.
  - If $u \in A$ and $t \in (T, w)$, let $\delta(u, t)$ denote the number of occurrences of the atom u in t.
  - For each $a_i \in A$ we introduce the equation:

$$\delta(a_i, w) = \sum_{j=0}^{n} \delta(a_i, t_j) * x_j$$

which states that the number of occurrences of a_i in w is equal to the sum of the number of occurrences of a_i in (a sum of) t_j's. The system $D(T, w)$ is the conjunction of these equations:

$$D(T, w) := \bigwedge_{i=1}^{m} \sum_{j=0}^{n} \delta(a_i, t_j) * x_j = \delta(a_i, w)$$

Lemma 1 shows that this system has a solution if and only if w is deducible from T in one step using the rule (GX). Since the former problem is in PTIME [KKS87], w is deducible in one-step from T in PTIME. □

Example 2 *Let $T = \{a_1 \oplus a_2 \oplus a_3, a_1 \oplus a_4, a_2 \oplus a_4\}$ and $w = a_1 \oplus a_2$, where all the a_i are not headed with $\oplus$. We introduce numerical variables x_0, x_1, x_2, that is one numerical variable for each element of T:*

$$\begin{array}{lll} x_0 & \textit{for} & a_1 \oplus a_2 \oplus a_3 \\ x_1 & \textit{for} & a_1 \oplus a_4 \\ x_2 & \textit{for} & a_2 \oplus a_4 \end{array}$$

For every atom a_i we create an equation, we get the following equation system:

$$\begin{cases} a_1 & : & x_0 + x_1 = 1 \\ a_2 & : & x_0 + x_2 = 1 \\ a_3 & : & x_0 = 0 \\ a_4 & : & x_1 + x_2 = 0 \end{cases}$$

We solve this system over $\mathbb{Z}/2\mathbb{Z}$ to know if w is deducible in one-step from T.

Lemma 1 *Let $T_S = \{t_1, \dots, t_n\}$ and let w_S be such that for $1 \leq i \leq n$, $t_i = c_{1,i} * a_1 \oplus \dots \oplus c_{m,i} * a_m$ and $w_S = d_1 * a_1 \oplus \dots \oplus d_m * a_m$ where $\{a_1, \dots, a_m\}$ is the set of atoms of T_S, w_S. Let S be the following system of equations:*

$$\left\{ \begin{array}{ccc} c_{1,1}x_1 + \dots + c_{1,n}x_n & = & d_1 \\ \vdots & \vdots & \vdots \\ c_{m,1}x_1 + \dots + c_{m,n}x_n & = & d_m \end{array} \right.$$

Then (S) is satisfiable if and only if w_S is deducible from T_S with exactly one instance of the rule (GX).

Proof: The proof is in Appendix A □

Notice that the transformation adapted in the proof of Theorem 2 can be used to prove the one-step deducibility for several other AC-like theories, for example the theory of Abelian groups. In the general case the domain over which this system of equations is solved depends on the equational theory considered. For instance, in case of Abelian groups the equations have to be solved over $\mathbb{Z}$.

6 Locality in General Case for XDE

In the first part of this section we list all the definitions of subterms that we shall use. Then we define the different kinds of proofs and the relevant properties that we need. Finally we prove that $S_\oplus$-locality is decidable in EXPTIME for the XDE equational theory.

6.1 Terms and Subterms

We list the different definitions of subterms.as usual:

Definition 3 *The set of* syntactic subterms *of a term t is the smallest set $S(t)$ such that:*

- *$t \in S(t)$.*
- *If $\langle u, v\rangle \in S(t)$ then $u, v \in S(t)$.*
- *If $\{u\}_v \in S(t)$ then $u, v \in S(t)$.*
- *If $u = u_1 \oplus \dots \oplus u_n \in S(t)$ then atoms$(u) \subseteq S(t)$.*

S is extended to a set T of terms in normal form by $(T) := \bigcup_{t \in T} S(t)$.

In the definition of $S(t)$ we do not take care of the distributivity of the encryption rule. Because we work only on normal forms the notion of a syntactic subterm ignores the fact that the term $\{a\}_v \oplus \{b\}_v \oplus \{c\}_v$ is equal to $\{a \oplus b \oplus c\}_v$, and that $a \oplus b \oplus c$ should be considered to be a subterm of $\{a\}_v \oplus \{b\}_v \oplus \{c\}_v$. This is accounted for in the next definition.

Definition 4 *For any term t, $S_T(t)$ is the smallest set such that:*

(i) $S(t) \subseteq S_T(t)$.

(ii) If $n > 1$ and $\{u_1\}_v \oplus \ldots \oplus \{u_n\}_v \in S_T(t)$ then $u_1 \oplus \ldots \oplus u_n \in S_T(t)$.

The definition is extended to a set T of terms in normal form by setting $S_T(T) := \bigcup_{t \in T} S_T(t)$, hence $S(T) \subseteq S_T(T)$.

Example 3 *Let $T = \{\langle a \oplus \{c\}_k, \{d\}_k\rangle, \{a\}_k \oplus \{\{b\}_k\}_k\}$, then $S(T) = T \cup \{a, b, c, d, k, \{a\}_k, \{b\}_k, \{c\}_k, \{d\}_k, \{\{b\}_k\}_k\}$ and $S_T(T) = S(T) \cup \{a \oplus \{b\}_k\}$*

Proposition 2 *For any set of terms $M \subseteq T_\Sigma$, we have:*

- *atoms$(M) \subseteq S(M)$ for any set of terms $M \subseteq T_\Sigma$*
- *atoms$(S_T(M)) \subseteq S_T(M)$*
- *$S(S(M)) = S(M)$ and $S_T(S_T(M)) = S_T(M)$*

Proof: Obvious from the definitions of S, atoms and S_T. □

Example 4 *Let P be a proof of $T \vdash w$, where $T = \{u \oplus v, \{v\}_k, k\}, w = \{u\}_k$.*

$$(GX)\dfrac{(C_k)\dfrac{(A)\dfrac{u \oplus v \in T}{T \vdash u \oplus v} \quad (A)\dfrac{k \in T}{T \vdash k}}{T \vdash \{u\}_k \oplus \{v\}_k} \quad (A)\dfrac{\{v\}_k \in T}{T \vdash \{v\}_k}}{T \vdash \{u\}_k}$$

We compute $S_T(T, w) = \{u, v, u \oplus v, k, \{u\}_k, \{v\}_k\}$. This proof is not S_T-local since $\{u\}_k \oplus \{v\}_k$ is not in $S_T(T, w)$.

Example 4 suggests that we define a new notion of subterm such that $\{u\}_k \oplus \{v\}_k \in S_\oplus(T, w)$. This new definition takes into account the partial sums of $S_T(T)$ and is required to get the locality property.

Definition 5 *Define $S_\oplus$ as all the combinations of terms of $S_T(T)$ by $\oplus$:*

$$S_\oplus(T) := \left\{ (\bigoplus_{s \in M} s) \downarrow \mid M \subseteq S_T(T) \right\}$$

Note that 0 always belongs to $S_\oplus$, that the size of $S_\oplus$ is exponential in the size of T and $S_T(T) \subseteq S_\oplus(T)$.

Example 5 *Let $T = \{\langle a, b\rangle\}$. Then we get $S_T(T) = \{\langle a, b\rangle, a, b\}$ and $S_\oplus(T) = S_T(T) \cup \{0, a \oplus b, b \oplus \langle a, b\rangle, a \oplus \langle a, b\rangle, a \oplus b \oplus \langle a, b\rangle\}$.*

We state now the main properties of subterms that we use in the following.

Proposition 3 *Let A and B be two sets of terms in normal forms. The mappings S, S_T and $S_\oplus$ are monotone and have the property:*

- *$S(A \cup B) = S(A) \cup S(B)$*
- *$S_T(A \cup B) = S_T(A) \cup S_T(B)$*

- $S_\oplus(A) \cup S_\oplus(B) \subseteq S_\oplus(A \cup B)$

Proof: Monotonicity is obvious from the definitions of $S(T)$, $S_T(T)$ and $S_\oplus(T)$. □

In the last of the previous proposition inclusion may hold. For instance, let $S_\oplus(\{a\}) = \{0, a\}$ and $S_\oplus(\{0, b\}) = \{b\}$. Then $S_\oplus(A) \cup S_\oplus(B) = \{0, a, b\} \subseteq S_\oplus(A \cup B) = \{0, a \oplus b, a, b\}$ and $S_\oplus(A) \cup S_\oplus(B) \neq S_\oplus(A \cup B)$.

Lemma 2 *If T be a set of terms in normal form then $S_T(S_\oplus(T)) = S_\oplus(T)$.*

Proof: By definition 4, $S_\oplus(T) \subseteq S_T(S_\oplus(T))$.

We prove the reverse inclusion by induction on the number of applications of the rule for $\oplus$ in the construction of $S_T(S_\oplus(T))$ (step (ii) in Definition 4).

Let $u \in S_T(S_\oplus(T))$, and let n be the number of applications of the rule for $\oplus$. By induction hypothesis, we assume that each term $u' \in S_T(S_\oplus(T))$ obtained with less than n applications of the rule for $\oplus$ is in $S_\oplus(T)$.

Base case $n = 0$: $u \in S_T(v)$ for some $v \in S_\oplus(T)$, where $v = v_1 \oplus \ldots \oplus v_p$ and all $v_i \in S_T(T)$. If $u = v$ then $u \in S_\oplus(T)$.

Otherwise $u \neq v$. In this case $u \in S(v_i) \subseteq S_T(v_i)$ for some i (since $v_i \in S_T(T)$ and $S(S_T(T)) = S_T(T)$). Since $v \in S_\oplus(T)$ there exists a $t_i \in T$ such that $v_i \in S_T(t_i)$. Therefore $v_i \in S_T(t_i) \subseteq S_T(T)$ with $t_i \in T$, hence $u \in S_T(S_T(T)) = S_T(T) \subseteq S_\oplus(T)$ by idempotence of S_T.

Induction step: let $u = u_1 \oplus \ldots \oplus u_n$ be obtained from $\{u_1\}_v \oplus \ldots \oplus \{u_n\}_v \in S_T(S_\oplus(T))$. By induction hypothesis $\{u_1\}_v \oplus \ldots \oplus \{u_n\}_v \in S_\oplus(T)$. Hence there exists a partition $I_1 \cup \ldots \cup I_q = \{1, \ldots, n\}$ such that for every j, $1 \leq j \leq q$, $w_j = \oplus_{i \in I_j} \{u_i\}_v \in S_T(t_j)$. Hence, $\oplus_{i \in I_j} u_i \in S_T(t_j)$ by definition of S_T. As a consequence, $u \in S_\oplus(T)$. □

Corollary 3 *Let M be a set of terms in normal form then $S_\oplus(S_\oplus(M)) = S_\oplus(M)$. The mappings S, S_T and $S_\oplus$ are transitive.*

Proof: The first property is a consequence of Lemma 2 and Proposition 2. The second property is a consequence of the first one and Propositions 1 and 2. □

6.2 Different Kinds of Proofs

We define several notions on proofs that we use in the remainder of the paper.

Definition 6 *Let P be a proof of $T \vdash w$.*

- *A* subproof *P' of P is a sub-tree of P.*
- *The* size of a proof *P, denoted by $|P|$, is the number of nodes in P.*
- *The proof P is* simple *if each node $T \vdash v$ occurs at most once on each branch and a node $T \vdash v$ occurs in every instance of (GX) at most once as hypothesis of the rule (GX).*

- *The proof P is* flat *if there is no (GX) rule immediately above another (GX) rule.*

Since two successive (GX) rules can be merged into a single (GX) rule, each proof can be transformed into an equivalent flat proof. To get a simple proof, we eliminate the part of the proof between two occurrences of the same node in a branch and in the hypothesis of a rule (GX). This simplification terminates since it decreases $|P|$.

Lemma 4 *Let P be a simple proof then (i) there is no rule (D_v) just after a rule (C_v) in P, (ii) there is no rule (C_v) just after a rule (D_v) in P.*

Proof: This is an immediate consequence of the simplicity. □

Definition 7 *Let P be a flat proof of $T \vdash w$. P is a* **$\oplus$-eager** *proof if there is at most one rule (C_v) with the same key v immediately above a (GX) in P and there is no rule (D_v) just after a (GX) with a rule (C_v) just above (GX).*

Intuitively, in a $\oplus$-*eager* proof the (GX) rule is applied as early as possible.

Example 6 *The following proof with $T = \{u, v, k, \{v\}_k\}$ and $w = \{u\}_k$ is not $\oplus$-eager.*

$$(GX)\dfrac{(C)\dfrac{(A)\dfrac{u \in T}{T \vdash u} \qquad (A)\dfrac{k \in T}{T \vdash k}}{T \vdash \{u\}_k} \qquad (C)\dfrac{(A)\dfrac{v \in T}{T \vdash v} \qquad (A)\dfrac{k \in T}{T \vdash k}}{T \vdash \{v\}_k} \qquad (A)\dfrac{\{v\}_k \in T}{T \vdash \{v\}_k}}{T \vdash \{u\}_k}$$

We can transform it into a $\oplus$-eager simple proof:

$$(GX)\dfrac{(C)\dfrac{(A)\dfrac{k \in T}{T \vdash k} \qquad (GX)\dfrac{(A)\dfrac{u \in T}{T \vdash u} \qquad (A)\dfrac{v \in T}{T \vdash v}}{T \vdash u \oplus v}}{T \vdash \{u \oplus v\}_k = \{u\}_k \oplus \{v\}_k} \qquad (A)\dfrac{\{v\}_k \in T}{T \vdash \{v\}_k}}{T \vdash \{u\}_k}$$

However, these proofs are not the shortest ones since there is a smaller proof:

$$(C)\dfrac{(A)\dfrac{u \in T}{T \vdash u} \qquad (A)\dfrac{k \in T}{T \vdash k}}{T \vdash \{u\}_k}$$

Now we present some transformations on proofs.

Proposition 4 *All the transformations of proofs given in Figures 3, 4 and 5 decrease the number of nodes.*

$$(GX)\dfrac{(GX)\dfrac{T\vdash x_1 \qquad \dots \qquad T\vdash x_n}{T\vdash x_1\oplus\dots\oplus x_n} \qquad T\vdash y_1 \qquad \dots \qquad T\vdash y_m}{T\vdash x_1\oplus\dots\oplus x_n\oplus y_1\oplus\dots\oplus y_m}$$

$$\Downarrow$$

$$(GX)\dfrac{T\vdash x_1 \qquad \dots \qquad T\vdash x_n \qquad T\vdash y_1 \qquad \dots \qquad T\vdash y_m}{T\vdash x_1\oplus\dots\oplus x_n\oplus y_1\oplus\dots\oplus y_m}$$

Figure 3: Transformation of (GX)-(GX) into (GX)

$$(GX)\dfrac{(C_v)\dfrac{T\vdash x_1 \qquad T\vdash v}{T\vdash \{x_1\}_v}\dots(C_v)\dfrac{T\vdash x_n \qquad T\vdash v}{T\vdash \{x_n\}_v}(R_1)\dfrac{\vdots}{T\vdash z_1}\dots(R_m)\dfrac{\vdots}{T\vdash z_m}}{T\vdash \{x_1\}_v\oplus\dots\oplus\{x_n\}_v\oplus z_1\oplus\dots\oplus z_m}$$

$$\Downarrow$$

$$(GX)\dfrac{(C_v)\dfrac{(GX)\dfrac{T\vdash x_1 \qquad \dots \qquad T\vdash x_n}{T\vdash x_1\oplus\dots\oplus x_n} \qquad T\vdash v}{T\vdash \{x_1\}_v\oplus\dots\oplus\{x_n\}_v} \quad (R_1)\dfrac{\vdots}{T\vdash z_1}\dots(R_m)\dfrac{\vdots}{T\vdash z_m}}{T\vdash \{x_1\}_v\oplus\dots\oplus\{x_n\}_v\oplus z_1\oplus\dots\oplus z_m}$$

Figure 4: Transformation of (C_v)-(GX) into (GX)-(C_v), all (R_i) are different of (C_v) and if $n\geq 2$

$$(D_v)\dfrac{(GX)\dfrac{(R_1)\dfrac{T\vdash B_1}{T\vdash B_1'} \quad \dots \quad (R_n)\dfrac{T\vdash B_n}{T\vdash B_n'} \quad (C_v)\dfrac{T\vdash B \qquad T\vdash v}{T\vdash \{B\}_v}}{T\vdash \{u\}_v} \quad T\vdash v}{T\vdash u}$$

$$\Downarrow$$

$$(D_v)\dfrac{(GX)\dfrac{(GX)\dfrac{(R_1)\dfrac{T\vdash B_1}{T\vdash B_1'} \quad \dots \quad (R_n)\dfrac{T\vdash B_n}{T\vdash B_n'}}{T\vdash \{c\}_v} \quad (C_v)\dfrac{T\vdash B \qquad T\vdash v}{T\vdash \{B\}_v}}{T\vdash \{u\}_v} \quad T\vdash v}{T\vdash u}$$

$$\Downarrow$$

$$(GX)\dfrac{(D_v)\dfrac{(GX)\dfrac{(R_1)\dfrac{T\vdash B_1}{T\vdash B_1'} \quad \dots \quad (R_n)\dfrac{T\vdash B_n}{T\vdash B_n'}}{T\vdash \{c\}_v} \quad T\vdash v}{T\vdash c} \quad T\vdash B}{T\vdash u=c\oplus B}$$

Figure 5: Elimination of a rule (D_v) after a (GX) with a rule (C_v) just above the (GX) (C_k)-(GX)-(D_k), with $n\geq 2$.

Proof: We denote by π_x the subproof of P with root $T \vdash x$. Observe first that all the transformations transform a proof with some hypotheses and a conclusion into a proof of the same hypotheses and the same conclusion.

In Figure 3 it is obvious.

In Figure 4 the number of nodes of the initial proof is $\Sigma_{i=1}^{i=m}|\pi_{z_i}|+\Sigma_{i=1}^{i=n}|\pi_{x_i}|+n|\pi_v|+n+1$ and the final proof contains $\Sigma_{i=1}^{i=m}|\pi_{z_i}|+\Sigma_{i=1}^{i=n}|\pi_{x_i}|+|\pi_v|+3$ nodes, which is less since $n \geq 2$.

In Figure 5 the first proof has $\Sigma_{i=1}^{i=n}|\pi_{B_i'}| + |\pi_B| + 2|\pi_v| + 3$ nodes and the last proof has $\Sigma_{i=1}^{i=n}|\pi_{B_i'}| + |\pi_B| + |\pi_v| + 3$ nodes. Hence, the number of nodes decreases. □

Lemma 5 *If there is a proof of $T \vdash w$ then there is also a $\oplus$-eager and simple proof of $T \vdash w$.*

Proof: Let P be a proof of $T \vdash w$. The transformation rules given in Figures 3, 4 and 5 decrease $|P|$ as well as the transformation to get a simple rule. Therefore the application of rules eventually terminates with an $\oplus$-*eager* simple proof of $T \vdash w$. □

6.3 Properties of Proofs in the *XDE* Case

First, we demonstrate a technical lemma used in the proof of Lemma 7. Then we prove Lemma 7 and Lemma 8 which show that (i) the premise (a pair) in the rules (UL-UR) belongs to $S(T)$ (ii) the encrypted term which is a premise of (D) belongs to $S_\oplus(T)$. These lemmata are similar to lemmata in [CLS03].

Lemma 6 *Let P be a simple proof of the form:*

$$P = \left\{ \frac{P_1 \ldots P_n}{T \vdash w} \right.$$

If $T \vdash u$ does not occur in any of $P_1, \ldots, P_n$ and $\langle u, v\rangle \in S(w)$ then there is at least one P_i and there exists w' such that $\langle u, v\rangle \in S(w')$ and either the root of P_i is $T \vdash w'$ or $w' \in T$.

Proof: The proof is detailed in Appendix B. □

Lemma 7 *Let P be a simple proof of $T \vdash u$ or $T \vdash v$. If P is one of*

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

then $\langle u, v\rangle \in S(T)$.

Proof: Let us assume that the last rule is (UL), the case (UR) is similar.

$$P = \left\{ \begin{array}{c} \dfrac{\dfrac{P_1 \ldots P_n}{T \vdash \langle u, v\rangle}}{T \vdash u} \end{array} \right.$$

P is simple so $T \vdash u$ does not occur in any of $P_1, \ldots, P_n$. Hence, we can apply Lemma 6 to $\dfrac{P_1 \ldots P_n}{T \vdash \langle u, v\rangle}$. Either $\langle u, v\rangle \in T$, or there is some P_i with root $T \vdash w$ such that $\langle u, v\rangle \in S(w)$ and $T \vdash u$ does not occur in P_i. Lemma 6 can be applied again and the iteration of this reasoning finally leads to $\langle u, v\rangle \in T$. □

Lemma 8 *Let P be a $\oplus$-eager and simple proof of $T \vdash u$. If P is*

$$(D)\dfrac{(R)\dfrac{\vdots}{T \vdash \{u\}_v\downarrow = r} \qquad \dfrac{\vdots}{T \vdash v\downarrow}}{T \vdash u}$$

then $\{u\}_v \in S_\oplus(T)$.

Proof: The proof is technical and detailed in Appendix B. □

6.4 Normality

In the rest of the paper, we precise $S(T)$-local proof instead of S-local, where T is the set of terms on which S is applied.

We define first a new kind of proof, the *normal* proofs. A normal proof consist of initial subproofs which are $S_\oplus(T)$-local, followed by a proof tree consisting only of the rule (GX), (C), and (P). We show that we can transform any proof into a normal proof.

Definition 8 *Let P be a proof of $T \vdash u$. P is a* normal proof *if :*

- *either $u \in S_\oplus(T)$ and P is an $S_\oplus(T)$-local proof,*
- *or $P = C[P_1, \ldots, P_n]$ where every proof P_i is a normal proof of some $T \vdash v_i$ with $v_i \in S_\oplus(T)$ and the context C is built using the inference rules (P), (C), (GX) only.*

Lemma 9 *If there is a simple and $\oplus$-eager proof of $T \vdash w$ then there is a normal proof of $T \vdash w$.*

Proof: Using Lemma 8 and Lemma 7 we can construct a normal proof from a simple and $\oplus$-*eager* one. The details of the proof are in Appendix B. □

We prove now that a normal proof is stable by the transformation rules used in the construction of $\oplus$-*eager* and simple proofs.

Lemma 10 *Let P be a normal proof of $T \vdash w$. Then the application of the simplification rule and of the proof transformations of Figure 3, 4 and 5 to P terminates and yields a normal, $\oplus$-eager and simple proof of $T \vdash w$.*

Proof: We show that the simplification rule and the transformation rules of figures 3, 4 and 5 transform a normal proof into a normal proof. We assume that the initial proofs are normal and we prove that the resulting proofs are still normal.

- Simplification rule: straightforward since no new term is constructed and since the order of applications of rules is preserved.
- Flattening rule Figure 3: straightforward.
- Rule of Figure 4: we know that the sub-proofs of $T \vdash x_i, T \vdash z_i$ and $T \vdash v$ are normal. Since the transformation only permutes applications of (GX) and (C_v) rules, the only case to consider is the preservation of $S_\oplus(T)$-locality. If $w = \{x_1\}_k \oplus \ldots \oplus \{x_n\}_k \oplus z_1 \oplus \ldots \oplus z_m \in S_\oplus(T)$ then all nodes of the transformed proof are in $S_\oplus(w)$ and hence belong to $S_\oplus(T)$. As a consequence, the transformation yields a normal proof.
- Rule of Figure 5: Since the sub-proof is normal and the last rule is (D_v) it is in fact an $S_\oplus(T)$-local sub-proof. In the result we have to prove that all nodes are in $S_\oplus(T)$. Note that the second step of the transformation all nodes are in $S_\oplus(T)$ by definition of $S_\oplus$. Since $\{c\}_v, v, A \in S_\oplus(T)$ we obtain that all nodes of the transformed proof are in $S_\oplus(T)$. Hence, the result of the transformation is also an $S_\oplus(T)$-local sub-proof.

Since all the transformations decrease $|P|$, the application of rules eventually terminates. □

Lemma 11 *Let P be a $\oplus$-eager and simple proof of $T \vdash w$ then*
P is normal if and only if P is $S_\oplus(T, w)$-local.

Proof: We show the two directions of the equivalence.

Direction $\Leftarrow$: Let us assume that P is $S_\oplus(T, w)$-local and prove that P is normal.

- If $w \in S_\oplus(T)$ then P is $S_\oplus(T)$-local *i.e.* P is normal.
- If $w \notin S_\oplus(T)$ then we proceed by structural induction on P. The base case is trivial, consider the last rule:
  - (UR), (UL), (D): impossible since Lemma 7 and Lemma 8 show that $w \in S_\oplus(T)$ which contradicts the hypothesis.
  - (P), (C), (GX): by induction hypothesis, the hypotheses w_i of the rule stem from normal proofs. Since the last rule is (P), (C) or (GX) the proof P is normal.

Direction $\Rightarrow$: Let us assume that P is normal and prove that P is $S_{\oplus}(T, w)$-local.

- $w \in S_{\oplus}(T)$: In this case, P is $S_{\oplus}(T)$-local, hence $S_{\oplus}(T, w)$-local.
- $w \notin S_{\oplus}(T)$: We proceed by structural induction on P. The base case is trivial. Consider the last rule of P:
  - (UR), (UL), (D): impossible by definition of normal proof.
  - (P), (C): these cases are similar, we just give the proof for (C). P is s.t. $\dfrac{T \vdash w_1 \quad T \vdash w_2}{T \vdash \{w_1\}_{w_2} = w}$. By definition for $i = 1, 2$ $w_i \in S_{\oplus}(T, w_i)$, $w_i \in S_T(\{w_1\}_{w_2}) = S_T(w) \subseteq S_{\oplus}(w)$, and the induction hypothesis which guarantees that all nodes of the sub-proofs are in $S_{\oplus}(T, w_i)$, we conclude that P is $S_{\oplus}(T, w)$-local.
  - (GX): P is s.t. $(GX)\dfrac{(R_1)\dfrac{T \vdash B_1}{T \vdash B'_1} \quad \ldots \quad (R_n)\dfrac{T \vdash B_n}{T \vdash B'_n}}{T \vdash w}$. We will prove that all B'_i are in $S_{\oplus}(T, w)$. Consider the different cases for the (R_i):

    (A): by definition, $B'_i \in S_{\oplus}(T)$.

    (UR), (UL), (D): by Lemma 7 or by Lemma 8 we get that $B'_i \in S_{\oplus}(T)$.

    (GX): impossible since P is $\oplus$-*eager*, which implies that P is flat.

    (P): if $B'_i \in S_{\oplus}(T)$ then the claim holds, otherwise $B'_i \notin S_{\oplus}(T)$. Either B'_i is not canceled in the $\oplus$, then $B'_i \in S_T(w) \subseteq S_{\oplus}(w)$, or otherwise B'_i is canceled by another element of the sum B'_j. Since B'_i is a pair B'_j can neither stem from a rule (C) nor from a rule (P) since P is simple. Hence, it stems from one of the rules (A), (UL), (UR) or (D) and $B'_i \in S_T(B'_j)$. According to Lemma 7 and Lemma 8 we have that $B'_j \in S_{\oplus}(T)$, hence the claim holds by transitivity of $S_{\oplus}$.

    (C_k): if $B'_i \in S_{\oplus}(T)$ then the claim holds immediately, otherwise $B'_i \notin S_{\oplus}(T)$. Note that B'_i can be partially canceled in the sum. There are two possibilities for the atoms of B'_i: to be present in w, in which case atoms$(B'_i) \in$ atoms$(S_T(w)) \subseteq$ atoms$(S_{\oplus}(w))$, or to be canceled by other elements B'_j of the sum, in which case atoms$(B'_i) \in$ atoms$(S_{\oplus}(B'_j)) \subseteq$ atoms$(S_{\oplus}(T))$. In the latter case, since B'_i is encrypted by the key k, B'_j can neither be the result of a rule (C_v) with $v \neq k$, nor the result of a rule(C_k) since P is $\oplus$-*eager*, nor (P), hence it stems from one of the rules (A), (UL), (UR) or (D). We conclude from Lemma 7 and Lemma 8 that $B'_j \in S_{\oplus}(T)$ by using the transitivity of $S_{\oplus}$. In summary, for all i we get that atoms$(B'_i) \in$ atoms$(S_{\oplus}(T, w))$, that is $B'_i \in S_{\oplus}(T, w)$. Hence, P is $S_{\oplus}(T, w)$-local.

□

6.5 Locality in the General Case

Theorem 3 *If there exists a proof of $T \vdash w$ then there exists an $S_{\oplus}(T, w)$-local proof of $T \vdash w$.*

Proof: Let P be a proof of $T \vdash w$. By Lemma 5 we can get a $\oplus$-*eager* simple proof of $T \vdash w$. By Lemma 9, we can get a normal proof of $T \vdash w$. By Lemma 10 we obtain a $\oplus$-*eager* simple and normal proof of $T \vdash w$. We can now apply Lemma 11 to prove that there is an $S_{\oplus}(T, w)$-local proof of $T \vdash w$. □

Corollary 12 *The problem of intruder deduction for the theory XDE is decidable in EXPTIME.*

Proof: With Theorem 1 of locality adapted from McAllester's algorithm, Theorem 2 stating that one-step deducibility is in PTIME, the fact that computing $S_{\oplus}(T, w)$ is in EXPTIME, and Theorem 3 which ensures $S_{\oplus}$-locality we obtain that the problem of intruder deduction for the theory XDE is decidable in EXPTIME. □

7 Intruder Deduction in the Binary Case

We call a term in normal form *top-binary* if it is the sum of two different terms not headed with $\oplus$, and *at most binary* if all its syntactic subterms are either top-binary or not headed with $\oplus$. A set is *at most binary* if each of its elements is. A proof tree P is called *at most binary* if for all its nodes $T \vdash u$ the term u is at most binary.

We call two terms t_1 and t_2 in normal form *disjoint* if $\text{atoms}(t_1) \cap \text{atoms}(t_2) = \emptyset$.

Our goal is to give a polynomial algorithm for the intruder deduction problem when the set of hypotheses and the conclusion are at most binary. Existence of such an algorithm will be assured by a locality result for some subterm function which, when applied to at most binary terms, yields only at most binary terms. This requires a new proof normalization since, as the following example shows, a $\oplus$-*eager*-proof may involve terms with more than two atoms even if the hypotheses and the conclusion are at most binary.

Example 7 *The following proof of $\{b\}_k \oplus \{c\}_k$ with $T = \{k, a \oplus b, d \oplus c, \{d\}_k \oplus \{a\}_k\}$ is simple and $\oplus$-eager but not binary.*

$$(GX)\dfrac{(C)\dfrac{(A)\dfrac{k \in T}{T \vdash k}\quad (GX)\dfrac{(A)\dfrac{a \oplus b \in T}{T \vdash a \oplus b}\quad (A)\dfrac{c \oplus d \in T}{T \vdash c \oplus d}}{T \vdash a \oplus b \oplus c \oplus d}}{T \vdash \{a\}_k \oplus \{b\}_k \oplus \{c\}_k \oplus \{d\}_k}\quad (A)\dfrac{\{a\}_k \oplus \{d\}_k \in T}{T \vdash \{a\}_k \oplus \{d\}_k}}{T \vdash \{b\}_k \oplus \{c\}_k}$$

7.1 A Variant of the Proof System

Definition 9 *Let, for any set T of terms in normal form,*

$$S_{\oplus 2}(T) = S_T(T) \cup \{a_1 \oplus a_2 \mid a_1, a_2 \in S_T(T), a_1 \neq a_2\}$$

Obviously we have that $S_T(T) \subseteq S_{\oplus 2}(T) \subseteq S_{\oplus}(T)$ for any set T.

Our goal is to prove that a variant of the proof system, which is equivalent in deductive power to the original one, is $S_{\oplus 2}$-local. The difference between the original proof system and the new one is that certain proof-trees consisting of the rules (C), (GX) and (D) are collapsed into one instance of a new deduction rule called (GCD). We first define the general form of a GCD-proof tree which we will later slice into several instances of the (GCD) rule.

Definition 10 *A proof tree P is a* GCD-proof tree *with set of leaves L, set of keys K, and root u in any of the following cases:*

1. *P consists of a single node $T \vdash u$ and $L = \{u\}$, $K = \emptyset$,*
2. *or P is of the form (C) $\dfrac{P \qquad T \vdash k}{T \vdash u}$ where P is a GCD proof tree with root u', leaves L and set of keys K', $K = K' \cup \{k\}$, and $\{u'\}_k \downarrow = u$,*
3. *or P is of the form (D) $\dfrac{P \qquad T \vdash k}{T \vdash u}$ where P is a GCD proof tree with root u', leaves L and set of keys K', $K = K' \cup \{k\}$, and $\{u\}_k \downarrow = u'$,*
4. *or P consists of (GX) $\dfrac{P_1 \ \cdots \ P_n}{u}$ with $n \geq 1$ such that every P_i is a GCD-proof tree with respective leaves L_i, root u_1, and set of keys K_i, and $K = \bigcup_{i=1}^{n} K_i \cup K'$ and $L = \bigcup_{i=1}^{n} L_i$.*

In particular, any instance of one of the rules (GX), (C), or (D) is a GCD-proof tree. We can hence imagine any proof tree as composed of of the rules (A), (UL), (UR), (P), and of GCD-proof trees (remark that we shall use non-flat proofs).

7.2 Locality in the binary case

Definition 11 *A term v is in* key position *of a normal term w if $\{t\}_v \in S(w)$ for some term t.*

Lemma 13 *Let P be a $\oplus$-eager and simple proof of $T \vdash w$. All nodes which are hypothesis of a rule (A), (UR), (UL), (P) or which are in key position in a node of P are in $S(T, w)$.*

Proof: According to Lemma 7 or the definition of T if a node is an hypothesis of a rule (A), (UR), (UL) then this node is in $S(T)$.

If a node is the conclusion of the rule (P) then it is a pair, by Theorem 3 applying on P a simple and $\oplus$-*eager* proof of $T \vdash w$, the pair is in $S_{\oplus}(T, w)$. By definition of $S_\oplus$ the only possibility to get a pair in $S_\oplus(T, w)$ is that the pair is in $S(T, w)$. By consequence the hypothesis of the rule (P) are in $S(T, w)$.

Now consider the last case, a node $T \vdash v$ of P where v is in key position in a node $\{u\}_v$ of P. The result of locality on P gives that $\{u\}_v \in S_\oplus(T, w)$. The construction of $S_\oplus$ builds all possible sums of elements of $S_T(T, w)$. It does not add any applications of the encryption symbol. Hence $\{u\}_v \in S_T(T, w)$, since $v \in S(\{u\}_v)$, we obtain that $v \in S_T(T, w)$. By the definition of the set $S_T(T, w)$ if v is a term in key position and $v \in S_T(T, w)$ then $v \in S(T, w)$. □

The following proposition is the key to obtaining $S_{\oplus 2}$-locality. We recall that two terms are called *disjoint* if their respective sets of atoms are disjoint.

Proposition 5 *Let U be a finite set of at most binary terms and $u = \bigoplus U\downarrow$. There exist pairwise disjoint sets of terms $U_1, \ldots, U_k \subseteq U$ such that:*

1. *$\bigoplus U_i\downarrow$ is at most binary for $1 \le i \le k$.*
2. *$\bigoplus U_i\downarrow$ and $\bigoplus U_j\downarrow$ are disjoint for $i \neq j$.*
3. *$\bigoplus_{i=1}^k (\bigoplus U_i\downarrow) = u$.*

Note that, since all the $\bigoplus U_i\downarrow$ are pairwise disjoint, we have that

$$(\bigoplus_{i=1}^{k}(\bigoplus U_i\downarrow))\downarrow = \bigoplus_{i=1}^{k}(\bigoplus U_i\downarrow)$$

Proof: The proof is by induction on the number of elements in U. If $u = 0$ then we choose $k = 0$. Otherwise U is non-empty. Let $a \in \mathrm{atoms}(u)$. Then there exists a (at most binary) term $u_0 \in U$ such that $a \in \mathrm{atoms}(u_0)$. Application of the induction hypothesis to $U' = U \setminus \{u_0\}$ yields a decomposition into l pairwise disjoint sets $U'_1, \ldots, U'_l \subseteq U'$. Let $u' = \bigoplus U'\downarrow$. We have that $u = (u' \oplus u_0)\downarrow$.

1. If u_0 is not headed with $\oplus$ then, since u_0 appears in $\mathrm{atoms}(u)$, it can not appear in $\mathrm{atoms}(u')$. We choose $k = l + 1$, $U_i = U'_i$ for $i < k$, and $U_k = \{u_0\}$.
2. If u_0 is binary with $\mathrm{atoms}(u_0) \cap \mathrm{atoms}(u') = \emptyset$ then $u = u' \oplus u_0$. We choose $k = l + 1$, $U_i = U'_i$ for $i < k$, and $U_k = \{u_0\}$.
3. If u_0 is binary, say $u_0 = a \oplus b$, and $b \in \mathrm{atoms}(u')$ then there exists a term u_1 and some set U'_i such that $b \in \mathrm{atoms}(u_1)$ and $u_1 \in U'_i$. We choose $k = l$, $U_i = U'_i \cup \{u_0\}$, and $U_j = U'_j$ for $j \neq i$.

□

Lemma 14 *Let P be a GCD-proof tree with leaves L, set of keys K, and root r. If L and r are at most binary then there exists an at most binary GCD-proof tree P' with leaves $L' \subseteq L$, keys $K' \subseteq K$, and root r.*

Proof: First note that for any instance of a rule (C) or (D), which can be seen as special cases of GCD-proof trees, the root is at most binary if and only if the leaf is at most binary. Hence, if all instances of (GX) in the proof tree P have an at most binary result then P is at most binary.

Otherwise, there exists an instance of (GX) whose result is not at most binary and where all the leaves are at most binary. Since the root of P is at most binary, the path from the root of the instance of (GX) to the root of P eventually leads to another instance of the (GX) rule. That is, we have a proof tree of the following form

$$\text{(GX)}\ \dfrac{\begin{array}{c}\text{(GX)}\ \dfrac{T\vdash u_1 \qquad \cdots \qquad T\vdash u_n}{\text{(C,D)}\ \dfrac{T\vdash u \quad (= (u_1\oplus\cdots\oplus u_n)\downarrow)}{\vdots}} \\ \text{(C,D)}\ \dfrac{}{T\vdash u'}\end{array} \qquad P_1 \qquad \cdots \qquad P_n}{P}$$

where (C,D) is any instance of a rule (C) or (D), and where the keys are omitted for the sake of clarity. By Proposition 5 there are pairwise disjoint sets $U_1,\ldots,U_k \subseteq \{u_1,\ldots,u_n\}$ such that the normal forms of their respective sums are at most binary and pairwise disjoint, and such that $\bigoplus_i(\bigoplus U_i\downarrow) = u = (u_1\oplus\ldots\oplus u_n)\downarrow$. We hence obtain, where we abbreviate by $T\vdash U_i$ the set of sequents $\{T\vdash z \mid z\in U\}$:

$$\text{(GX)}\ \dfrac{\begin{array}{c}\text{(GX)}\ \dfrac{\text{(GX)}\ \dfrac{T\vdash U_1}{T\vdash v_1} \qquad \cdots \qquad \text{(GX)}\ \dfrac{T\vdash U_n}{T\vdash v_n}}{\text{(C,D)}\ \dfrac{T\vdash u \quad (= v_1\oplus\ldots\oplus v_n)}{\vdots}} \\ \text{(C,D)}\ \dfrac{}{T\vdash u'}\end{array} \qquad P_1 \qquad \cdots \qquad P_n}{P}$$

In this proof tree we hence have that $v_i = (\bigoplus U_i)\downarrow$ for every i. We can now apply the inverse transformation of Figure 4 and commute the (GX) rule with succeeding (C) rules. Since the premises of the newly introduced instance of (GX) are all disjoint we can also commute this (GX) rule with succeeding (D) rules. We hence obtain:

$$\text{(GX)}\ \dfrac{\begin{array}{c}\text{(GX)}\ \dfrac{T\vdash U_1}{\text{(C,D)}\ \dfrac{T\vdash v_1}{\vdots}} \\ \text{(C,D)}\ \dfrac{}{T\vdash u_1'}\end{array} \qquad \begin{array}{c}\text{(GX)}\ \dfrac{T\vdash U_n}{\text{(C,D)}\ \dfrac{T\vdash v_n}{\vdots}} \\ \text{(C,D)}\ \dfrac{}{T\vdash u_n'}\end{array} \qquad P_1 \qquad \cdots \qquad P_n}{P}$$

where $u' = u_1'\oplus\ldots\oplus u_n'$. We may now apply the induction hypothesis to this proof tree since the number of instances of (GX) with a non at most binary result has decreased by one. □

We can now define the (GCD) proof rule: An instance of this rule is a particular form of a GCD-proof tree.

Definition 12 *The rule (GCD) consists of all GCD-proof trees with exactly one instance of (GX), where all instances of (C) are above the (GX) rule, and all instances of (D) are below the (GX) rule.*

Lemma 15 *Let P be a simple and $\oplus$-eager GCD-proof tree with leaves L, keys K, and root r. If $L \cup \{r\} \in S_{\oplus 2}(T)$ for some set of terms T then there exists a proof tree using exclusively the (GCD) rule such that all nodes are in $S_{\oplus 2}(T)$.*

Proof: Let P' be the at most binary proof tree obtained from P by the transformation of the proof of Lemma 14. We may assume w.l.o.g. that P' is simple, hence the only possible sequence of rule applications between two consecutive (GX) rules is some applications of (D), followed by some applications of (C). The "frontier" between two instances of the rule (GCD) is at the end of the sequence of (D) rule applications. Since $u \in S_{\oplus 2}(T)$ by hypothesis, we also have that $v_i \in S_{\oplus 2}(T)$ since the v_i are pairwise disjoint. As a consequence, any term obtained by a sequence of decryptions form v_i is also in $S_{\oplus 2}(T)$. □

Theorem 4 *If T, w are at most binary then $T \vdash w$ is derivable if and only if there exists an $S_{\oplus 2}(T, w)$-local proof of $T \vdash w$ using rules (A),(UL), (UR), (P), and (GCD) only.*

Proof: If there is a proof of $T \vdash w$. By Theorem 3 there is a $S_{\oplus}$-local proof of $T \vdash w$ which is simple and $\oplus$-*eager*. By Lemma 13 all nodes which are hypotheses or conclusion of one of (A), (UL), (UR) or (P) are in $S_T(T, w) \subseteq S_{\oplus 2} = (T, w)$. By Lemma 15 we can transform any GCD-proof tree into a proof tree using the (GCD) rule only and where all nodes are in $S_{\oplus 2}(T, w)$. □

7.3 Deciding One-step deducibility by the (GCD) rule

In this section we will use an abbreviation for sequences of encryptions and write $\{m\}_{x_1 \cdots x_n}$ for $\{\cdots \{m\}_{x_1} \cdots\}_{x_n}$.

We are now faced with the problem of checking whether, for a given set U of at most binary terms and an at most binary term r there is an instance of rule (GCD) with leaves and keys contained in U and root r. There are three possible cases to consider:

1. r is not headed with $\oplus$, and there is a sequence of top-binary terms $(\{a_i\}_{v_i} \oplus \{b_i\}_{v_i})_{i=1,\dots,N}$ in U, a term $a_{N+1} \in U$ not headed with $\oplus$, and a sequence $(h_i)_{i=0,\dots,N+1}$ of words in U^* such that $\{r\}_{h_0} = \{a_1\}_{v_1 h_1}$ and $\{b_i\}_{v_i h_i} = \{a_{i+1}\}_{v_{i+1} h_{i+1}}$ for $i = 1, \dots, N$.
2. r is a top-binary term $r_1 \oplus r_2$, and there are two instances of the rule (GCD) as in the first case with roots r_1, resp. r_2, and with the same sequence of keys h_0.

3. r is a top-binary term $r_1 \oplus r_2$, and there is a sequence of top-binary terms $(\{a_i\}_{v_i} \oplus \{b_i\}_{v_i})_{i=1,\ldots,N}$ in U and a sequence $(h_i)_{i=0,\ldots,N}$ of words in U^* such that $\{r_1\}_{h_0} = \{a_1\}_{v_1 h_1}$, $\{b_i\}_{v_i h_i} = \{a_{i+1}\}_{v_{i+1} h_{i+1}}$ for $i = 1, \ldots, N-1$, and $\{b_N\}_{v_N h_N} = \{r_2\}_{h_0}$.

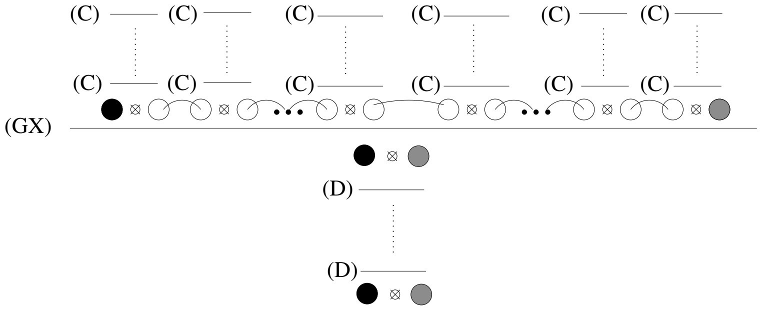

Figure 6: Illustration of the third case

In the following we will only consider the last case, which is illustrated by Figure 6, since the first two cases can be checked in a very similar way.

The idea is to reduce the problem to reachability in a prefix rewrite system [Cau92]. Let us first explain the construction at hand of a special case. We view a term $\{a\}_x$, where a is not headed with $\oplus$ and not of the form $\{m\}_k$, as the term ax. That is, the string representation consists of a constant denoting the message, followed by the sequence of keys from the innermost to the outermost encryption. Alternatively, this can be seen as a configuration of a pushdown process with state a and stack x, where the innermost encryption key is on top of the stack.

If we ignore for the moment possible instances of the rule (D), and if we assume for the moment that all terms in key positions of terms in U are also contained in U then we can just construct the prefix rewrite system which allows, for any binary term $\{a\}_v \oplus \{b\}_w \in L$, to rewrite any term avx into bwx, and vice versa:

$$\{av \to bw \mid \{a\}_v \oplus \{b\}_w \in U \text{ or } \{b\}_w \oplus \{a\}_v \in U\}$$

If we wish to check for an instance of the rule (GCD) with root $\{a\}_v \oplus \{b\}_w$ then we just have to test whether the string av rewrites to the string bw in this prefix rewrite system.

The first difficulty is that some of the keys may not be contained in U. In this case we may rewrite avx into bwx only if $x \in U^*$. We can implement this check, in terms of a pushdown process, by maintaining a marker symbol # on the stack which is always at the topmost position such that all symbols below # are in U. Formally, let *left*(x) and *right*(x), for any string x, be such that $x =$ *left*$(x) \cdot$ *right*(x), and such that *right*(x) is the maximal suffix of x which lies in U^*. Then we construct the rewrite system as follows, in order to assure that all redexes comprise, or are adjacent to the marker #:

$$\begin{aligned}\{\quad & a\ \mathit{left}(v)\#\mathit{right}(v) \to b\ \mathit{left}(w)\#\mathit{right}(w) \mid \\ & \qquad \{a\}_v \oplus \{b\}_w \in U \text{ or } \{b\}_w \oplus \{a\}_v \in U \quad \}\end{aligned}$$

Finally, it may be possible that the result of the (GCD) rule is only obtained after some sequence of decryptions from the result of the (GX) rule. We hence cut now the rewrite process in two consecutive processes. During the first process, if we have a stack x and wish to apply a rewrite rule the left-hand side of which contains x as a proper prefix then we just put the missing symbols with a negative sign on the stack. In the second process we do the reverse action, that is if some negative symbols are on the top of the stack and if the right hand side of the rewrite rule produces these symbols, then we just pop these negative symbols from the stack. We denote the negation of a symbol a as $\overline{a}$. The states of the second process are decorated with a hat in order to keep the two state spaces disjoint. We denote by $\overline{x}$ for any $x = x_1 \cdots x_n$ the string $\overline{x_n} \cdots \overline{x_1}$ (note the inversion of the order). The symbol $\perp$ is used to denote the right end of a string (i.e., the bottom of a stack).

Definition 13 *We define $\mathit{sta}(\{t\}_k) = \mathit{sta}(t)$, $\mathit{sta}(t) = \bigcup_{a \in \mathit{atoms}(t)} \mathit{sta}(a)$, and $\mathit{sta}(t) = \{t\}$ if t is not headed with $\oplus$ and not of the form $\{x\}_y$.*

We define $\mathit{keys}(\{t\}_k) = \mathit{keys}(t) \cup \{k\}$, $\mathit{keys}(t) = \bigcup_{a \in \mathit{atoms}(t)} \mathit{keys}(a)$, and $\mathit{keys}(t) = \{\}$ if t is not headed with $\oplus$ and not of the form $\{x\}_y$.

For a set T of terms we define $\mathit{sta}(T) = \bigcup_{t \in T} \mathit{sta}(t)$ and $\mathit{keys}(T) = \bigcup_{t \in T} \mathit{keys}(t)$.

Example 8 *Let $T = \{a\}_{bc} \oplus \{d\}_e \oplus \{d\}_{ce}$, then $\mathit{sta}(T) = \{a, d\}$ and $\mathit{keys}(T) = \{b, c, e\}$.*

We define, for given set U of at most binary terms and an at most binary term r two prefix rewrite systems. Let $Q = \mathit{sta}(U, r)$ and $C = \mathit{keys}(U, r)$.

1. The prefix rewrite system PR_1 is defined by the following rules:

$$\begin{aligned}\{\quad & a\ \mathit{left}(v)\#\mathit{right}(v) \to b\ \mathit{left}(w)\#\mathit{right}(w) \\ & a\ \mathit{left}(v)\#v_1\gamma \to b\ \mathit{left}(w)\#\mathit{right}(w)\overline{v_2}\gamma \mid \\ & \qquad \{a\}_v \oplus \{b\}_w \in U \text{ or } \{b\}_w \oplus \{a\}_v \in U, \\ & \qquad v_1 v_2 = \mathit{right}(v), \\ & \qquad \gamma \in \{\perp\} \cup \{\overline{u} \mid u \in U\} \ \}\end{aligned}$$

2. The prefix rewrite system PR_2 is defined by the following rules:

$$\begin{aligned}\{\quad & \hat{a}\ \mathit{left}(v)\#\mathit{right}(v) \to \hat{b}\ \mathit{left}(w)\#\mathit{right}(w) \\ & \hat{a}\ \mathit{left}(v)\#\mathit{right}(v)\overline{w_2} \to \hat{b}\ \mathit{left}(w)\#w_1 \mid \\ & \qquad \{a\}_v \oplus \{b\}_w \in U \text{ or } \{b\}_w \oplus \{a\}_v \in U, \\ & \qquad w_1 w_2 = \mathit{right}(w) \quad \}\end{aligned}$$

These two rewrite systems are symmetric one to the other with the technical exception that the symbol γ in the system PR_1 serves to ensure the invariant that no negative symbol occurs to the left of a non-negative symbol. The system PR_2 maintains this invariant since it can not push negative symbols.

The following two lemmata state the central property of each of these two prefix rewrite systems:

Lemma 16 *The following two assertions are equivalent for every $a, b \in Q$, $x_1, y_1 \in \{\epsilon\} \cup C^*(C \setminus U)$, $x_2, y_2, y_3 \in U^*$:*

1. *There is a prefix rewrite sequence by PR_1*

$$ax_1\#x_2\bot \mapsto^* by_1\#y_2\overline{y_3}\bot$$

2. *either $a = b$, $x_1 = y_1$, $x_2 = y_2$, $y_3 = \epsilon$,*
or there exists a sequence of binary terms $\{a_i\}_{v_i} \oplus \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, and a sequence of strings $h_i \in U^*$, $i = 1, \ldots, n$, such that

   (a) $\{a\}_{x_1x_2y_3} = \{a_1\}_{v_1h_1}$

   (b) $\{b_i\}_{w_ih_i} = \{a_{i+1}\}_{v_{i+1}h_{i+1}}$ *for* $i = 1, \ldots, n-1$

   (c) $\{b_n\}_{w_nh_n} = \{b\}_{y_1y_2}$

   and such that for some i the longest common suffix of y_3 and h_i is ϵ.

Lemma 17 *The following two assertions are equivalent for every $a, b \in Q$, $x_1, y_1 \in \{\epsilon\} \cup C^*(C \setminus U)$, $x_2, y_2, y_3 \in U^*$:*

1. *There is a prefix rewrite sequence by PR_2*

$$\hat{a}x_1\#x_2\overline{x_3}\bot \mapsto^* \hat{b}y_1\#y_2\bot$$

2. *either $a = b$, $x_1 = y_1$, $x_2 = y_2$, $y_3 = \epsilon$,*
or there exists a sequence of binary terms $\{a_i\}_{v_i} \oplus \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, and a sequence of strings $h_i \in U^*$, $i = 1, \ldots, n$, such that

   (a) $\{a\}_{x_1x_2} = \{a_1\}_{v_1h_1}$

   (b) $\{b_i\}_{w_ih_i} = \{a_{i+1}\}_{v_{i+1}h_{i+1}}$ *for* $i = 1, \ldots, n-1$

   (c) $\{b_n\}_{w_nh_n} = \{b\}_{y_1y_2x_3}$

   and such that for some i the longest common suffix of y_3 and h_i is ϵ.

The proof of these two lemmata can be found in the appendix, Section C.

We can finally define the complete rewrite system as consisting of the following rules:

$$PR_1 \cup PR_2 \cup \{a \to \hat{a} \mid a \in Q\}$$

Hence, if t and s are both not of the form $\{m\}_k$ then there is a proof of $T \vdash \{t\}_v \oplus \{s\}_w$ if and only if for some u, x_1, x_2, x_3:

$$t\ \mathit{left}(v)\#\mathit{right}(v)\bot \mapsto^* u\ x_1\#x_2\overline{x_3} \mapsto \hat{u}\ x_1\#x_2\overline{x_3} \mapsto^* \hat{s}\ \mathit{left}(w)\#\mathit{right}(w)\bot$$

Lemma 18 *Let L be a set of at most binary terms, K a sets of terms, and r an at most binary term. It is decidable in polynomial time whether there exists an instance of the (GCD) rule with leaves L, keys K, and root r.*

Proof: By Lemmata 16 and 17, checking an instance of (GCD) reduces to a reachability problem in a prefix rewrite system of polynomial size (note that we may w.l.o.g. exclude instances of (GCD) where all hypotheses of (GX) are obtained by some (C_v) and where there is (D_v) immediately below the (GX)). This can be done in polynomial time [Cau92]. □

As a consequence we obtain:

Theorem 5 *The binary intruder deduction problem for the equational theory XDE is decidable in polynomial time.*

8 Conclusion

Related works. The use of locality in the analysis of cryptographic protocols has been used first in [RT01], and later on by [CLS03, CKRT03]. In [LLT05], we studied the case of a homomorphic operator that distributes over some binary operation $\oplus$ which can be one of a the free associative-commutative operator, the *exclusive-or* operator, or the addition of an Abelian group. The EXPTIME result that we obtained for the intruder deduction problem for the theory of *exclusive-or* and a homomorphism has been strengthened in [Del05] to get a PTIME decision procedure by means of the resolution of polynomial equations in $\mathbb{Z}/2\mathbb{Z}[X]$. There are two main differences with the present work: First, the homomorphism is an isolated operation not related to the encryption operation, which is less realistic than our model. Second, the polynomial complexity obtained in [Del05] relies on the fact that there is only a fixed number of homomorphisms, while our case can be seen as the one of an infinite family of homomorphisms (one for every possible key). Even in light of a locality result, which implies that only the keys occurring in the goal term or in the set of hypotheses are relevant for a proof, the number of homomorphisms still depends in our case on the problem instance.

A main step of our approach in the binary case uses an idea which is similar to the one used in [Del05]: regroup certain combinations of "$\oplus$-constructions", encryptions, and (in our case) decryptions into one "macro" rule, the instances of which are then decided by an ad-hoc method.

Further work. The first main issue raised by our result is to extend this framework to the case of a commutative encryption, i.e. $\{\{x\}_y\}_z = \{\{x\}_z\}_y$. A preliminary work in this direction suggests that the same approach can be used successfully, but that a lower EXPSPACE bound could be established in case of non-symmetric keys, i.e. when there is an explicit operation I to compute the inverse of a key such that a term $\{x\}_y$ can be decrypted only if one knows $I(y)$.

The second main issue that is still not solved is the extension to the case of an active intruder. Although it seems that the problem is decidable for the case of a homomorphic operation which is not the encryption, the extension to our framework seems quite difficult.

References

[AG99] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, January 1999.

[AR00] Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *Proc. 1st IFIP International Conference on Theoretical Computer Science (IFIP–TCS)*, volume 1872 of *Lecture Notes in Computer Science*, pages 3–22. Springer-Verlag, 2000.

[BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[Cau92] Didier Caucal. On the regular structure of prefix rewriting. *Theoretical Computer Science*, 106(1):61–86, 1992.

[CDL05] Véronique Cortier, Stéphanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 2005. To appear.

[CJ97] J. Clark and J. Jacob. A survey of authentication protocol literature. `http://www.cs.york.ac.uk/~jac/papers/drareviewps.ps`, 1997.

[CKRT03] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with XOR. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 261–270, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.

[CLS03] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 271–280, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.

[CLT03] Hubert Comon-Lundh and Ralf Treinen. Easy intruder deductions. In Nachum Dershowitz, editor, *Verification: Theory & Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday*, volume 2772 of *Lecture Notes in Computer Science*, pages 225–242. Springer-Verlag, 2003.

[CR05] Yannick Chevalier and Michaël Rusinowitch. Combining intruder theories. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 639–651. Springer, 2005.

[Del05] Stéphanie Delaune. Easy intruder deduction problems with homomorphisms. Research Report LSV-05-10, Laboratoire Spécification et Vérification, ENS Cachan, France, July 2005. `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2005-10.pdf`.

[DJ90] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B - Formal Models and Semantics, chapter 6, pages 243–320. Elsevier Science Publishers and The MIT Press, 1990.

[DY83] D. Dolev and A.C. Yao. On the security of public-key protocols. In *Transactions on Information Theory*, volume 29, pages 198–208. IEEE Computer Society Press, March 1983.

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

[KKS87] E. Kaltofen, M. S. Krishnamoorthy, and B. D. Saunders. Fast parallel computation of hermite and smith forms of polynomial matrices. *SIAM J. Algebraic Discrete Methods*, 8(4):683–690, 1987.

[LLT04] Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Intruder deduction for *AC*-like equational theories with homomorphisms. Research Report LSV-04-16, LSV, ENS de Cachan, November 2004. Available at `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rapports-year-2004-list.php`.

[LLT05] Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Intruder deduction for AC-like equational theories with homomorphisms. In Jürgen Giesl, editor, *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05)*, volume 3467 of *Lecture Notes in Computer Science*, pages 308–322, Nara, Japan, April 2005. Springer-Verlag.

[Low95] G. Lowe. An attack on the Needham-Schroeder public key authentication protocol. *Information Processing Letters*, 56(3):131–133, November 1995.

[McA93] David A. McAllester. Automatic recognition of tractability in inference relations. *Journal of the ACM*, 40(2):284–303, April 1993.

[Nar96] Paliath Narendran. Solving linear equations over polynomial semirings. In *Proc. of 11th Annual Symposium on Logic in Computer Science (LICS'96)*, pages 466–472, July 1996.

[RT01] M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proc. 14th Computer Security Foundations Workshop (CSFW'01)*, pages 174–190, Cape Breton (Canada), 2001. IEEE Comp. Soc. Press.

A Proof for Section 5

In the following we use the notation ct for an integer constant $c \in \{0, 1\}$ and term t, defined by $0t = 0$ and $1t = t$.

Lemma 19 *Let $T_S = \{t_1, \ldots, t_n\}$ and let w_S be such that for $1 \leq i \leq n$, $t_i = c_{1,i} * a_1 \oplus \ldots \oplus c_{m,i} * a_m$ and $w_S = d_1 * a_1 \oplus \ldots \oplus d_m * a_m$ where $\{a_1, \ldots, a_m\}$ is the set of atoms of T_S, w_S. Let S be the following system of equations:*

$$\left\{ \begin{array}{rcl} c_{1,1}x_1 + \ldots + c_{1,n}x_n & = & d_1 \\ \vdots & \vdots & \vdots \\ c_{m,1}x_1 + \ldots + c_{m,n}x_n & = & d_m \end{array} \right.$$

Then (S) is satisfiable if and only if w_S is deducible from T_S with exactly one instance of the rule (GX).

Proof:

- If (S) is satisfiable then there exists a solution of (S) α such that:

$$\left\{ \begin{array}{rcl} c_{1,1}\alpha(x_1) + \ldots + c_{1,n}\alpha(x_n) & = & d_1 \\ \vdots & \vdots & \vdots \\ c_{m,1}\alpha(x_1) + \ldots + c_{m,n}\alpha(x_n) & = & d_m \end{array} \right.$$

  Hence, we compute w_S from T_S and α:

$$\begin{array}{rcl} \alpha(x_1) * t_1 \oplus \ldots \oplus \alpha(x_n) * t_n & = & \alpha(x_1) * (c_{1,1} * a_1 \oplus \ldots \oplus c_{m,1} * a_m) \\ & & \oplus \ldots \oplus \\ & & \alpha(x_n) * (c_{1,n} * a_1 \oplus \ldots \oplus c_{m,n} * a_m) \\ & = & c_{1,1} * \alpha(x_1) * a_1 \oplus \ldots \oplus c_{1,n} * \alpha(x_n) * a_1 \\ & & \oplus \ldots \oplus \\ & & c_{m,1} * \alpha(x_1) * a_m \oplus \ldots \oplus c_{m,n} * \alpha(x_n) * a_m \\ & = & d_1 * a_1 \oplus \ldots \oplus d_m * a_m = w_S \end{array}$$

- Let P be a proof of $T_S \vdash w_S$, using only (A)(GX). We construct the system (S) from T_S and w_S.

$$T_S = \{t_1, \ldots, t_n\}, \Sigma = \{a_1, \ldots, a_m\} = \text{atoms}(T_S, w_S)$$
$$w_S = d_1 * a_1 \oplus \ldots \oplus d_m a_m$$

  For all i, $1 \leq i \leq n$ there exist $c_{j,i}$ for $1 \leq j \leq m$ such that

$$t_i = c_{1,i} * a_1 \oplus \ldots \oplus c_{m,i} * a_m$$

  We can deduce w_S from T_S, there exist a decomposition of d_i into $c_{i,j}$, hence we obtain the following system:

$$\left\{ \begin{array}{rcl} c_{1,1}x_1 + \ldots + c_{1,n}x_n & = & d_1 \\ \vdots & \vdots & \vdots \\ c_{m,1}x_1 + \ldots + c_{m,n}x_n & = & d_m \end{array} \right.$$

□

B Proofs for Section 6

Lemma 20 *Let P be a simple proof of the form:*

$$P = \left\{ \begin{array}{c} P_1 \ldots P_n \\ \hline T \vdash w \end{array} \right.$$

If $T \vdash u$ does not occur in any of $P_1, \ldots, P_n$ and $\langle u, v \rangle \in S(w)$ then there is at least one P_i and there exists w' such that $\langle u, v \rangle \in S(w')$ and either the root of P_i is $T \vdash w'$ or $w' \in T$.

Proof: We consider all possible rules for the root of P:

- The last rule is (A): obvious since all elements of T are normalized.
- The last rule is (UL) or (UR): $\langle u, v \rangle \in S(w)$ by hypothesis, we denote $w' = \langle u_1, u_2 \rangle$ and by construction $w \in S(\langle u_1, u_2 \rangle)$. We deduce by transitivity of the subterm relation that $\langle u, v \rangle \in S(w')$ and conclude with the induction hypothesis.
- The last rule is (D): $\langle u, v \rangle \in S(w)$ by hypothesis, we denote $w' = \{u_1\}_{u_2}$ and by construction $w \in S(\{u_1\}_{u_2})$. We deduce by transitivity of the subterm relation that $\langle u, v \rangle \in S(w')$ and conclude with the induction hypothesis.
- The last rule is (GX): $\langle u, v \rangle \in S(w)$ by hypothesis and $w = (u_1 \oplus \ldots \oplus u_n)\downarrow$. Hence by definition of the subterm relation $\langle u, v \rangle \in \cup_i S(u_i)$, more precisely there exists i such that $\langle u, v \rangle \in S(u_i)$, because $\langle u, v \rangle$ is not headed with $\oplus$ and conclude with the induction hypothesis.
- The last rule is (P): since $T \vdash u$ can not occur in P we have that $w = \langle w_1, w_2 \rangle \neq \langle u, v \rangle$. But $\langle u, v \rangle \in S(w)$ by hypothesis so $\langle u, v \rangle \in S(\langle w_1, w_2 \rangle)$. It is a subterm of w_1 or of w_2 and we conclude with the induction hypothesis.
- The last rule is (C): We have that $w = \{w_1\}_{w_2} \neq \langle u, v \rangle$. But $\langle u, v \rangle \in S(w)$ by hypothesis so $\langle u, v \rangle \in S(\{w_1\}_{w_2})$. It is a subterm of w_1 or of w_2 and we conclude with the induction hypothesis. □

Lemma 21 *Let P be a $\oplus$-eager and simple proof of $T \vdash u$. If P is*

$$(D)\dfrac{(R)\dfrac{\vdots}{T \vdash \{u\}_v\downarrow = r} \qquad \dfrac{\vdots}{T \vdash v\downarrow}}{T \vdash u}$$

then $\{u\}_v\downarrow \in S_\oplus(T)$.

Proof: The proof is by structural induction on P.

Base case: obvious.

Induction step: we perform a case analysis on the last rule (R) used in the subproof of P with root $\{u\}_v\downarrow$

- (R) is (A), (UL), (UR): the result is true by definition (rule (A)) or Lemma 7 (rule (UL), (UR)).

- (R) is some rule (P): this cannot happen since $\{u\}_v\downarrow$ is not a pair.

- (R) is some rule (C): this cannot happen since either (C) is (C_v) and P is not simple or either (C) is $(C_{v'})$ and $\{u\}_v = \{u'\}_{v'}$ with $v \neq v'$ which is impossible.

- (R) is some rule (D) s.t. $\dfrac{T \vdash \{\{u\}_v\}_{v'} \quad T \vdash v'}{T \vdash \{u\}_v}$. Then by induction hypothesis $\{\{u\}_v\}_{v'} \in S_\oplus(T)$, yielding $\{u\}_v \in S_\oplus(T)$.

- (R) is (GX). The last deductions in the proof P are described in Figure 7 and we consider the different cases according to the rules (R_i) and to the structure of $\{u\}_v\downarrow$.

$$(D_v)\dfrac{(GX)\dfrac{(R_1)\dfrac{T \vdash B_1}{T \vdash B'_1} \quad \dots \quad (R_n)\dfrac{T \vdash B_n}{T \vdash B'_n}}{T \vdash \{u\}_v\downarrow} \qquad \dfrac{\vdots}{T \vdash v\downarrow}}{T \vdash u\downarrow}$$

Figure 7: Illustration of the case (D) in Lemma 8

We will show that every atom of $\{u\}_v\downarrow$ is in fact an element of $S_T(T)$. Let $a \in \text{atoms}(\{u\}_v\downarrow)$. Note that a is necessarily of the form $\{a'\}_v$, and that there is an i such that $a \in \text{atoms}(B'_i)$. We consider different possible cases for the rule (R_i):

  - (R_i) is (C_v) or (GX): impossible since the rule is $\oplus$-*eager* and flat.
  - (R_i) is $(C_{v'})$ with $v \neq v'$. Then $B'_i = \{u'\}_{v'}\downarrow = \bigoplus_{j=1}^n \{u'_j\}_{v'\downarrow}$. Since $v' \neq v$ none of these $\{u'_j\}_{v'\downarrow}$ can be equal to a, hence this case is impossible, too.
  - (R_i) is (P). Then $B'_i = \langle w_1, w_2\rangle \neq a$ and again this case is impossible.
  - (R_i) is (A) (UL) or (UR). By definition or Lemma 7, $B'_i \in S_\oplus(T)$, hence $a \in S_T(T)$.
  - (R_i) is $(D_{v'})$ s.t. $(D_{v'})\,\dfrac{T \vdash \{w_1\}_{v'} \; T \vdash v'}{T \vdash w_1 = B'_i}$. By induction hypothesis $\{w_1\}_{v'} \in S_\oplus(T)$, therefore $B'_i \in S_\oplus(T)$ and $a \in S_T(T)$.

□

Lemma 22 *Let P be a simple and $\oplus$-eager proof of $T \vdash u$. Then there exists a normal proof of $T \vdash u$.*

Proof: Consider first the case where $u \in S_\oplus(T)$. We proceed by structural induction on the proof P and case distinction of the last rule (R) of P:

- (R) is (A): P is obviously a normal proof.

- (R) is some rule (UL) or (UR) s.t. $\dfrac{T \vdash \langle u_1, u_2\rangle}{T \vdash u}$ The induction hypothesis yields that there exists a normal proof of $\langle u_1, u_2\rangle$. P is simple, we apply Lemma 7 and get $\langle u_1, u_2\rangle \in S(T) \subseteq S_\oplus(T)$. Hence, the normal proof of $\langle u_1, u_2\rangle$ is $S_\oplus(T)$-local so P is normal since $u \in S_\oplus(T)$.

- (R) is some rule (D) s.t. $\dfrac{T \vdash \{u\}_v \quad T \vdash v}{T \vdash u}$ The induction hypothesis yields that there exists a normal proof of $\{u\}_v$. Since P is $\oplus$-*eager* we get with Lemma 8 that $\{u\}_v \in S(T) \subseteq S_\oplus(T)$. Hence, the normal proof of $\{u\}_v$ is $S_\oplus(T)$-local so we deduce that P is normal since $u \in S_\oplus(T)$.

- (R) is some rule (P), (C): Since these two case are similar We only give the proof for the case (C), that is $u = \{u_1\}_{u_2}$. (R) is some (C) s.t. $\dfrac{T \vdash u_1 \quad T \vdash u_2}{T \vdash \{u_1\}_{u_2}}$ Since $\{u_1\}_{u_2} = u \in S_\oplus(T)$ we deduce that $u_1 \in S_\oplus(T)$ and $u_2 \in S_\oplus(T)$. Hence, applying the induction hypothesis, there are normal proofs of u_1 and u_2 that are $S_\oplus$-local, hence P is normal.

- (R) is some rule (GX) such that

$$(GX)\ \dfrac{(R_1)\dfrac{T \vdash B_1}{T \vdash B'_1} \qquad ...(R_n)\dfrac{T \vdash B_n}{T \vdash B'_n}}{T \vdash u}$$

  We will show that for every (R_i) we have that $B'_i \in S_\oplus(T)$. We consider the different cases for the rules (R_i):

  - (R_i) is (GX): impossible since P is $\oplus$-*eager* which implies P is flat.
  - (R_i) is (A), (UL), (UR) or (D): with the definition or Lemma 7 or Lemma 8 we obtain that $B'_i \in S_\oplus(T)$. Applying the induction hypothesis there is a normal proof of B'_i which is $S_\oplus(T)$-local.
  - (R_i) is (P), there are two possible cases: B'_i is in $S_T(u)$ or not.
    - $B'_i \in S_T(u) \subseteq S_\oplus(T)$: we can apply the induction hypothesis and get a normal proof of B'_i which is $S_\oplus(T)$-local.
    - $B'_i \notin S_T(u)$: we have that B'_i is canceled by some other element B'_j. B'_j can not stem from a rule (P) since P is simple, nor from a rule (C) since a pair is not an encrypted term. Hence, B'_j stems from a rule (A), (UL), (UR) or (D). With the definition or Lemma 7 or Lemma 8 we obtain that $B'_j \in S_\oplus(T)$. More precisely $\bigoplus B'_j \in S_\oplus(T)$, since $B'_i \in S_\oplus(\bigoplus B'_j)$ we deduce that $B'_i \in S_\oplus(T)$. We can apply the induction hypothesis and get a normal proof of B'_i which is $S_\oplus(T)$-local.
  - (R_i) is (C_k), this case is similar to the previous case. There are two possible cases: B'_i is in $S_T(u)$ or not:

  - $B'_i \in S_T(u) \subseteq S_\oplus(T)$: we can apply the induction hypothesis and get a normal proof of B'_i which is $S_\oplus(T)$-local.
  - $B'_i \notin S_T(u)$: B'_i is canceled by some other element B'_j. B'_j can not stem from a rule (P) since a pair is not an encrypted term, nor from a rule (C_v) with $v \neq k$ since B'_i is not encrypted with the key k, nor from another rule (C_k) since there is at most one occurrence of an encryption rule by a key above a (GX) rule by the fact that P is $\oplus$-*eager*. So B'_j stems from a rule (A), (UL), (UR) or (D). With the definition or Lemma 7 or Lemma 8 we obtain that $B'_j \in S_\oplus(T)$. More precisely $\bigoplus B'_j \in S_\oplus(T)$, since $B'_i \in S_\oplus(\bigoplus B'_j)$ we deduce that $B'_i \in S_\oplus(T)$. we can apply the induction hypothesis and get a normal proof of B'_i which is $S_\oplus(T)$-local.

Since all the subproofs of $T \vdash B'_i$ are normal we can conclude that P is normal.

In the second case, we assume that $u \notin S_\oplus(T)$ and the proof is of the form $C[P_1, \ldots, P_n]$ where $P_1, \ldots, P_n$ are maximal $S_\oplus$-local subproofs. We prove the result by structural induction on P:

- If C is empty, then $u \in S_\oplus(T)$
- If the last rule is (UL) (UR) or (D) we use the definition and Lemma 7 and Lemma 8 to get $u \in S_\oplus(T)$.
- In the others cases we apply the induction hypothesis.

□

C Proofs for the Section 7

Definition 14 *We call a string* admissible *for given Q, C, U if it is of the form $qx_1\#x_2\overline{x_3}\bot$ where*

- *there is some $a \in Q$ such that $q = a$ or $q = \hat{a}$*
- *either $x_1 = \epsilon$ or $x_1 \in C^*(C \setminus U)$*
- *$x_2, x_3 \in U^*$*

Proposition 6 *The prefix rewrite system of section 7 rewrites admissible strings into admissible strings.*

The following proposition lists some basic properties of the decomposition and inversion of strings which we will use in the sequel without further reference:

Proposition 7 *For all $x, y \in (C \cup U)^*$:*

1. $\overline{xy} = \overline{y}\,\overline{x}$

2. *If $y \in U^*$ then left$(xy) =$ left(x) and right$(xy) =$ right$(x)y$*

We now prove the central lemmata of the prefix rewrite construction:

Lemma 23 *The following two assertions are equivalent for every $a, b \in Q$, $x_1, y_1 \in \{\epsilon\} \cup C^*(C \setminus U)$, $x_2, y_2, y_3 \in U^*$:*

1. *There is a prefix rewrite sequence by PR_1*

$$ax_1\#x_2\bot \mapsto^* by_1\#y_2\overline{y_3}\bot$$

2. *either $a = b$, $x_1 = y_1$, $x_2 = y_2$, $y_3 = \epsilon$,*
or there exists a sequence of binary terms $\{a_i\}_{v_i} \oplus \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, and a sequence of strings $h_i \in U^*$, $i = 1, \ldots, n$, such that

   (a) $\{a\}_{x_1x_2y_3} = \{a_1\}_{v_1h_1}$

   (b) $\{b_i\}_{w_ih_i} = \{a_{i+1}\}_{v_{i+1}h_{i+1}}$ *for $i = 1, \ldots, n-1$*

   (c) $\{b_n\}_{w_nh_n} = \{b\}_{y_1y_2}$

   and such that for some i the longest common suffix of y_3 and h_i is ϵ.

Lemma 24 *The following two assertions are equivalent for every $a, b \in Q$, $x_1, y_1 \in \{\epsilon\} \cup C^*(C \setminus U)$, $x_2, y_2, y_3 \in U^*$:*

1. *There is a prefix rewrite sequence by PR_2*

$$\hat{a}x_1\#x_2\overline{x_3}\bot \mapsto^* \hat{b}y_1\#y_2\bot$$

2. *either $a = b$, $x_1 = y_1$, $x_2 = y_2$, $y_3 = \epsilon$,*
or there exists a sequence of binary terms $\{a_i\}_{v_i} \oplus \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, and a sequence of strings $h_i \in U^*$, $i = 1, \ldots, n$, such that

   (a) $\{a\}_{x_1x_2} = \{a_1\}_{v_1h_1}$

   (b) $\{b_i\}_{w_ih_i} = \{a_{i+1}\}_{v_{i+1}h_{i+1}}$ *for $i = 1, \ldots, n-1$*

   (c) $\{b_n\}_{w_nh_n} = \{b\}_{y_1y_2x_3}$

   and such that for some i the longest common suffix of y_3 and h_i is ϵ.

Proof: First note that the two prefix rewrite systems PR_1 and PR_2 are completely symmetrical (the only purpose of the occurrences of γ in PR_1 is to guarantee admissibility of all reachable configurations). We hence prove only the first lemma, corresponding to the rewrite system PR_1. The proof of the second lemma is completely symmetrical.

For the direction from (1) to (2) we proceed by induction on the length of the rewrite sequence. If the length of the rewrite sequence is 0 then obviously $a = b$, $x_1 = y_1$, $x_2 = y_2$, and $y_3 = \epsilon$. If there is exactly one rewrite step then there are two possible cases:

1. The rewrite rule is of the form

   $$a\ \mathit{left}(r)\#\mathit{right}(r) \to b\ \mathit{left}(s)\#\mathit{right}(s)$$

   Then there exists a u such that

   $$\begin{array}{ll} x_1 = \mathit{left}(r) & y_1 = \mathit{left}(s) \\ x_2 = \mathit{right}(r)u & y_2 = \mathit{right}(s)u \\ & y_3 = \epsilon \end{array}$$

   We conclude by choosing

   $$\begin{array}{rcl} \{a_1\}_{v_1} \oplus \{b_1\}_{w_1} & := & \{a\}_r \oplus \{b\}_s \\ h_1 & := & u \end{array}$$

   since then

   $$\begin{array}{l} \{a\}_{x_1x_2y_3} = \{a\}_{x_1x_2} = \{a\}_{ru} = \{a_1\}_{v_1h_1} \\ \{b_1\}_{w_1h1} = \{b\}_{su} = \{b\}_{y_1y_2} \end{array}$$

2. The rewrite rule is of the form

   $$a\ \mathit{left}(r)\#r_1\bot \to b\ \mathit{left}(s)\#\mathit{right}(s)\overline{r_2}\bot$$

   with $\mathit{right}(r) = r_1r_2$. Then we have

   $$\begin{array}{ll} x_1 = \mathit{left}(r) & y_1 = \mathit{left}(s) \\ x_2 = r_1 & y_2 = \mathit{right}(s) \\ & \overline{y_3} = \overline{r_2},\ \text{hence } y_3 = r_2 \end{array}$$

   We conclude by choosing

   $$\begin{array}{rcl} \{a_1\}_{v_1} \oplus \{b_1\}_{w_1} & := & \{a\}_r \oplus \{b\}_s \\ h_1 & := & \epsilon \end{array}$$

   since then

   $$\begin{array}{l} \{a\}_{x_1x_2y_3} = \{a\}_r = \{a_1\}_{v_1h_1} \\ \{b_1\}_{w_1h_1} = \{b\}_s = \{b\}_{y_1y_2} \end{array}$$

In both cases, the longest common suffix of h_1 and y_3 is ϵ.

In case there are $N > 1$ rewrite steps, the string obtained in $N - 1$ steps is by Proposition 6 admissible. Hence, there are $b \in Q$, $y_1 \in \{\epsilon\} \cup C^*(C \setminus U)$, and $y_2, y_3 \in U^*$ such that

$$ax_1\#x_2\bot \to^* by_1\#y_2\overline{y_3}\bot \to cz_1\#z_2\overline{z_3}\bot$$

By induction hypothesis, there exists a sequence of binary terms $\{a_i\}_{v_i}\oplus\{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, and a sequence of strings $h_i \in U^*$, $i = 1, \ldots, n$, such that

1. $\{a\}_{x_1x_2y_3} = \{a_1\}_{v_1h_1}$

2. $\{b_i\}_{w_i h_i} = \{a_{i+1}\}_{v_{i+1} h_{i+1}}$ for $i = 1, \ldots, n-1$

3. $\{b_n\}_{w_n h_n} = \{b\}_{y_1 y_2}$

and such that the longest common suffix of y_3 and some h_i is ϵ. We will show that there exists some $\{a_{n+1}\}_{v_{n+1}} \oplus \{b_{n+1}\}_{w_{n+1}} \in U$, and a sequence of key strings $k_i \in K^*$, $i = 1, \ldots, n+1$ such that

1. $\{a\}_{x_1 x_2 z_3} = \{a_1\}_{v_1 k_1}$

2. $\{b_i\}_{w_i k_i} = \{a_{i+1}\}_{v_{i+1} k_{i+1}}$ for $i = 1, \ldots, n$

3. $\{b_{n+1}\}_{w_{n+1} k_{n+1}} = \{c\}_{z_1 z_2}$

and such that the common longest suffix of y_3 and some k_j is ϵ. There are two possible cases for the rewrite rule used in the last rewrite step:

1. The rewrite rule is of the form

$$b \ \mathit{left}(r)\#\mathit{right}(r) \rightarrow c \ \mathit{left}(s)\#\mathit{right}(s)$$

Then there exists u such that

$$\begin{array}{ll} y_1 = \mathit{left}(r) & z_1 = \mathit{left}(s) \\ y_2 = \mathit{right}(r)u & z_2 = \mathit{right}(s)u \\ & \overline{z_3} = \overline{y_3}, \text{ hence } z_3 = y_3 \end{array}$$

We conclude by choosing

$$\begin{array}{rcl} \{a_{n+1}\}_{v_{n+1}} \oplus \{b_{n+1}\}_{w_{n+1}} & := & \{b\}_r \oplus \{c\}_s \\ k_i & := & h_i \quad (i = 1, \ldots, n) \\ k_{n+1} & := & u \end{array}$$

since

$$\begin{array}{l} \{a\}_{x_1 x_2 z_3} = \{a\}_{x_1 x_2 y_3} = \{a_1\}_{v_1 h_1} = \{a_1\}_{v_1 k_1} \\ \{b_i\}_{w_i k_i} = \{b_i\}_{w_i h_i} = \{a_{i+1}\}_{v_{i+1} h_{i+1}} = \{a_{i+1}\}_{v_{i+1} k_{i+1}} \quad (i = 1, \ldots, n-1) \\ \{b_n\}_{w_n k_n} = \{b\}_{y_1 y_2} = \{b\}_{ru} = \{a_{n+1}\}_{v_{n+1} k_{n+1}} \\ \{b_{n+1}\}_{w_{n+1} k_{n+1}} = \{c\}_{su} = \{c\}_{z_1 z_2} \end{array}$$

If the longest common suffix of y_3 and h_i, $1 \leq i \leq n$, is ϵ then the longest common suffix of $z_3 = y_3$ and $k_i = h_i$ is ϵ.

2. The rewrite rule is of the form

$$b \ \mathit{left}(r)\#r_1\gamma \rightarrow c \ \mathit{left}(s)\#\mathit{right}(s)\overline{r_2}\gamma$$

with $\mathit{right}(r) = r_1 r_2$, and $\gamma \in \{\overline{u} \mid u \in U\} \cup \{\bot\}$. Then we have

$$\begin{array}{ll} y_1 = \mathit{left}(r) & z_1 = \mathit{left}(s) \\ y_2 = r_1 & z_2 = \mathit{right}(s) \\ & \overline{z_3} = \overline{r_2}\ \overline{y_3}, \text{ hence } z_3 = y_3 r_2 \end{array}$$

We conclude by choosing

$$\begin{array}{rcl} \{a_{n+1}\}_{v_{n+1}} \oplus \{b_{n+1}\}_{w_{n+1}} & := & \{b\}_r \oplus \{c\}_s \\ k_i & := & h_i r_2 \quad (i = 1, \ldots, n) \\ k_{n+1} & := & \epsilon \end{array}$$

since

$$\begin{array}{l} \{a\}_{x_1 x_2 z_3} = \{a\}_{x_1 x_2 y_3 r_2} = \{a_1\}_{v_1 h_1 r_2} = \{a_1\}_{v_1 k_1} \\ \{b_i\}_{w_i k_i} = \{b_i\}_{w_i h_i r_2} = \{a_{i+1}\}_{v_{i+1} h_{i+1} r_2} = \{a_{i+1}\}_{v_{i+1} k_{i+1}} \quad (i = 1, \ldots, n-1) \\ \{b_n\}_{w_n k_n} = \{b_n\}_{w_n h_n r_2} = \{b\}_{y_1 y_2 r_2} = \{b\}_r = \{a_{n+1}\}_{v_{n+1} k_{n+1}} \\ \{b_{n+1}\}_{w_{n+1} k_{n+1}} = \{b_{n+1}\}_{w_{n+1}} = \{c\}_s = \{c\}_{z_1 z_2} \end{array}$$

The longest common suffix of z_3 and $k_{n+1} = \epsilon$ is ϵ.

For the direction from (2) to (1), if $a = b$, $x_1 = y_1$, $x_2 = y_2$, and $y_3 = \epsilon$ then we obviously have that $ax_1 \# x_2 \bot \mapsto^* by_1 \# y_2 \overline{y_3} \bot$. Otherwise, we proceed by induction on n.

If $n = 1$ then there exists $\{a_1\}_{v_1} \oplus \{b_1\}_{w_i} \in U$ and $h_1 \in U^*$ such that

1. $\{a\}_{x_1 x_2 y_3} = \{a_1\}_{v_1 h_1}$
2. $\{b_1\}_{w_1 h_1} = \{b\}_{y_1 y_2}$

and the longest common suffix of y_3 and h_1 is ϵ, that is $y_3 = \epsilon$ or $h_1 = \epsilon$.

1. If $y_3 = \epsilon$ then $x_1 \# x_2 = \mathit{left}(v_1) \# \mathit{right}(v_1) h_1$ and $y_1 \# y_2 = \mathit{left}(w_1) \# \mathit{right}(w_2) h_1$, hence $ax_1 \# x_2 \bot \mapsto by_1 \# y_2 \bot$ by virtue of the the binary term $\{a_1\}_{v_1} \oplus \{b_1\}_{w_2} \in U$.
2. If $h_1 = \epsilon$ then $x_1 \# x_2 = \mathit{left}(v_1) \# v_1^1$ and $y_3 = v_1^2$ for $\mathit{right}(v_1) = v_1^1 v_1^2$, and $y_1 \# y_2 = \mathit{left}(w) \# \mathit{right}(w)$. Hence $ax_1 \# x_2 \bot \mapsto by_1 \# y_2 \overline{y_3} \bot$ by virtue of the the binary term $\{a_1\}_{v_1} \oplus \{b_1\}_{w_2} \in U$.

If $n \geq 2$ then there exists a sequence of binary terms $\{a_i\}_{v_i} \oplus \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, and a sequence of strings $h_i \in U^*$, $i = 1, \ldots, n$, such that

1. $\{a\}_{x_1 x_2 y_3} = \{a_1\}_{v_1 h_1}$
2. $\{b_i\}_{w_i h_i} = \{a_{i+1}\}_{v_{i+1} h_{i+1}}$ for $i = 1, \ldots, n-1$
3. $\{b_n\}_{w_n h_n} = \{b\}_{y_1 y_2}$

and such that for some i the longest common suffix of y_3 and h_i is ϵ.

1. If there is an $i < n$ such that the longest common suffix of y_3 and h_i is ϵ then, by induction hypothesis,

$$ax_1 \# x_2 \mapsto^* b_{n-1} \ \mathit{left}(w_{n-1}) \# \mathit{right}(w_{n-1}) h_{n-1} \overline{y_3}$$

Now, we have that $b_{n-1} \mathit{left}(w_{n-1})\#\mathit{right}(w_{n-1})h_{n-1} = a_n \mathit{left}(v_n)\#\mathit{right}(v_n)h_n$ and that $\{b\}_{y_1 y_2} = \{b_n\}_{w_n h_n}$ Hence,

$$\begin{array}{rl} & b_{n-1}\ \mathit{left}(w_{n-1})\#\mathit{right}(w_{n-1})h_{n-1}\overline{y_3} \\ = & a_n\ \mathit{left}(v_n)\#\mathit{right}(v_n)h_n\overline{y_3} \\ \mapsto & b_n\ \mathit{left}(w_n)\#\mathit{right}(w_n)h_n\overline{y_3} \\ = & by_1\#y_2\overline{y_3} \end{array}$$

2. Otherwise, the longest common suffix of h_n and y_3 is ϵ. Let s be the longest common suffix of y_3 and the h_i for $i < n$, and let y'_3, h'_i $(1 \leq i < n)$ be such that $y_3 = y'_3 s$ and $h'_i = h_i s$. Hence, we also have that

   (a) $\{a\}_{x_1 x_2 y'_3} = \{a_1\}_{v_1 h'_1}$

   (b) $\{b_i\}_{w_i h'_i} = \{a_{i+1}\}_{v_{i+1} h'_{i+1}}$ for $i = 1, \ldots, n-2$

   and for some $i < n$ the longest common suffix of y'_3 and h'_i is ϵ. Hence, by induction hypothesis,

   $$ax_1\#x_2 \mapsto^* b_{n-1}\ \mathit{left}(w_{n-1})\#\mathit{right}(w_{n-1})h'_{n-1}\overline{y_3}'$$

   Now, we have that $\{b_{n-1}\}_{w_{n-1}h_{n-1}} = \{a_n\}_{v_n h_n}$, that is $w_{n-1}h'_{n-1}s = v_n h_n$. Since s is a suffix of y_3, and since the longest common suffix of y_3 and h_n is ϵ, we conclude that $h_n = \epsilon$, and s is a suffix of v_n. We decompose $v_n = v_n^1 s$ and obtain that

   $$\begin{array}{rl} & b_{n-1}\ \mathit{left}(w_{n-1})\#\mathit{right}(w_{n-1})h'_{n-1}\overline{y'_3} \\ = & a_n\ \mathit{left}(v_n)\#v_n^1\overline{y_3}' \\ \mapsto & b_n\ \mathit{left}(w_n)\#\mathit{right}(w_n)\overline{s}\overline{y'_3} \\ = & b_n\ \mathit{left}(w_n)\#\mathit{right}(w_n)h_n\overline{y_3} \\ = & by_1\#y_2\overline{y_3} \end{array}$$

□