Intruder Deduction for the Equational Theory of Abelian Groups with Distributive Encryption

Pascal Lafourcade [a,b] Denis Lugiez [a] Ralf Treinen [b]

[a] *LIF, Université Aix-Marseille 1 & CNRS UMR 6166*

[b] *LSV, ENS de Cachan & CNRS UMR 8643 & INRIA Futurs project SECSI*

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 vulnerability to passive attacks in presence of equational theories which model the protocol specification and properties of the cryptographic operators.

In the present paper we consider the case where the encryption distributes over the operator of an *Abelian group* or over an *exclusive-or* operator. We prove decidability of the intruder deduction problem in both cases. We obtain a PTIME decision procedure in a restricted case, the so-called *binary* case.

These decision procedures are based on a careful analysis of the proof system modeling the deductive power of the intruder, taking into account the algebraic properties of the equational theories under consideration. The analysis of the deduction rules interacting with the equational theory relies on the manipulation of $\mathbb{Z}$-modules in the general case, and on results from prefix rewriting in the binary case.

1 Introduction

Cryptographic protocols are ubiquitous in distributed computing applications. They are employed for instance in internet banking, video on demand services,

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

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 [1] 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 [2], more recent data bases of protocols and known flaws are [3,4].

There are different approaches to modeling cryptographic protocols and analyzing their security properties: process calculi like the *spi-calculus* [5], so-called cryptographic proofs (see, for instance, [6]), and the approach of Dolev and Yao [7] 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 proved too idealistic: There are protocols which can be proved 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). Algebraic properties which are not used explicitly in the protocol can still be exploited by an attacker to mount an attack; see [8] for an overview of the verification of cryptographic protocols in presence of algebraic properties. For instance, many cryptographic protocols manipulate data and operations that actually use an *Abelian group*. The Wired Equivalent Privacy protocol [9], Gong's protocol [10], and Bull's protocol [11] use explicitly in their specification the *exclusive-or* operation (which has, in addition to being an Abelian group, the *nilpotence* property $x + x = 0$). Moreover, the cryptographic DES algorithm and the more recent AES rely on the algebraic property of the *exclusive-or*. These two properties employed with properties of the encryption algorithms are the most commonly used, hence the most important ones.

Finally, note that the well-known cryptosystem RSA has the property $\{a * b\}_k = \{a\}_k * \{b\}_k$, where $\{x\}_k$ denotes the encryption of message x with key k, if we abstract from the modulus used in the RSA encryption operation. This property of RSA is the distributivity of the encryption operation over the multiplication of non-null integers modulo $n = pq$ where p, q are large prime numbers. Strictly speaking, this structure is not an Abelian group but a commutative semigroup since some of the elements do not have an inverse. In fact, the only elements without inverse are the multiples of p (i.e $p, 2p, \ldots, (q-1)p$) and of q (i.e. $q, 2q, \ldots, (p-1)q$), totaling $p + q - 2 = O(\sqrt{n})$ elements without inverse, while the total number of elements is $pq - 1 = O(n)$. Hence, this structure is "mostly" an Abelian group since the vast majority of elements does have an inverse. Besides, the assumption of an Abelian group leads to an over-approximation of the intruder capabilities (since we assume that the intruder can inverse all elements, where in reality this is not true). This may lead to false positives (claiming that there is an attack when none exists), but is safe for proving non-existence of an attack.

The state of the art. The first step in deciding protocol (in)security is usually to prove that the intruder deduction problem is decidable, therefore this question can be seen as a prerequisite to solving the more general problem of an active intruder. Both problems are decidable[1] for the *exclusive-or* and for Abelian groups [12,13], as well as for modular exponentiation [14,15] which is modeled by a restricted axiomatization. Therefore, the intruder deduction problem is decidable in all these cases, usually with a polynomial time complexity. The intruder deduction problem is decidable in polynomial time in the case of the equational theory of a homomorphism [16], and in both the case of *exclusive or* combined with a homomorphism and Abelian groups combined with a homomorphism [17]. Protocol security has been shown decidable in case of a homomorphism that distributes over the *exclusive-or* [18], and undecidable in case of a homomorphism that distributes over an Abelian group [19].

Several works [20,21,22,23] have been done to characterize classes of equational theories for which a generic algorithm could apply. These theories are presented by a rewrite system, and the required property is that the right-hand side of a rewrite rule is a strict subterm of the left-hand side. Another trend [24] is to devise a combination algorithm which allows us to combine decision algorithms that have been defined for independent equational theories, yielding a solution for the union of the theories (in the active case). Finally, a

[1] Protocol security is considered in these papers for a fixed number of sessions.

recent approach [25] is to try to have a general resolution technique that relies on narrowing, but this has not yet succeeded to come up with a satisfactory solution.

On passive and active attacks Proving security of a protocol against passive attacks is in itself an important problem. In fact, there are many situations in which an intruder cannot interact with the legitimate protocol participants, be it due to physical restrictions of the communication channel, or due to the fact that an intruder only gets hold of a log of a protocol session after the communication channel is closed. In general, however, one usually wishes in the end to obtain a guarantee of security against active attacks, that is security even in a scenario where an attacker completely controls the network.

Security against passive attacks is obviously a sub-problem of security against active attacks, since any active attacker can of course try to obtain a secret by purely passive means. Security against active attacks is undecidable if the number of parallel protocol sessions is unbounded [26], even without any equational theory. In case of a bounded number of sessions, finding decision algorithms for proving security against active attacks is still a difficult problem. One important technique for obtaining such decision procedures is based on the technique of symbolic constraint solving, where a single constraint expresses that the intruder can deduce a term from some finite set of terms. In contrast to the intruder deduction problem considered here, the terms used in the constraints may contain variables, and we are interested in knowing whether these variables can be instantiated in such a way that the constraint holds. In the special case where the constraint does not contain any variables, solving a constraint amounts to solving an intruder deduction problem.

There are different ways to solve the constraints. One way consists in showing that any solvable constraint system has a "small solution". Thus, one obtains a non-deterministic procedure which guesses a small solution, and then uses a decidability result of the intruder deduction problem in order to verify that the guess is correct. This is the approach of for instance [15] for the theory of Diffie-Hellman exponentiation with products occurring in exponents, or [13] for the equational theory of *exclusive-or*.

Another approach to constraint solving consists in successively simplifying the constraints, possibly combined with non-deterministic guessing steps [27]. This is the approach of for instance [12] for the theory of *exclusive-or*, or of [18] for the theory of *exclusive-or* with one homomorphism. In fact, the technical core of the decision procedure for the intruder deduction problem given in the present paper is a *locality result*, stating that if the intruder can deduce a certain knowledge w from an initial knowledge T then there is a deduction using only terms already contained (in a sense which we will make precise along this paper) in T or w. An important first step in solving constraints consists in lifting this locality lemma to the general case of constraints containing variables, yielding that if there is a solution to a constraint then there is a "simple" solution (Lemma 2 in [18]), and that for this simple solution there is a "simple" proof (Lemma 3 in [18]). These lemmas justify a first non-deterministic step in constraints solving which consists in reducing constraints to sequences of one-step deduction constraints.

Our contribution. In this paper we consider protocols that use the Abelian group axioms together with an encryption algorithm which distributes over the binary operator of an Abelian group denoted by $+$, i.e. $\{x+y\}_k = \{x\}_k + \{y\}_k$. This property is used in several protocols, like for instance the TMN protocol [28], which use the distributivity of the RSA cryptosystem over the Abelian group operator (see [8] for details). It is related to the homomorphism property $h(x+y) = h(x) + h(y)$ if we consider encryption by a fixed key as a homomorphic operator. However, our theory is more general than the theory of one (or finitely many) homomorphic operator since distributivity holds for *all* keys, that means that we have an infinite number of homomorphisms to deal with. Actually, a homomorphism h can be simulated by a public key that is known by all participants (but where the owner of the associated private key does not play any role in the protocol).

We show that the intruder deduction problem for an encryption algorithm which distributes over the operator of an Abelian group is decidable. Moreover, we give a polynomial complexity bound in the binary case. Since the theory of homomorphism with Abelian groups is undecidable for an active intruder [19] and since this theory can be simulated in our framework, our result is the strongest possible one (with respect to decidability questions). The decision procedure relies first on a careful analysis of the proof system modeling the intruder deduction abilities, allowing us to state the existence of proofs that have good syntactical properties. This can be seen as the analog of proof normalization techniques which are widely used in logic to show that proof systems have good properties (like the subformula property for Gentzen sequent calculus, see e.g. [29]). A second step is to use algebraic properties of the equational theory to decide the deducibility of a term using a restricted subset of the rules. This is done easily for the *exclusive-or* case but it uses more complex properties of $\mathbb{Z}$-modules for the Abelian group case. Finally, we combine these results to state a locality theorem generalizing McAllester's *locality* method explained in Section 5. For the sake of simplicity we present our result in the symmetric encryption framework, but it can be lifted to the public key encryption framework.

The theory of Abelian groups with a distributive encryption can not be treated by the general approaches that have been devised so far. The subterm property required in the general approach of [22] does not hold here since $\{x\}_k$ is not a subterm of $\{x + y\}_k$, and there is no way to adapt this technique to our case. Furthermore, the combination result of [24] can't be used neither since the theory that we consider cannot be split into disjoint simpler equational theories which is the starting point in the combination approach, and the finite variant property required in [25] is not satisfied. Actually, we believe that the theory that we consider falls into a class that requires another approach than what these works propose.

Plan of the paper: In the next section we give an example of a protocol on which there exists a passive attack exploiting the distributivity of encryption over *exclusive-or* or an Abelian group operator. We present in Section 3 the usual notions needed in the rest of the paper. In Section 4 we introduce the Dolev-Yao model of intruder capacities extended by a rewrite system modulo AC to model the distributivity of the encryption symbol over the Abelian group operator. In Section 5 we explain the generalization of McAllester's proof technique. In the following sections we provide the two main ingredients which allow us to obtain a decision procedure: We show in Section 6 a syntactic locality result considering the rules of encryption, decryption and addition in a macro rule and we demonstrate in Section 7 and 8 the decidability for this macro rule using $\mathbb{Z}$-modules. We sum up in Section 9 our main results and discuss in Section 10 the restriction to the binary case and give a decision procedure in PTIME using prefix rewrite systems. Finally, we conclude in Section 11.

2 An Introductory Example

Figure 1 gives an example of a simple protocol which is designed to distribute a symmetric key K to two principals A, B using a service provider S. The principals A, B already share a weak secret c that states their right to share a common symmetric key (for instance, c is some item that proves subscription to the service). The value of c is also known to the server. The server has a public key K_S which is known to A and B. In the following, the pair of messages m, m' is denoted by $\langle m, m' \rangle$ and the encryption of a message m by a key K is denoted by $\{m\}_K$. For sake of readability, we denote in the following example the pair $\langle m, m' \rangle$ by m, m'.

The first message of A signals to S that she wants to establish a connection with B. The server S can compute the nonce N_A since she can decrypt the message encrypted by K_S and subtract A from this message. The server S then computes c in the same way. Then, S informs B that A wants to start the

$$A \to S : A, B, \{A + N_A\}_{K_S}, \{N_A + c\}_{K_S}$$
$$S \to B : A, B, S$$
$$B \to S : B, A, \{B + N_B\}_{K_S}, \{N_B + c\}_{K_S}$$
$$S \to A : K + \{N_A\}_{K_S}$$
$$S \to B : K + \{N_B\}_{K_S}$$

Figure 1. A protocol for key distribution.

protocol with him, and a similar connection is established with B allowing S to check that A and B both know c and are hence allowed get the same symmetric key K. Then, the server sends the key K to A, using the Vernam encryption scheme with $\{N_A\}_{K_S}$ to protect the key. The agent A can compute $\{N_A\}_{K_S}$ and retrieve K. The use of the nonce N_A provides a (weak) authentication in this last step. The agent B performs the same operation and obtains the same key K. A property of the protocol is that all encryptions by K_S use different terms, which is useful in preventing replay attacks.

The question of whether there exists an attack (passive or active) against this protocol depends on the algebraic theory taken into account. First, we have checked with the AVISPA tool that there exists no active attack against this protocol for three principals A, B, C and two parallel sessions without any equational theory. This means in particular that there exists no passive attack against the protocol in the empty equational theory.

Second, we have used the tools OFMC [30] and Cl-Atse [31] from the AVISPA project in order to check for existence of active attacks when taking into account the algebraic theory of *exclusive or*. For this equational theory, there still is no active attack in the sense that the intruder cannot obtain the secret key K generated by the server. As a consequence, there is no passive attack for the equational theory of *exclusive or*.

However, there is a (passive) attack if we take into account the full equational theory of *exclusive or* with distributive encryption, and also for the equational theory of Abelian groups with distribute encryption. The attack goes as follows. By intercepting the first message, the intruder can compute $\{N_A\}_{K_S}$ using the distributivity of encryption since A is public, hence $\{A\}_{K_S}$ is too. Therefore, the intruder can retrieve K from the last message by subtracting $\{N_A\}_{K_S}$ from it. If we modify the protocol by distributing all encryptions over $+$, for instance replacing the term $\{A+N_A\}_{K_S}$ by $\{A\}_{K_S}+\{N_A\}_{K_S}$, then the tools OFMC and Cl-Atse find the passive attack described previously.

The results of this paper allow to detect these attacks automatically on the unmodified protocol.

3 Preliminaries

We summarize some basic notations used in this paper, see [32,33] for an overview of rewriting.

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 $|t| = \mathit{cardinality}(\mathcal{O}(t))$. 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, written $t \mapsto s$. 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 $t \to s$. If $t \to^* s$ 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 $t_1 \to t_2 \to t_3 \to \cdots$.
- *locally confluent*, that is if $t \to s_1$ and $t \to s_2$ then there exists a term r with $s_1 \to^* r$ and $s_2 \to^* r$.

By a well known result (see, e.g., [32]), every convergent rewrite system is *confluent*, that is if $t \to^* s_1$ and $t \to^* s_2$ then there exists a term r with $s_1 \to^* r$ and $s_2 \to^* r$. 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 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 terms. If the mapping S is idempotent and monotone then it is transitive.*

Proof: straightforward. $\square$

4 Our Model

We use the classic model of deduction rules [7] introduced by Dolev and Yao in order to model the deductive capacities of a passive intruder. We present here an extension of this model which takes into account an equational theory.

4.1 Equational Theory

We consider the equational theory where encryption, denoted by $\{.\}_.$, distributes over the binary operator of an Abelian group. The Abelian group is modeled by the operator $+$, a neutral element 0 and the inversion operator $-$.

The equational theory E consists of the following axioms:

$$
\begin{aligned}
(x+y)+z &= x+(y+z) && \text{Associativity} \\
x+y &= y+x && \text{Commutativity} \\
x+0 &= x && \text{Neutral Element} \\
x+(-x) &= 0 && \text{Inversion} \\
\{x+y\}_k &= \{x\}_k+\{y\}_k && \text{Distributivity 1} \\
\{-x\}_k &= -\{x\}_k && \text{Distributivity 2}
\end{aligned}
$$

This equational theory is represented by a convergent rewrite system R modulo AC, that is R is terminating and confluent modulo associativity and commutativity of $+$, and for all terms $t, s \in T(\Sigma)$ we have that $t =_E s$ if and only if $t\downarrow_{R/AC} =_{AC} s\downarrow_{R/AC}$. Note that $\{0\}_z = 0$ is a consequence of the equational axioms. The convergent rewrite system R consists of the following rules:

$$
\begin{aligned}
x+0 &\to x \\
x+(-x) &\to 0 \\
-0 &\to 0 \\
-(-x) &\to x \\
-(x+y) &\to (-x)+(-y) \\
\{x+y\}_z &\to \{x\}_z+\{y\}_z \\
\{-x\}_k &\to -\{x\}_k \\
\{0\}_z &\to 0
\end{aligned}
$$

The complete signature is $\Sigma = \{\langle\cdot,\cdot\rangle, \{\cdot\}_\cdot, 0, +, -\} \uplus \Sigma_0$, where Σ_0 is a set of free constant symbols. The symbols not pertaining to the Abelian group come from the Dolev-Yao model. The first one $\langle\cdot,\cdot\rangle$ is used to build a pair of two messages and the second one $\{\cdot\}_\cdot$ is used to encrypt a message by a key. For the sake of simplicity we here only consider symmetric encryption.

This equational theory can be extended to model the *exclusive-or* operation by adding the axioms[2]

$$
\begin{aligned}
(-x) &= x \\
x+x &= 0
\end{aligned}
$$

[2] The second axiom is a logical consequence of the other axioms but it is convenient to have the associated rewrite rule.

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

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

$$(C)\frac{T \vdash u \qquad T \vdash v}{T \vdash \{u\}_v\downarrow_{R/AC}} \qquad (D)\frac{T \vdash \{u\}_v\downarrow_{R/AC} \qquad T \vdash v}{T \vdash u\downarrow_{R/AC}}$$

$$(GX)\frac{T \vdash u_1 \qquad \ldots \qquad T \vdash u_n}{T \vdash (\alpha_1 u_1 + \ldots + \alpha_n u_n)\downarrow_{R/AC}} \quad \textit{where } \alpha_1, \ldots, \alpha_n \in \mathbb{Z} \setminus \{0\}.$$

Figure 2. A Dolev-Yao proof system working on normal forms by the rewrite system R modulo AC.

which are oriented as rewrite rules

$$(-x) \rightarrow x$$

$$x + x \rightarrow 0$$

to get a convergent rewrite system R modulo AC.

In the rest of the paper, we use an abbreviation for sums of terms in Abelian groups: Given an integer $\alpha \in \mathbb{Z}$ and a term t, we denote by αt the sum of α times the term t if $\alpha \geq 0$, and the sum of $|\alpha|$ times the term $-t$ when $\alpha < 0$.

4.2 An Extended Dolev-Yao Model for our Equational Theory

We assume that the intruder can exploit the equational theory given above to mount an attack. The knowledge of the intruder is represented by terms built over the signature Σ defined previously. Let T be a finite set of ground terms and u be a ground term, a sequent $T \vdash u$ denotes the fact that the intruder can deduce u from the initial knowledge T. The deduction system describing the deductive capacities of an intruder is given in Figure 2.

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) and (UR) he can extract each member of a pair, (C) he can encrypt a message u with a key v, (D) if he knows a key v he can decrypt a message encrypted by v. Sometimes, we shall annotate the rules (C) and (D) by the key that they use, yielding rules (C_v) and (D_v). Finally, there is a family (GX) of rules which allow the intruder to construct a sum of terms, possibly using the same term several times.

Note that for all sequents $T \vdash u$ derivable by the inference system the term u is in normal form by the rewrite system R/AC. This is obvious for the rule (A) by definition. For the rules (UL), (UR) and (P) it holds by induction since the rewrite system does not concern the pairing symbol. The rules (C), (GX) and (D) explicitly state normalization of the resulting term. An example of an instance of rule (D) is

$$(D)\frac{T \vdash 2\{a\}_k + 3\{\{b\}_l\}_k \qquad T \vdash k}{2a + 3\{b\}_l}$$

since $2\{a\}_k + 3\{\{b\}_l\}_k$ is the normal form of $\{2a + 3\{b\}_l\}_k$.

It is easy to see that 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 [16] without AC axioms; in [34] this has been extended to the case of a rewrite system modulo AC.

From now on we will omit the index R/AC and write $\downarrow$. We assume that the set T consists only of terms in normal form.

In the case of *exclusive-or*, the same deduction system works, but we may assume that all terms u_i in the premises of the (GX) rule are different and that all coefficients α_i in the conclusion are equal to 1.

5 Generalization of Locality and Complexity of the Intruder Deduction Problem

Our starting point is the locality technique introduced by McAllester [35]. He considers deduction systems that 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 3 where S is a function which maps any set of terms to its set of subterms (the set of *syntactic* subterms in McAllester's original algorithm). In this algorithm we denote the

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 $T_p = T$
return $w \in T$

Figure 3. Checking existence of an S-local proof.

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 [12]. Therefore we use a rule (GX) with an arbitrary number of hypotheses because we need to collapse several applications of this rule into a single one to establish commutation properties. However, we are now stuck with an infinite number of rules. Fortunately, we can implement the test of one-step deducibility in the loop of McAllester's algorithm in a clever way that allows us 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 an 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 $S \vdash^{=1} u$ 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 3, 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 $g(f(n))$. $\square$

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

(i) show that one-step deducibility can be tested in time $g(n)$, for some complexity measure g,
(ii) define a notion of subterms which can be computed in time $f(n)$, for some complexity measure f,
(iii) show locality with respect to this notion of subterms.

We first notice that one-step deducibility is decidable in polynomial time for all the rules except the rule (GX), since these rules have a fixed, bounded number of hypotheses. One-step deducibility of the rule (GX) for the equational theory of *exclusive-or* and Abelian group with a distributive encryption is analyzed in Section 8.

In Section 6, we adapt the definition of syntactic subterms to our case (Definition 3) and we establish several properties of the proof system that allow us to get a locality result for a modified system consisting of the rules (A), (UR), (UL), (P) and $(GXCD)$, the last one representing combinations of the $(GX), (C), (D)$ rules (Section 7). The applicability of this last rule is established in Section 8. All these results yield the decidability of the intruder deduction problem for the case of *exclusive-or* with distributive encryption and for the case of Abelian groups with distributive encryption (Section 9). In Section 10 we shall define a polynomial notion of subterms in the binary case, which allows us to get a polynomial-time complexity in this case.

6 Syntactic Locality

We first define the notion of syntactic subterms. Second we characterize the kind of proofs which allows us to demonstrate some technical lemmas. Finally we prove in Lemma 5 a partial locality result for a modified proof system called S_{GXCD}.

We need first to characterize when a term is a sum of terms or a negative term.

Definition 2 *Let u be a term in normal form, u is* headed with $+$ *if u is of the form $u_1 + \ldots + u_n$ with $n > 1$. Otherwise u is* not headed with $+$. *Let u be a term in normal form, u is* headed with $-$ *if u is of the form $-v$ where v is a term not headed with $+$. Otherwise u is* not headed with $-$.

Example 1 *$t_1 = -2u + 3\langle v, w\rangle$ is headed with $+$ and not headed with $-$, and $t_2 = \{\langle 3v, w\rangle\}_k$ is headed neither with $+$ nor with $-$. Notice that according to our notations $-3u = (-u) + (-u) + (-u)$ is headed with $+$ and that $-a$ is headed with $-$ and not with $+$.*

We define a notion of syntactic subterms.

Definition 3 *The set of* syntactic subterms *of a term t in normal form 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 $u = u_1 + \ldots + u_n + (-u_{n+1}) + \ldots + (-u_{n+m}) \in S(t)$ and u_i not headed with $+$ and not headed with $-$ then $S(u_i) \subseteq S(t)$.*

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

Example 2 *For $t = 2a + \langle 3b + c, d\rangle$ we have that*

$$S(t) = \{t, a, \langle 3b + c, d\rangle, 3b + c, b, c, d\}$$

Note that, by our definition, $2a$ and $3b$ are no syntactic subterms of t.

We demonstrate some properties of the syntactic subterms which will be used implicitly many times in the rest of the paper.

Proposition 2 *Let A and B be two sets of terms in normal form, the mapping S of syntactic subterms has the following properties:*

- *$S(A \cup B) = S(A) \cup S(B)$.*
- *S is idempotent : $S(S(A)) = S(A)$.*
- *S is monotone : if $A \subseteq B$ then $S(A) \subseteq S(B)$.*
- *S is transitive.*

Proof: These properties are consequences of Definition 3 of syntactic subterms and Proposition 1. □

Example 3 below demonstrates that the notion of syntactic subterm is not sufficient to get the locality result. In this example, the first proof applies the rule (GX) only once in the end, while in the second proof "partial sums" are formed as early as possible. With this latter kind of proof, which we will formally define in the next section, we can limit the number of encryption symbols used in the terms of the proof. This point is an important ingredient of our approach to demonstrate the decidability of the intruder deduction problem.

Example 3 *Consider the following proof with* $T = \{a - \{b\}_k, \{b\}_k - c, \{c\}_k - d, k\}$ *and* $w = \{a\}_k - d$ *where* $\Sigma_0 = \{a, b, c, d, k\}$. *We compute*

$$S(T, w) = T \cup \{w\} \cup \{\{a\}_k, a, \{b\}_k, b, \{c\}_k, c, d, k\}$$

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

This proof of $T \vdash w$ *is not* S*-local since* $\{\{b\}_k\}_k$ *is not in* $S(T, w)$.

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

In this second proof of $T \vdash w$, *the term* $a - c$ *is not in* $S(T, w)$, *hence this proof is not* S*-local.*

6.2 Minimal, Simple and Flat Proofs

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

Definition 4 *Let* P *be a proof of* $T \vdash w$.

- *A* subproof P' *of* P *is a subtree of* P.
- *The* size of a proof P, *denoted by* $|P|$, *is the number of nodes in* P.
- *A proof* P *of* $T \vdash w$ *is minimal if for all proofs* P' *of* $T \vdash w$: $|P| \leq |P'|$.

$$
(GX)\dfrac{(GX)\dfrac{T \vdash x_1 \quad \ldots \quad T \vdash x_n}{T \vdash \alpha_1 x_1 + \ldots + \alpha_n x_n} \qquad T \vdash y_1 \quad \ldots \quad T \vdash y_m}{T \vdash \beta(\alpha_1 x_1 + \ldots + \alpha_n x_n) + \beta_1 y_1 + \ldots + \beta_m y_m}
$$

$$
\Downarrow
$$

$$
(GX)\dfrac{T \vdash x_1 \quad \ldots \quad T \vdash x_n \quad T \vdash y_1 \quad \ldots \quad T \vdash y_m}{T \vdash \beta\alpha_1 x_1 + \ldots + \beta\alpha_n x_n + \beta_1 y_1 + \ldots + \beta_m y_m}
$$

Figure 4. Transformation of (GX)-(GX) into (GX).

- *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 as it is described in Figure 4.

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 $|P|$.

Proposition 3 *Let P be a simple proof then :*

(1) there is no rule (D_v) just after a rule (C_v) in P.
(2) there is no rule (C_v) just after a rule (D_v) in P.

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

Proposition 4 *Let P be a proof of $T \vdash w$.*

- *If P is minimal then P is flat.*
- *If P is minimal then P is simple.*

Proof: If the proof P of $T \vdash w$ is not flat then the proof is not minimal since we can merge two rule (GX) and obtain a smaller proof. Similarly, if a proof P of $T \vdash w$ is not simple then the proof is not minimal since we can cut the loop in the proof and construct a smaller proof. □

These two propositions will be used implicitly in the rest of the paper.

We demonstrate a technical lemma used in the proof of Lemma 3. Then we prove Lemma 3.

Lemma 2 *Let P be a simple proof of the form:*

$$P = \left\{ (R)\frac{P_1 \qquad \dots \qquad P_n}{T \vdash w} \right.$$

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

Proof: We consider all possible rules for the last rule (R) of P:

- The last rule is (A): obvious
- The last rule is (UL) or (UR): In this case we have $n = 1$ and $w_1 = \langle u_1, u_2\rangle$ where w is one of u_1 or u_2. We conclude by induction hypothesis since $\langle u, v\rangle \in S(w) \subseteq S(w_1)$.
- The last rule is (D): In this case we have $n = 2$ and $w_1 = \{w\}_{w_2}$. We conclude by induction hypothesis since $\langle u, v\rangle \in S(w) \subseteq S(w_1)$.
- The last rule is (GX): $\langle u, v\rangle \in S(w)$ by hypothesis and $w = (w_1 + \ldots + w_n)\downarrow$. Hence by definition of the subterm relation $\langle u, v\rangle \in \cup_i S(w_i)$, more precisely there exists i such that $\langle u, v\rangle \in S(w_i)$, since $\langle u, v\rangle$ is not headed with $+$. We conclude with the induction hypothesis.
- The last rule is (P): since $T \vdash w$ can not occur in P by simplicity of the proof P, we have that $w = \langle w_1, w_2\rangle \neq \langle u, v\rangle$. Since $\langle u, v\rangle \in S(w)$ by hypothesis we obtain that $\langle u, v\rangle \in S(w_1) \cup S(w_2)$ and we conclude with the induction hypothesis.
- The last rule is (C): We have $n = 2$ and $w = \{w_1\}_{w_2}$. Since $\langle u, v\rangle \in S(w)$ by hypothesis we obtain that $\langle u, v\rangle \in S(w_1) \cup S(w_2)$ and we conclude with the induction hypothesis. $\square$

Lemma 3 *Let P be a simple proof of $T \vdash u$. If P is one of*

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

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

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

$$P = \begin{cases} (UL)\dfrac{\dfrac{P_1 \ldots P_n}{T \vdash \langle u, v\rangle}}{T \vdash u} \end{cases}$$

P is simple so $T \vdash u$ does not occur in any of $P_1, \ldots, P_n$. Hence, we can apply Lemma 2 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 2 can be applied again and the iteration of this reasoning finally leads to $\langle u, v\rangle \in T$. $\square$

Lemma 4 *Let P be a minimal proof of $T \vdash w$. If the proof P contains a rule of pairing (P) of the following form:*

$$(P)\dfrac{\dfrac{\vdots}{T \vdash u} \quad \dfrac{\vdots}{T \vdash v}}{T \vdash \langle u, v\rangle}$$

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

Proof: We prove the result by structural induction on the the proof P of $T \vdash w$. There are different cases according to the last rule of the proof P:

- The last rule is (A). This case is trivial.
- The last rule is (UR) or (UL).

$$(UL)\dfrac{\dfrac{P_1}{T \vdash \langle w, v\rangle}}{T \vdash w}$$

  By induction hypothesis we know that all terms generated by a rule (P) in the proof P_1 are in $S(T, \langle w, v\rangle)$. Since a minimal proof is simple we obtain by Lemma 3 that $\langle w, v\rangle \in S(T)$. As a consequence, all terms generated by a rule (P) are in $S(T) \subseteq S(T, w)$.
- The last rule is (C).

$$(C)\dfrac{\dfrac{P_1}{T \vdash u} \quad \dfrac{P_2}{T \vdash k}}{T \vdash \{u\}_k = w}$$

  By induction hypothesis we know that all terms generated by a rule (P) are in $S(T, u)$ for the subproof P_1 and that all the pairs generated by a rule (P) are in $S(T, k)$ for the subproof P_2. By definition of S we have that $u \in S(\{u\}_k) = S(w)$ and $k \in S(\{u\}_k) = S(w)$. Hence, all terms generated by a rule (P) in the proof P are in $S(T, w)$.

- The last rule is (P). The claim is obviously true for the occurrence of (P) which is at the root of the proof P. The rest of the demonstration is similar to the previous case with rule (C).

$$(P)\frac{\dfrac{P_1}{T \vdash u} \quad \dfrac{P_2}{T \vdash v}}{T \vdash \langle u, v\rangle = w}$$

By induction hypothesis we know that all terms generated by a rule (P) are in $S(T, u)$ for the subproof P_1 and that all the pairs generated by a rule (P) are in $S(T, v)$ for the subproof P_2. By definition of S we have that $u \in S(\langle u, v\rangle) = S(w)$ and $v \in S(\langle u, v\rangle) = S(w)$. Hence, all terms generated by a rule (P) in the proof P are in $S(T, w)$.
- The last rule is (GX).

$$(GX)\frac{\dfrac{P_1}{T \vdash u_1} \quad \cdots \quad \dfrac{P_n}{T \vdash u_n}}{T \vdash w = (\alpha_1 u_1 + \ldots + \alpha_n u_n)\downarrow}$$

Consider an occurrence of rule (P) yielding some term $v = \langle v_1, v_2\rangle$ in some subproof P_i. By induction hypothesis, $v \in S(T, u_i)$. Assume that $v \notin S(T, w)$. This implies that $v \notin S(T)$, hence $v \in S(u_i)$.

By consequence, this occurrence of v is "canceled out" in the sum, that is we have that $v \in S(u_j)$ for some $j \neq i$. We can now obtain a smaller proof of $T \vdash w$ as follows: We ascend in both subproofs P_i and P_j the maximal paths of nodes from u_i, resp. u_j on which the nodes contain v as a subterm. Either one of the paths ends in an application of rule (A), in which case we conclude that $v \in S(T)$ in contradiction to the hypothesis $v \notin S(T, w)$. Otherwise all these paths end in an application of rule (P). In this case we replace this application of rule (P) by the subproof leading to $T \vdash v_1$, we cut the subproof of v_2, and in all nodes on the two paths we replace the subterm v by v_1. This yields a smaller proof, in contradiction to the minimality of the proof.
- The last rule is (D).

$$(D)\frac{\dfrac{P_1}{T \vdash \{w\}_k} \quad \dfrac{P_2}{T \vdash k}}{T \vdash w}$$

We use a similar reasoning as in the previous case for the rule (GX). Consider an occurrence of rule (P) yielding some term $v = \langle v_1, v_2\rangle$ in some subproof P_i, and assume that $v \notin S(T, w)$.

If the application of (P) occurs in P_1, we get $v \in S(T, \{w\}_k)$ by induction hypothesis, hence $v \in S(T, k)$ since we assumed that $v \notin S(T, w)$ and a pair cannot be an encrypted term. If the application of (P) occurs in P_2 we get that $v \in S(T, k)$ by induction hypothesis.

We show how to get a smaller proof of $T \vdash w$. We consider the maximal subpaths of the proof P such that all nodes of the path are labeled by a term $T \vdash u$ such that $v = \langle v_1, v_2 \rangle \in ST(u)$. The rule corresponding to the ending node of these path cannot be (A), otherwise we get $v \in ST(T)$ which is not possible since we assume that $v \notin S(T, w)$. Therefore the rule labeling these nodes is (P). Let us consider the new tree obtained by replacing each subtree $\dfrac{\dfrac{\vdots}{T \vdash v_1} \quad \dfrac{\vdots}{T \vdash v_2}}{T \vdash v = \langle v_1, v_2 \rangle}$ by $\dfrac{\vdots}{T \vdash v_1}$ and every occurrence of v under this node by v_1. A straightforward induction on the structure of proofs shows that these rewriting process still yields a valid proof of $T \vdash w$. By definition this proof is smaller than the initial one, which contradicts the minimality assumption. □

6.4 Partial Locality Result

We now prove Lemma 5 which states a locality property of a variant of the proof system: we consider all successive applications of the rules (GX), (C) and (D) as a "macro" rule denoted $(GXCD)$. This rule takes as hypotheses all the hypotheses of the rules (GX), (C) and (D) and yields the result of all these rules. In this proof system, called S_{GXCD}, we only have the rules of (A), (UR), (UL), (P) and $(GXCD)$.

Lemma 5 *Let P be a proof of $T \vdash w$ in S_{GXCD}, then there exists a proof P' of $T \vdash w$ in S_{GXCD} such that all the nodes of P' are in $S(T, w)$.*

Proof: Let P be a proof of $T \vdash w$ in S_{GXCD}, we consider a proof of $T \vdash w$ in the initial system which exists by construction of the rule $(GXCD)$. We construct from this proof the minimal proof of $T \vdash w$ in the initial system. By Lemma 2 and 4, all nodes resulting from a rule (UR), (UL) or (P) are in $S(T, w)$. We reconstruct with this proof a proof in S_{GXCD}, we obtain that:

- all nodes which are hypotheses or conclusion of a rule (UR), (UL) or (P) are in $S(T, w)$.
- all hypothesis of all the rules $(GXCD)$ stem from T or from a rule (UR), (UL) or (P), and by consequence are in $S(T, w)$.
- all conclusions of all the rules $(GXCD)$ are either an hypothesis of one rule (UR), (UL) or (P), or it is w, and by consequence are in $S(T, w)$.

We conclude that all nodes of this new proof of $T \vdash w$ in S_{GXCD} are in $S(T, w)$. □

All the proofs of all the lemmas of this section carry over to the *exclusive-or* case without any modification.

7 Elementary Provability

By Theorem 1 and Lemma 5 we have now to show that one-step deducibility by the "macro rule" $(GXCD)$ is decidable. The goal of the present section is to show that it is sufficient to search for a $(GXCD)$ proof such that its expansion into single steps (GX), (C) and (D) has the following properties:

- All nodes are in a particular form (Lemma 8).
- All keys used as hypothesis of a rule (C) or (D) are hypotheses to the macro rule (Lemma 9).

We will show in Section 8 that existence of such a restricted $(GXCD)$ proof is decidable.

7.1 Definitions

We define first an important notion of atom of a term in normal form.

Definition 5 *The set of* atoms *of a term t in normal form is defined by:*

- $atoms(t_1 + t_2) = atoms(t_1) \cup atoms(t_2)$
- $atoms(-t) = atoms(t)$
- $atoms(\{t_1\}_{t_2}) = \{\{t_1\}_{t_2}\} \cup atoms(t_1) \cup atoms(t_2)$
- $atoms(t) = \{t\}$ *if t is not headed with $+$ and not headed with $-$.*

We write $atoms(T)$ for $\bigcup_{t \in T} atoms(t)$, and $atoms(T, t)$ for $atoms(T \cup \{t\})$.

Proposition 5 *For every term t, $atoms(t) \subseteq S(t)$.*

Proposition 6 *Let $atoms(t) = \{a_1, \ldots, a_n\}$. Then there exist $\alpha_1, \ldots, \alpha_n \in \mathbb{Z} \setminus \{0\}$ such that $t = \sum_{i=1}^{i=n} \alpha_i a_i$.*

Proof: These two properties are consequences of Definition 3 of syntactic subterms and the Definition 5 of atoms. □

Obviously there is an instance of the macro rule $(GXCD)$

$$(GXCD)\frac{T \vdash t_1 \qquad \ldots \qquad T \vdash t_n}{T \vdash t}$$

$$
(GX)\dfrac{(C_v)\dfrac{T\vdash x_1 \qquad T\vdash v}{T\vdash \{x_1\}_v} \quad \ldots \quad (C_v)\dfrac{T\vdash x_n \qquad T\vdash v}{T\vdash \{x_n\}_v} \quad (R_1)\dfrac{\vdots}{T\vdash z_1} \quad \ldots \quad (R_m)\dfrac{\vdots}{T\vdash z_m}}{T\vdash \alpha_1\{x_1\}_v + \ldots + \alpha_n\{x_n\}_v + \beta_1 z_1 + \ldots \beta_m + z_m}
$$

$$
\Downarrow
$$

$$
(GX)\dfrac{(C_v)\dfrac{(GX)\dfrac{T\vdash x_1 \quad \ldots \quad T\vdash x_n}{T\vdash \alpha_1 x_1 + \ldots + \alpha_n x_n} \qquad T\vdash v}{T\vdash \alpha_1\{x_1\}_v + \ldots + \alpha_n\{x_n\}_v} \quad (R_1)\dfrac{\vdots}{T\vdash z_1} \quad \ldots \quad (R_m)\dfrac{\vdots}{T\vdash z_m}}{T\vdash \alpha_1\{x_1\}_v + \ldots + \alpha_n\{x_n\}_v + \beta_1 z_1 + \ldots + \beta_m z_m}
$$

Figure 5. Transformation of (C_v)-(GX) into (GX)-(C_v), where $n \geq 2$ and all (R_i) are different from (C_v).

if and only if there exists a proof of $T \vdash t$ using only the rules (A), (GX), (C) and (D).

Definition 6 *A proof of $T \vdash w$ using only (A), (C), (D) and (GX) is called* elementary *if for each node $T \vdash u$ we have that $atoms(u) \subseteq atoms(T, w)$.*

Our goal in this section is to show that whenever there is a proof of $T \vdash w$ using the rules (A), (C), (D) and (GX) then there is an elementary proof of $T \vdash w$.

The following notion is central in establishing the main result of this section. Intuitively, in a $+$-*eager* proof the (GX) rule is applied as early as possible.

Definition 7 *Let P be a flat proof of $T \vdash w$. P is a* $+$-eager *proof if*

(1) for every v there is at most one rule (C_v) with the key v immediately above a (GX) in P,
(2) and there is no rule (D_v) just after a (GX) with a rule (C_v) just above (GX).

Example 4 *We consider the two proofs of $T \vdash w$ given in Example 3. The first proof presented is simple but not $+$-eager since there are two rules (C_k) above a rule (GX), while the second one is $+$-eager and simple.*

We will prove in Lemma 8 that every simple and $+$-*eager* proof is elementary.

7.2 Proof Transformations and Technical Lemmas

Now we present some transformations on proofs used to demonstrate that every proof can be transformed into an elementary one.

Proposition 7 *All the transformations of proofs given in Figures 4, 5 and 6*

$$(D_v)\dfrac{(GX)\dfrac{(R_1)\dfrac{T\vdash B_1}{T\vdash B_1'}\quad\dots\quad(R_n)\dfrac{T\vdash B_n}{T\vdash B_n'}\quad(C_v)\dfrac{T\vdash B\qquad T\vdash v}{T\vdash\{B\}_v}}{T\vdash\alpha_1B_1'+\ldots+\alpha_nB_n'+\alpha\{B\}_v=\{u\}_v}\qquad T\vdash v}{T\vdash u}$$

$$\Downarrow$$

$$(D_v)\dfrac{(GX)\dfrac{(GX)\dfrac{(R_1)\dfrac{T\vdash B_1}{T\vdash B_1'}\quad\dots\quad(R_n)\dfrac{T\vdash B_n}{T\vdash B_n'}}{T\vdash\alpha_1B_1'+\ldots+\alpha_nB_n'=\{c\}_v}\qquad(C_v)\dfrac{T\vdash B\qquad T\vdash v}{T\vdash\{B\}_v}}{T\vdash\{c\}_v+\alpha\{B\}_v=\{u\}_v}\qquad T\vdash v}{T\vdash c+\alpha B=u}$$

$$\Downarrow$$

$$(GX)\dfrac{(D_v)\dfrac{(GX)\dfrac{(R_1)\dfrac{T\vdash B_1}{T\vdash B_1'}\quad\dots\quad(R_n)\dfrac{T\vdash B_n}{T\vdash B_n'}}{T\vdash\alpha_1B_1'+\ldots+\alpha_nB_n'=\{c\}_v}\qquad T\vdash v}{T\vdash c}\qquad T\vdash B}{T\vdash c+\alpha B=u}$$

Figure 6. Elimination of a rule (D_v) after a (GX) with a rule (C_v) just above the (GX) (C_k)-(GX)-(D_k), with $n \geq 2$.

decrease the number of nodes.

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 with the same hypotheses and the same conclusion.

In Figure 4 it is obvious.

In Figure 5 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 6, we decompose the message $\{u\}_v$ into two parts $\{u\}_v = \{c\}_v + \alpha\{B\}_v$, where the term $\{c\}_v$ represents the sum of all terms that compose the term $\{u\}_v$ except the term B just encrypted by the key v. Using this decomposition, we can apply the decryption rule earlier, and obtain a new proof of $T \vdash u$. Hence, 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. We deduce that the number of nodes decreases. □

Lemma 6 *If there is a proof of $T \vdash w$ then there is also a +-eager and simple proof of $T \vdash w$.*

Proof: Let P be a proof of $T \vdash w$. The transformation rules given in Figures 4, 5 and 6 decrease $|P|$ as well as the transformation to get a simple rule. Therefore the application of rules eventually terminates with a +-*eager* simple proof of $T \vdash w$. □

$$
\dfrac{
  (GX)\dfrac{
    (R_1)\dfrac{\vdots}{T \vdash u_1} \quad \dots \quad (R_n)\dfrac{\vdots}{T \vdash u_n}
  }{T \vdash \{u\}_v\!\downarrow}
  \qquad
  \dfrac{\vdots}{T \vdash v\!\downarrow}
}{T \vdash u\!\downarrow}(D_v)
$$

Figure 7. Illustration of the case (D) in Lemma 7

Lemma 7 *Let P be a $+$-eager and simple proof of $T \vdash u$ of the form*

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

Then $atoms(\{u\}_v) \subseteq atoms(T)$.

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): the result is true by definition of the rule (A).
- (R) is some rule (C): this cannot happen since either (C) is (C_v) and P is not simple or (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 atoms$(\{\{u\}_v\}_{v'}) \subseteq$ atoms(T), yielding by definition of atoms that atoms$(\{u\}_v) \subseteq$ atoms(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 $\{u\}_v\!\downarrow$.

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

  · (R_i) is (A), hence $a \in$ atoms(T).

  · (R_i) is $(D_{v'})$ s.t. $(D_{v'})\ \dfrac{T \vdash \{w_1\}_{v'}\ T \vdash v'}{T \vdash w_1 = u_i}$.

  By induction hypothesis atoms$(\{w_1\}_{v'}) \subseteq$ atoms(T), therefore by definition of atoms we conclude that atoms$(u_i) \subseteq$ atoms(T) and $a \in$ atoms(T).

  · (R_i) is (C_v) or (GX): Impossible since the proof is $+$-*eager* and flat.

  · (R_i) is $(C_{v'})$ with $v \neq v'$. Then $u_i = \{u'\}_{v'}\!\downarrow$. Since $v' \neq v$ none of atoms$(\{u'\}_{v'}\!\downarrow)$ can be equal to a, all these atoms are canceled out by

other occurrences of the same atom in one of the u_j with $j \neq i$. Since the proof is $+$-*eager* and flat it is impossible that the other terms stem from the rule (GX) or the rule (C), consequently the other terms stem only from a rule (A) or (D). In the first case atoms are obviously in atoms(T), in the second case we apply the induction hypothesis. We conclude that atoms$(u_i) \subseteq$ atoms(T). $\square$

7.3 Elementary Proofs

Now we have all ingredients to demonstrate the existence of an elementary proof, and as consequence we prove some lemmas on the keys used in the proof which will be employed in the next section.

Lemma 8 *Every simple and $+$-eager proof is elementary.*

Proof: We proceed by structural induction on the proof P and case distinction of the last rule (R) of P a simple and $+$-*eager* proof of $T \vdash u$:

- (R) is (A): P is obviously an elementary proof.
- (R) is some rule (D) s.t. $\dfrac{T \vdash \{u\}_v \quad T \vdash v}{T \vdash u}$. The induction hypothesis yields that the proof P_1 which yields $T \vdash \{u\}_v$ and the proof P_2 which yields $T \vdash v$ are elementary, that is that for all nodes $T \vdash w$ in P_1, resp. P_2, we have that atoms$(w) \subseteq$ atoms$(T, \{u\}_v)$, resp. atoms$(w) \subseteq$ atoms(T, v). Since P is $+$-*eager* we obtain with Lemma 7 that atoms$(\{u\}_v) \subseteq$ atoms$(T) \subseteq$ atoms(T, w). We conclude that atoms$(w) \subseteq$ atoms(T, u) for all $T \vdash w$ in P.
- (R) is some rule (C): We have that $u = \{u_1\}_{u_2}$. (R) is some (C) s.t. $\dfrac{T \vdash u_1 \quad T \vdash u_2}{T \vdash \{u_1\}_{u_2}}$. The induction hypothesis yields that the proof P_1 which yields $T \vdash u$ and the proof P_2 which yields $T \vdash v$ are elementary, that is that for all nodes $T \vdash w$ in P_1, resp. P_2, we have that atoms$(w) \subseteq$ atoms(T, u_1), resp. atoms$(w) \subseteq$ atoms(u_2). We conclude by the fact that atoms$(T, u_1) \subseteq$ atoms$(T, \{u_1\}_{u_2})$ and atoms$(T, u_2) \subseteq$ atoms$(T, \{u_1\}_{u_2})$.
- (R) is some rule (GX) such that

$$(GX)\ \dfrac{(R_1)\dfrac{\vdots}{T \vdash u_1} \quad \ldots \quad (R_n)\dfrac{\vdots}{T \vdash u_n}}{T \vdash u}$$

By induction hypothesis, each of the proofs P_i yielding $T \vdash u_i$ is elementary, that is for each $T \vdash w$ in P_i we have that atoms$(w) \subseteq$ atoms(T, u_i). In order to conclude we will show that atoms$(u_i) \subseteq$ atoms(T, u) for every i. We proceed by case distinction on the last rule (R_i) of the proof P_i.

· (R_i) is (GX): Impossible since P is $+$-*eager*, hence flat.
· (R_i) is (A), (D): By Lemma 7, $\text{atoms}(u_i) \subseteq \text{atoms}(T) \subseteq \text{atoms}(T, u)$.
· (R_i) is (C_k): Let $a \in \text{atoms}(u_i)$, and assume that $a \notin \text{atoms}(T, u)$. Since u_i is obtained by application of rule (C_k) we have that a is of the form $\{a'\}_k$.

Since $a \notin \text{atoms}(u)$ it must be the case that a is "canceled out" by some other term u_j, $j \neq i$, with $a \in \text{atoms}(u_j)$. Since we assumed that $a \notin \text{atoms}(T)$ we conclude as above that the last rule (R_j) of the proof P_j cannot be (A), (D), or (C). Hence, R_j is some rule (C), and by the fact that $a = \{a'\}_k$ we obtain that R_j is (C_k). This means that we have two distinct applications of (C_k) above (GX), which contradicts the assumption that P is $+$-*eager*. □

Definition 8 *A term v is in* key position *of a term in normal form w if $\{t\}_v \in S(w)$ for some term t.*

Lemma 9 *Let P be a $+$-eager and simple proof of $T \vdash w$. All terms occurring in key position of some node of P are in $S(T, w)$.*

Proof: If a term k appears in key position of a term t occurring in the proof P then it is syntactic subterm of an atom of t, and hence by Lemma 8 a subterm of some atom of T, w. By Proposition 5 we obtain $k \in S(T, w)$. □

Lemma 9 allows us to obtain a refined version of the locality theorem obtained above: not only do we obtain locality when we cluster successive applications of (GX), (C) and (D), but we even have locality for the following refined "macro" rule:

Definition 9 *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.

Again, the reasoning performed in this section applies also to the case of the *exclusive-or*.

8 Deciding *GCD*-Deducibility in the General Case

Our goal in this section is to decide elementary deducibility.

Definition 10 *We say that a term w is* elementary deducible *from a finite set L of terms if there exists a* GCD-proof tree *with set of leaves $L' \subseteq L$, set of keys $K \subseteq L$, and root w, and such that for all nodes $T \vdash t$ of the proof we have that $atoms(t) \subseteq atoms(L, w)$.*

We first demonstrate the *exclusive-or* case which is an immediate consequence of the result on elementary proofs. Second we decide elementary deducibility for the more complex case of Abelian group using the mathematical notion of $\mathbb{Z}$-module.

8.1 The Exclusive-Or *Case*

This case is easy: by definition, the nodes of the GCD-proof trees have the form $a_1 + \ldots + a_n$ where $a_i \in \text{atoms}(L, w)$ and $a_i \neq a_j$ for $i \neq j$. Therefore, there are only exponentially (in the size of $|L| + |w|$) many possible nodes, hence only a finite number of possible proofs. The one-step deducibility for the rule (GX) in this case is equivalent to solving linear Diophantine equations over $\mathbb{Z}/2\mathbb{Z}$. This approach is similar to the method used by [12,13] to prove one-step deducibility by the rule (GX).

8.2 The Abelian Group Case

The above reasoning does not apply in the Abelian group case since there is no a priori bound on the coefficients of a sum $\Sigma_{i=1}^{i=n} \alpha_i a_i$. Let

$$\text{atoms}(L, w) = \{a_1, \ldots, a_n\}$$

We call an (L, w)*-elementary term* a term t such that $\text{atoms}(t) \subseteq \text{atoms}(L, w)$. By Proposition 6 an (L, w)-elementary term t can be written in the form $\alpha_1 a_1 + \ldots + \alpha_n a_n$ with $\alpha_i \in \mathbb{Z}$. We define the representation of t as $\overline{t} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}^n$. Obviously, $\overline{t_1 + t_2} = \overline{t_1} + \overline{t_2}$, and $\overline{-t} = -\overline{t}$. Furthermore, by definition, $\overline{a_i} = e_i$ where e_i is the i-th unit vector.

We recall that a $\mathbb{Z}$-*module*, or simply *module*, is an Abelian group M equipped with an operation of scalar multiplication such that $\alpha(x+y) = \alpha x + \alpha y$ and $\alpha(-x) = -\alpha x$ for all $\alpha \in \mathbb{Z}$ and $x, y \in M$. Here we are interested in the module $\mathbb{Z}^n$, where scalar multiplication with an integer is defined as usual, and sub-modules thereof.

For $x_1, \ldots, x_m \in \mathbb{Z}^n$ we denote by $\langle x_1, \ldots, x_m \rangle$ the sub-module of $\mathbb{Z}^n$ generated by $x_1, \ldots, x_m$, that is

$$\langle x_1, \ldots, x_m \rangle = \{\alpha_1 x_1 + \ldots \alpha_m x_m \mid \alpha_i \in \mathbb{Z}\}$$

It is of course decidable whether $y \in \langle x_1, \ldots, x_m \rangle$ for given $y, x_1, \ldots, x_m$ since deciding this question amounts to solving a system of linear equations over $\mathbb{Z}$. We will construct, for any given finite set L and term w, a finite set of generators for the set of (L, w)-elementary terms that are elementarily deducible from L. This will be achieved by a fixed point construction. In order to guarantee that the fixed point is reached in a finite number of steps we need an operation which allows us to extend a submodule by a new generator, but in such a way that the new generator is only added if necessary, and such that the generator is "small". In order to make this notion of "small" precise we write for $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathbb{Z}^n$ that $x \trianglelefteq y$ if $|x_i| \leq |y_i|$ for all i, and $x \triangleleft y$ if $x \trianglelefteq y$ and $x \neq y$.

Definition 11 *We define the relation $Ext(l_1, x, l_2)$ where l_1 and l_2 are finite lists over $\mathbb{Z}^n$ and $x \in \mathbb{Z}^n$ by: $Ext((x_1, \ldots, x_n), x, l)$ holds in any of the following cases:*

- *$x \in \langle x_1, \ldots, x_n \rangle$ and $l = (x_1, \ldots, x_n)$.*
- *$x \notin \langle x_1, \ldots, x_n \rangle$ and $l = (x_1, \ldots, x_n, y)$ where y is minimal in the order $\trianglelefteq$ such that $y \in x + \langle x_1, \ldots, x_n \rangle$.*

Note that in the second case of this definition there may be different choices for y, These choices differ only in the sign of the components.

We recall that Presburger arithmetic, that is the first-order theory of $(\mathbb{Z}, +, >)$, is decidable. The absolute value of $y \in \mathbb{Z}$ is definable by a formula, that is $z = |y|$ can be expressed as $(y \geq 0 \rightarrow z = y) \wedge (y < 0 \rightarrow z = -y)$. As a consequence, the formula defining $x \trianglelefteq y$ is a also a formula of Presburger arithmetic. Membership $x \in \langle x_1, \ldots, x_n \rangle$ is expressible by $\exists \alpha_1, \ldots, \alpha_n \ (x = \Sigma_i \alpha_i x_i)$. Moreover $x \notin \langle x_1, \ldots, x_n \rangle$ is simply the negation of the membership formula. The set of minimal elements w.r.t. $\trianglelefteq$ in a set $S \subseteq \mathbb{Z}^n$ defined by a Presburger formula $x \in S$ is defined by $x \in S \wedge \neg(\exists y \ (y \in S \wedge y \triangleleft x))$. Putting all these formulas together, we get a definition of all l_2 such that $Ext(l_1, x, l_2)$ holds for given l_1 and x. From this formula, one can easily compute one particular l_2.

We recall Dickson's classical lemma [36]. Note that in this lemma tuples are tuples of *natural* numbers.

Lemma 10 (Dickson's lemma) *For each infinite sequence $t_1, t_2, \ldots$ of distinct n-tuples of $\mathbb{N}$ there exists $i < j$ such that $t_i \lhd t_j$.*

We can now show that there is no infinite chain in the relation Ext:

Lemma 11 *Let $x_1, \ldots, x_i, \ldots \in \mathbb{Z}^n$ be an infinite sequence, l_0 the empty sequence, and $Ext(l_i, x_{i+1}, l_{i+1})$ for every i. Then there exists an i such that $l_i = l_j$ for every $j \geq i$.*

Proof: Assume that the sequence $(l_i)_{i \in \mathbb{N}}$ is not eventually stationary. There exists an infinite subsequence of $(l_i)_{i \in \mathbb{N}}$ such that its respective last elements have the same sign (since there are only 2^n many possible signs). It follows that the chosen subsequence is not eventually stationary either.

We can decompose for any $i : l_i = l'_i \cdot x'_i$ where x'_i is the last element of l_i. By Dickson's lemma there are l_i, l_j in this subsequence with $i < j$ and $x'_i \unlhd x'_j$. By construction, l_i is a prefix of the sequence l'_j, and hence $x'_i \in \langle l'_j \rangle$. Since $x'_j \in x_j + \langle l'_j \rangle$ for some x_j, this means that $x'_j - x'_i \in x_j + \langle l'_j \rangle$ by definition of $\langle l'_j \rangle$. Since x'_i and x'_j have the same sign, $x'_j - x'_i$ is strictly smaller than x'_j in the order $\unlhd$. This contradicts the minimality of x'_j in the definition of Ext. $\square$

We use the $\mathbb{Z}$-modules to model all the terms that the rule (GX) can construct. Now, we analyze more precisely in Lemma 12 how the generators of a $\mathbb{Z}$-module are modified by the application of a rule (D_k).

Lemma 12 *Let $g^1, \ldots, g^m \in \mathbb{Z}^n$. We can compute a finite set of generators of*

$$D_k(g^1, \ldots, g^m) = \{\overline{t} \mid \overline{\{t\}_k} \in \langle g^1, \ldots, g^m \rangle\}$$

Proof: First we calculate a set of generators of

$$K_k(g^1, \ldots, g^m) = \{\overline{\{t\}_k} \mid \overline{\{t\}_k} \in \langle g^1, \ldots, g^m \rangle\}$$

Let I_k be the set of indices corresponding to atoms encrypted by the key k, that is

$$I_k = \{i \mid a_i = \{a_j\}_k \text{ for some } j\}$$

where the $(a_1, \ldots, a_n)$ is the enumeration of atoms(L, w) chosen at the beginning of the section. An (L, w)-elementary term t, whose representation $\overline{t}$ is $(\alpha_1, \ldots, \alpha_n)$, is of the form $\{u\}_k$ iff $\alpha_i = 0$ for every $i \notin I_k$. Hence, $(\alpha_1, \ldots, \alpha_n) \in K_k(g^1, \ldots, g^m)$ iff

$$\begin{cases} (\alpha_1, \ldots, \alpha_n) = \beta_1 g^1 + \ldots + \beta_m g^m \\ \text{and } \alpha_i = \beta_1 g_i^1 + \ldots \beta_m g_i^m = 0 \text{ for every } i \in I_k \end{cases}$$

The set of m-tuples $(\beta_1, \ldots, \beta_m)$ satisfying this system of equations forms a sub-module of $\mathbb{Z}^m$. We can compute a finite set B of generators of this submodule as follows. Each β_i can be written $\gamma_i - \delta_i$ with $\gamma_i, \delta_i \geq 0$. Then the equations $\beta_1 g_i^1 + \ldots \beta_m g_i^m = 0$ for $\in I_k$ define an homogeneous system (H) where the unknowns γ_i, δ_i belong to $\mathbb{N}$. Therefore a 2m-tuple $(\gamma_1, \ldots, \gamma_m, \delta_1, \ldots, \delta_m)$ is a solution of (H) iff it is a linear combination with coefficients in $\mathbb{N}$ of the (finitely many) minimal solutions of (H). Any linear combination with coefficients in $\mathbb{Z}$ is also a solution of (H).

Therefore, any solution β can be expressed as a linear combination with coefficient in $\mathbb{Z}$ of elements of the finite set

$$B = \{(\gamma_1^\mu - \delta_1^\mu, \ldots, \gamma_m^\mu - \delta_m^\mu) \mid (\gamma_1^\mu, \ldots, \gamma_m^\mu, \delta_1^\mu, \ldots, \delta_m^\mu) \text{ minimal solution of (H)}\}$$

Conversely, any linear combination with coefficient in $\mathbb{Z}$ of elements of B is a solution of the original set of equations.

This proves that B is a finite set of generators.

Then we obtain that

$$G = \{b_1 g^1 + \ldots + b_m g^m \mid (b_1, \ldots, b_m) \in B\}$$

is a finite set of generators of $K_k(g^1, \ldots, g^m)$.

We finally obtain a finite set of generators of $D_k(g^1, \ldots, g^m)$ by "shifting" the elements of G. Let $\mathit{shift}_k(\alpha_1, \ldots, \alpha_n)$ be the vector $(\beta_1, \ldots, \beta_n)$ defined by

$$\beta_i = \begin{cases} \alpha_j & \text{if } \{a_i\}_k = a_j \in \text{atoms}(L, w) \\ 0 & \text{if } \{a_i\}_k \notin \text{atoms}(L, w) \end{cases}$$

The finite set of generators of $D_k(g^1, \ldots, g^m)$ is $\mathit{shift}_k(G)$. $\square$

Lemma 13 below is the analog of Lemma 12 for the rule (D_k).

Lemma 13 *Let $g^1, \ldots, g^m \in \mathbb{Z}^n$. We can compute a finite set of generators of*

$$C_k(g^1, \ldots, g^m) = \{\overline{\{t\}_k} \mid \overline{t} \in \langle g^1, \ldots, g^m \rangle, \mathit{atoms}(\{t\}_k) \subseteq \mathit{atoms}(L, w)\}$$

Proof: The proof is analogous to the proof of Lemma 12. We now construct first a finite set B of generators of the set

$$K_k^{-1}(g^1, \ldots, g^m) = \{\overline{t} \mid \overline{t} \in \langle g^1, \ldots, g^m \rangle \text{ and atoms}(\{t\}_k) \subseteq \text{atoms}(L, w)\}$$

that is of the set of terms whose encryption with k is again an (L, w) elementary term. The finite set of generators of $C_k(g^1, \ldots, g^m)$ is obtained as

$shift_k^{-1}(B)$, where $shift_k^{-1}(\alpha_1, \ldots, \alpha_n)$ is the vector $(\beta_1, \ldots, \beta_n)$ defined by

$$\beta_i = \begin{cases} \alpha_j & \text{if } a_i = \{a_j\}_k \text{ with } a_j \in \text{atoms}(L, w) \\ 0 & \text{if } a_i \text{ is not of the form } \{a_j\}_k \text{ with } a_j \in \text{atoms}(L, w) \end{cases}$$

□

Lemma 14 *We can compute, given a finite set L of terms and a term w, a finite set of generators of the set of (L, w)-elementary terms that are elementarily deducible from L.*

Proof: Let, as above, $\text{atoms}(L, w) = \{a_1, \ldots, a_n\}$, and $L = \{t_1, \ldots, t_p\}$. We define a relation Φ between finite lists of vectors as follows:

Given a finite set l of vectors, let $x_1, \ldots, x_p$ be the union of the finite sets of generators of $D_k(l)$ with $k \in L$ computed according to Lemma 12, and let $x_{p+1}, \ldots, x_q$ be the union of the finite sets of generators of $C_k(l)$ with $k \in L$ computed according to Lemma 13. Then $\Phi(l, l')$ holds iff there are finite lists of vectors $l_0, \ldots, l_q$ such that $l = l_0$, $l' = l_q$, and $Ext(l_{i-1}, x_i, l_i)$ holds for all i.

Let $l_0 = (\overline{t_p}, \ldots, \overline{t_p})$, and $\Phi(l_i, l_{i+1})$ for any i. By Lemma 11, there exists an i such that $l_{i+1} = l_i$. By construction, this l_i is the finite set of generators of the set of (L, w)-elementary terms that are elementarily deducible from L. □

9 Decidability of the Intruder Deduction Problem in the General Case

We sum up the results obtained so far to state our main results for the general case.

Theorem 15 *The intruder deduction problem can be decided in EXPTIME for the theory of* exclusive-or *with distributive encryption.*

Proof: By Lemmas 9,8 and the decidability of elementary deducibility, we get that there is a proof iff there is a proof such that all nodes belong to $ST_+(T, w) = \{a_1 + \ldots + a_p \mid a_i \in \text{atoms}(T, w)\ a_i \neq a_j, i, j = 1, \ldots, p\}$ which has an exponential size in $|T| + |w|$. □

Theorem 16 *The intruder deduction problem is decidable for the theory of Abelian groups with distributive encryption.*

Proof: By Lemmas 9,8 and the decidability of elementary deducibility, we get that there is a proof consisting of applications of rules (A), (UR), (UL), (P) and $(GXCD)$ for which the premises and the conclusions belong to $S(T, w)$.

This proof has polynomial size in $|T| + |w|$, but we cannot give a polynomial complexity result since deciding elementary deducibility for the rule $(GXCD)$ relies on Dickson's lemma. □

10 Decidability of the Intruder Deduction Problem in the Binary Case

We call a term in normal form *top-binary* if it is the difference of two different terms not headed with $+$ or with $-$, and *at most binary* if all its syntactic subterms are either top-binary or not headed with $+$ or $-$. Note that for example $a - b$ is at most binary, while $2a$ and $3a - 2b$ are not.

A set is *at most binary* if each of its elements is. Note that if the set T is at most binary then $S(T)$ is at most binary as well. A proof tree P is called *at most binary* if for all its nodes $T \vdash u$ the term u is at most binary.

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. By Lemma 5 it is sufficient to show that one-step deducibility by the rule $(GXCD)$ is decidable in polynomial time in case the hypotheses and the conclusion are at most binary.

The decision algorithm for one-step deducibility by $(GXCD)$ in the binary case is obtained by analyzing the proof trees that make up the "macro rule" $(GXCD)$. In the general case, this was done by showing that we can restrict the search of such a proof tree to elementary trees (Section 7), and then that existence of such an elementary proof tree can be decided by algebraic means (Section 8). In the binary case we will show that we can restrict the search of proof trees to proof trees which are at most binary (Subsection 10.1), and then that existence of such a proof tree can be decided by using methods of prefix rewriting (Subsection 10.2).

10.1 At Most Binary Proofs

In this section we will show how to transform such a proof into a proof which is at most binary.

Example 5 *The following (GCD) proof tree of* $\{c\}_k - \{b\}_k$ *with* $L = \{a -$

$b, c-d, \{d\}_k - \{a\}_k\}$ and $K = \{k\}$ is not at most binary:

$$(GX)\dfrac{(C)\dfrac{(GX)\dfrac{T \vdash a-b \qquad T \vdash c-d}{T \vdash a-b+c-d} \qquad T \vdash k}{T \vdash \{a\}_k - \{b\}_k + \{c\}_k - \{d\}_k} \qquad T \vdash \{d\}_k - \{a\}_k}{T \vdash \{c\}_k - \{b\}_k}$$

However, there is a proof tree which is at most binary:

$$(GX)\dfrac{(C)\dfrac{T \vdash a-b \qquad T \vdash k}{T \vdash \{a\}_k - \{b\}_k} \quad (C)\dfrac{T \vdash c-d \qquad T \vdash k}{T \vdash \{c\}_k - \{d\}_k} \quad T \vdash \{d\}_k - \{a\}_k}{T \vdash \{c\}_k - \{b\}_k}$$

Definition 12 *For a set U of terms we denote by $\langle U \rangle$ the set of terms*

$$\langle U \rangle = \{(\alpha_1 u_1 + \ldots + \alpha_n u_n)\downarrow \mid \alpha \in \mathbb{Z}, u_i \in U\}$$

In other words, $\langle U \rangle$ is the set of terms in normal form that can be obtained from some subset of U by applying the rule (GX).

Proposition 8 *Let U be a finite set of at most binary terms and $u \in \langle U \rangle$. Then there exist at most binary terms $v_1, \ldots, v_k \in \langle U \rangle$ with $u \in \langle v_1, \ldots, v_k \rangle$ and $atoms(v_1, \ldots, v_k) \subseteq atoms(u)$.*

Proof: The proof is by induction on the cardinality of U. If $u = 0$ then we choose $k = 0$. Otherwise there exists an $a \in \text{atoms}(u)$, and a term $u_0 \in U$ such that $a \in \text{atoms}(u_0)$. Let α be the factor with which u_0 contributes to the construction of u, that is $u = \alpha u_0 + u'$ with $u' \in \langle U \setminus \{u_0\} \rangle$.

By induction hypothesis there are at most binary terms $v'_1, \ldots, v'_l \in \langle U \setminus \{u_0\} \rangle$ such that $u' \in \langle v'_1, \ldots, v'_l \rangle$ and $\text{atoms}(v'_1, \ldots, v'_l) \subseteq \text{atoms}(u')$.

There are three cases:

(1) u_0 is not headed with $+$, that is $u_0 = a$. We choose $k = l+1$, $v_i = v'_i$ for $i < k$, and $v_k = u_0$. Since $u_i \in \langle U \setminus \{u_0\} \rangle$ for $i < k$ we obviously also have $u_i \in \langle U \rangle$ for $i < k$, and $v_k \in \langle U \rangle$ holds since $u_0 \in U$.
   By construction we have that $u \in \langle v_1, \ldots, v_k \rangle$, and since $a \in \text{atoms}(u)$ we have that $\text{atoms}(u) = \text{atoms}(u') \cup \{a\}$. Hence, $\text{atoms}(v_i) \subseteq \text{atoms}(u') \subseteq \text{atoms}(u)$ for $i < k$ by induction hypothesis, and $\text{atoms}(v_k) = \{a\} \subseteq \text{atoms}(u)$ by choice of a.

(2) $u_0 = a - b$ or $u_0 = b - a$ for $b \in \text{atoms}(u)$. This case is similar to the first case: We choose $k = l+1$, $v_i = v'_i$ for $i < k$, and $v_k = u_0$. As above, $u_i \in \langle U \rangle$ for $i \leq k$. Also, we have again that $u \in \langle v_1, \ldots, v_k \rangle$. Concerning the atoms of u we now have that $\text{atoms}(u) = \text{atoms}(u') \cup \{a, b\}$. Hence, $\text{atoms}(v_1, \ldots, v_k) \subseteq \text{atoms}(u)$.

(3) $u_0 = a - b$ or $u_0 = b - a$ for some atom b, and $b \notin \text{atoms}(u)$. We can assume w.l.o.g. that $u_0 = a - b$.
We can write every v'_i with $1 \leq i \leq l$ as $v'_i = \lambda_i a + \gamma_i b + v''_i$ with $a, b \notin \text{atoms}(v''_i)$. We choose $k = l$, and $v_i = (\lambda_i + \gamma_i)a + v''_i$ for $1 \leq i \leq k$.
For any i there can be at most two atoms in v''_i since v'_i is at most binary.
- If v''_i contains two atoms then $\lambda_i = \gamma_i = 0$, and $v_i = v'_i$ is at most binary.
- Otherwise, if v''_i contains at most one atom, then v is at most binary since $b \notin \text{atoms}(v_i)$.

For any i, $v'_i + \gamma_i u_0 = (\lambda_i + \gamma_i)a + v''_i = v_i$, and hence $v_i \in \langle U \rangle$.
By construction, $\text{atoms}(v_i) \subseteq \text{atoms}(v'_i) \cup \{a\} \setminus \{b\} \subseteq \text{atoms}(u)$.
It remains to show that $u \in \langle v_1, \ldots, v_k \rangle$. Let $u' = \Sigma_{i=1}^{i=n} \alpha_i v'_i$. First note that, since $u = \alpha u_0 + \Sigma_{i=1}^{i=n} \alpha_i v'_i = \alpha a - \alpha b + \Sigma_{i=1}^{i=n} \alpha_i(\lambda_i a + \gamma_i b + v''_i)$ and $b \notin \text{atoms}(u, v''_1, \ldots, v''_n)$ that $\alpha = \Sigma_{i=1}^{i=n} \alpha_i \gamma_i$. Now, we have that

$$\Sigma_{i=1}^{i=n} \alpha_i v_i = \Sigma_{i=1}^{i=n} \alpha_i (v'_i + \gamma_i u_0) = \Sigma_{i=1}^{i=n} \alpha_i v'_i + (\Sigma_{i=1}^{i=n} \alpha_i \gamma_i) u_0 = u' + \alpha u_0 = u \ \Box$$

Lemma 17 *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

$$\text{(GX)}\ \dfrac{\text{(C,D)}\ \dfrac{\text{(C,D)}\ \dfrac{\text{(GX)}\ \dfrac{T \vdash u_1 \quad \cdots \quad T \vdash u_n}{T \vdash u}\ (\in \langle u_1, \ldots, u_n \rangle)}{\vdots}}{T \vdash u'} \qquad P_1 \quad \cdots \quad P_n}{P}$$

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 8 there are at most binary terms $v_1, \ldots, v_k \in \langle u_1, \ldots, u_n \rangle$ such that $u \in \langle v_1, \ldots, v_k \rangle$ and $\text{atoms}(v_1, \ldots, v_k) \subseteq \text{atoms}(u)$. We hence obtain, where we abbreviate by $T \vdash U$ the set of sequents

$\{T \vdash z \mid z \in U\}$:

$$
\text{(GX)}\ \cfrac{\text{(GX)}\ \cfrac{\text{(C,D)}\ \cfrac{\cfrac{\text{(GX)}\ \cfrac{T \vdash U}{T \vdash v_1} \quad \cdots \quad \text{(GX)}\ \cfrac{T \vdash U}{T \vdash v_n}}{T \vdash u \quad (\in \langle v_1, \ldots, v_n \rangle)}}{\vdots}}{\text{(C,D)}\ \overline{T \vdash u'}} \qquad P_1 \quad \cdots \quad P_n}{P}
$$

In this proof tree we hence have that $v_i \in \langle U \rangle$ is at most binary for every i. We can now apply the inverse transformation of Figure 5 and commute the (GX) rule with succeeding (C) rules. Since $\text{atoms}(v_i) \subseteq \text{atoms}(u)$ for $1 \leq i \leq k$ we can also commute this (GX) rule with succeeding (D) rules. We hence obtain:

$$
\text{(GX)}\ \cfrac{\begin{array}{c} \text{(GX)}\ \cfrac{T \vdash U}{T \vdash v_1} \\ \text{(C,D)}\ \overline{\quad\vdots\quad} \\ \text{(C,D)}\ \overline{T \vdash u'_1} \end{array} \qquad \begin{array}{c} \text{(GX)}\ \cfrac{T \vdash U}{T \vdash v_n} \\ \text{(C,D)}\ \overline{\quad\vdots\quad} \\ \text{(C,D)}\ \overline{T \vdash u'_n} \end{array} \qquad P_1 \quad \cdots \quad P_n}{P}
$$

where $u' \in \langle u'_1, \ldots, u'_n \rangle$. 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. $\Box$

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

Definition 13 *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.*

Definition 14 *Let, for any set T of terms in normal form,*

$$S_{+2}(T) = \mathit{atoms}(T) \cup \{a_1 - a_2 \mid a_1, a_2 \in \mathit{atoms}(T), a_1 \neq a_2\}$$

Lemma 18 *Let P be a GCD-proof tree with leaves L, keys K, and root r. If $L \cup \{r\} \in S_{+2}(T)$ for some set of terms T then there exists a proof tree using exclusively the (GCD) rule such that all nodes are in $S_{+2}(T)$.*

Proof: By Lemma 6 there is a simple and and $+$-*eager* GCD-proof tree P'. We now apply the transformation of the proof of Lemma 17. Since P' is simple 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

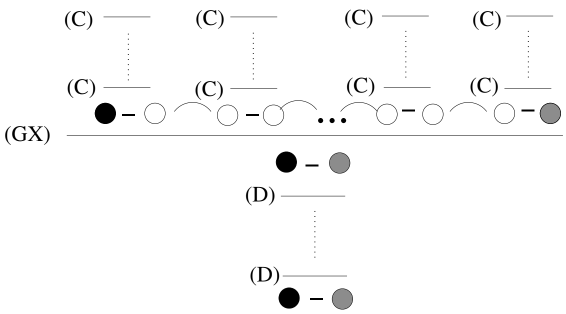

Figure 8. Illustration of the third case

sequence of (D) rule applications. Since $u \in S_{+2}(T)$ by hypothesis, we also have that $v_i \in S_{+2}(T)$ since for all i we have that $atoms(v_) \subseteq$ atoms(u) and v_i is at most binary. As a consequence, any term obtained by a sequence of decryptions form v_i is also in $S_{+2}(T)$. $\square$

10.2 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 $+$, and there is a sequence of top-binary terms $(\{a_i\}_{v_i} - \{b_i\}_{v_i})_{i=1,\ldots,N}$ such that for every i one of $\{a_i\}_{v_i} - \{b_i\}_{v_i}$ and $\{b_i\}_{v_i} - \{a_i\}_{v_i}$ is in U, a term $a_{N+1} \in U$ not headed with $+$, and a sequence $(h_i)_{i=0,\ldots,N+1}$ 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 $i = 1, \ldots, N$.
(2) r is a top-binary term $r_1 - r_2$, 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_i.
(3) r is a top-binary term $r_1 - r_2$, and there is a sequence of top-binary terms $(\{a_i\}_{v_i} - \{b_i\}_{v_i})_{i=1,\ldots,N}$ such that for every i one of $\{a_i\}_{v_i} - \{b_i\}_{v_i}$ and $\{b_i\}_{v_i} - \{a_i\}_{v_i}$ is 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 $\{b_N\}_{v_N h_N} = \{r_2\}_{h_0}$.

In the following we will only consider the last case, which is illustrated by Figure 8, since the first two cases can be checked in a very similar way. In this figure the sequence of encryption keys is not displayed. We have a binary term as the conclusion of the rule (GX), to which a sequence of decryptions

(D) is applied, and a sequence of binary terms as hypotheses of the rule (GX) each of which is obtained by a sequence of encryptions (C). In this sequence of terms, the atoms are canceled out in pairs (indicated by the arcs in the figure), leaving only the very first and the very last one to form the result of (GX).

The idea is to reduce the problem to reachability in a prefix rewrite system [37]. Let us first explain the construction at hand of a special case. We view a term $\{a\}_{xyz}$, where a is not headed with $+$ and not of the form $\{m\}_k$, as the term $axyz$. 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 xyz, 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, for any binary term $\{a\}_v - \{b\}_w \in L$, rewrites any term avx into bwx, and vice versa:

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

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

Example 6 *Consider $U = \{a - \{b\}_{12}, \{b\}_1 - \{c\}_3, 1, 2, 3, 4\}$, and $r = \{a\}_4 - \{c\}_{324}$ (in this, as in the following example, we use numbers for keys). There is a (GCD) proof tree with leaves and keys and L and root r:*

$$(GX)\dfrac{(C_4)\dfrac{T \vdash a - \{b\}_{12}}{T \vdash \{a\}_4 - \{b\}_{124}} \qquad (C_4)\dfrac{(C_2)\dfrac{T \vdash \{b\}_1 - \{c\}_3}{T \vdash \{b\}_{12} - \{c\}_{32}}}{T \vdash \{b\}_{124} - \{c\}_{324}}}{T \vdash \{a\}_4 - \{c\}_{324}}$$

The prefix rewrite system obtained from L is

$$\{a \to b12, b12 \to a, b1 \to c3, c3 \to b1\}$$

With this rewrite system we have the rewrite sequence

$$a4 \mapsto b124 \mapsto c324$$

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 when $x \in U^*$, that is when x is a sequence of symbols from 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 $\mathit{left}(x)$ and $\mathit{right}(x)$, for any string x, be such that $x = \mathit{left}(x) \cdot \mathit{right}(x)$, and such that $\mathit{right}(x)$ 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 #:

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

Example 7 *We apply this refined construction to the variant obtained from Example 6 by removing the key* 1 *from set* U*. Then we obtain the same (GCD) proof tree as in Example 6 since this proof tree does not use an encoding with the key* 1*. The rewrite system is now*

$$\{a\# \to b1\#2, b1\#2 \to a, b1\# \to c\#3, c\#3 \to b1\#\}$$

With this rewrite system we have the rewrite sequence

$$a\#4 \mapsto b1\#24 \mapsto c\#324$$

However, if we remove the key 2 *as well from the set* U *then the above proof tree is not legitimate (since we can not encode with* 2*). The rewrite system in this case is*

$$\{a\# \to b12\#, b12\# \to a, b1\# \to c\#3, c\#3 \to b1\#\}$$

With this system, we can rewrite $a\#4$ *into* $b12\#$*, but then we can no longer apply the rewrite rule* $b1\# \to c\#3$*.*

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 15 *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* $+$ *and not of the form* $\{x\}_y$*.*

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

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

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

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

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

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

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

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

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. Note that we have in the first case a rewrite rule for every decomposition of $right(v)$ into v_1 and v_2, and in the second case a rewrite rule for every decomposition of $right(w)$ into w_1 and w_2.

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

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

Example 9 *Let* $U = \{\{a\}_{12} - b, \{b\}_{34} - c, c - \{d\}_{234}, 1, 2, 3, 4\}$, *and* $r = a - \{d\}_1$. *We only give the rewrite rules which are relevant for this example: The system* PR_1 *contains, among others, the rules* $a\#\bot \to b\#\overline{21}\bot$ *and* $b\#\overline{2} \to c\#\overline{432}$. *The system* PR_2 *contains the rule* $\hat{c}\#\overline{432} \to \hat{d}\#$. *Hence, we have the rewrite sequence*

$$a\#\bot \mapsto b\#\overline{21}\bot \mapsto c\#\overline{4321}\bot \to \hat{c}\#\overline{4321}\bot \to \hat{d}\#\overline{1}\bot$$

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

Lemma 19 *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 $\{a_i\}_{v_i} - \{b_i\}_{w_i} \in U$, $i = 1, \dots, n$, and a sequence of strings $h_i \in U^*$, $i = 1, \dots, n$, 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 $i = 1, \dots, n-1$
(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 20 *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 $a = b$, $x_1 = y_1$, $x_2 = y_2$, $x_3 = \epsilon$,
or there exists a sequence of binary terms $\{a_i\}_{v_i} - \{b_i\}_{w_i} \in U$, $i = 1, \dots, n$, and a sequence of strings $h_i \in U^*$, $i = 1, \dots, n$, 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 $i = 1, \dots, n-1$
(c) $\{b_n\}_{w_nh_n} = \{b\}_{y_1y_2x_3}$
and such that for some i the longest common suffix of x_3 and h_i is ϵ.

The proof of these two lemmas can be found in the appendix.

Hence, if t and s are both not of the form $\{m\}_k$ then there is a proof of $T \vdash \{t\}_v - \{s\}_w$ 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 21 *Let L be a set of at most binary terms, K a set 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 Lemmas 19 and 20, 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 [37]. □

As a consequence we obtain:

Theorem 22 *The binary intruder deduction problem for the equational theory of Abelian groups with distributive encryption is decidable in polynomial time.*

11 Conclusion

Public key encryption. Moving from symmetric key encryption (as used in this paper) to public key encryption simply amounts to adding a new operator I which computes the private key associated to a public key. Then, the decryption rule becomes

$$\frac{T \vdash \{u\}_v \downarrow_{R/AC} \qquad T \vdash I(v) \downarrow_{R/AC}}{T \vdash u \downarrow_{R/AC}}$$

and the definition of subterms is completed by $S(I(t)) = \{I(t)\}$, stating that the inverse operation hides its argument. The lemmas and proofs stated in the symmetric case are extended to this framework in a straightforward way.

Related works. The use of locality in the analysis of cryptographic protocols has been used first in [38], and later on by [12,13]. In [39], we studied the case of a homomorphic operator that distributes over some binary operation + 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 [17] 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 [17] 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.

Further work. A main step of our approach uses an idea which is similar to the one used in [17]: regroup certain combinations of "+-constructions", encryptions, and (in our case) decryptions into one "macro" rule, the instances of which are then decided by an ad-hoc method (here linear algebra or prefix rewriting in the binary case). This could be probably formalized into a generic solution to solve the intruder deduction problem. The active case is more problematic since undecidability results show that this case is much more complex.

Another 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 [40], 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 $I(y)$.

References

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

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

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

[4] AVISPA Project, Avispa protocol library, available at `http://www.avispa-project.org/`.

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

[6] M. Abadi, P. Rogaway, Reconciling two views of cryptography (the computational soundness of formal encryption), in: Proc. 1st IFIP International Conference on Theoretical Computer Science (IFIP–TCS), Vol. 1872 of Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 3–22.

[7] D. Dolev, A. Yao, On the security of public-key protocols, in: Transactions on Information Theory, Vol. 29, IEEE Computer Society Press, 1983, pp. 198–208.

[8] V. Cortier, S. Delaune, P. Lafourcade, A survey of algebraic properties used in cryptographic protocols, Journal of Computer Security 14 (1) (2006) 1–43.

[9] IEEE 802.11 Local and Metropolitan Area Networks: Wireless LAN Medium Acess Control (MAC) and Physical (PHY) Specifications (1999). URL `http://grouper.ieee.org/groups/802/11/main.html`

[10] L. Gong, Using one-way functions for authentication, SIGCOMM Computer Communication 19 (5) (1989) 8–11.

[11] J. Bull, D. J. Otway, The authentication protocol, Tech. Rep. DRA/CIS3/PROJ/CORBA/SC/1/CSM/436-04/03, Defence Research Agency (1997).

[12] H. Comon-Lundh, 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), IEEE Comp. Soc. Press, Ottawa (Canada), 2003, pp. 271–280.

[13] Y. Chevalier, R. Küsters, M. Rusinowitch, 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), IEEE Comp. Soc. Press, Ottawa (Canada), 2003, pp. 261–270.

[14] J. Millen, V. Shmatikov, Symbolic protocol analysis with products and Diffie-Hellman exponentiation, in: Proc. 16th Computer Security Foundation Workshop (CSFW'03), IEEE Comp. Soc. Press, Pacific Grove (California, USA), 2003, pp. 47–62.

[15] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani, Deciding the security of protocols with Diffie-Hellman exponentiation and products in exponents., in: P. K. Pandya, J. Radhakrishnan (Eds.), FSTTCS, Vol. 2914 of Lecture Notes in Computer Science, Springer, 2003, pp. 124–135.

[16] H. Comon-Lundh, R. Treinen, Easy intruder deductions, in: N. Dershowitz (Ed.), Verification: Theory & Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday, Vol. 2772 of Lecture Notes in Computer Science, Springer-Verlag, 2003, pp. 225–242.

[17] S. Delaune, Easy intruder deduction problems with homomorphisms, Information Processing Letters 97 (6) (2006) 213–218.

[18] S. Delaune, P. Lafourcade, D. Lugiez, R. Treinen, Symbolic protocol analysis in presence of a homomorphism operator and *exclusive or*, in: Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06), Vol. 4052 of Lecture Notes in Computer Science, Springer, Venice, Italy, 2006, pp. 132–141.

[19] S. Delaune, An undecidability result for AGh, Theoretical Computer Science.To appear.

[20] M. Baudet, Deciding security of protocols against off-line guessing attacks, in: Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS'05), ACM Press, Alexandria, Virginia, USA, 2005, pp. 16–25.

[21] S. Delaune, F. Jacquemard, A decision procedure for the verification of security protocols with explicit destructors, in: V. Atluri, B. Pfitzmann, P. McDaniel (Eds.), Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS'04), ACM Press, Washington, D.C., USA, 2004, pp. 278–287.

[22] M. Abadi, V. Cortier, Deciding knowledge in security protocols under equational theories, in: Proc. 31st International Colloquium on Automata, Languages, and Programming (ICALP'04), Vol. 3142 of Lecture Notes in Computer Science, Springer-Verlag, Turku (Finland), 2004, pp. 46–58.

[23] M. Abadi, V. Cortier, Deciding knowledge in security protocols under (many more) equational theories, in: CSFW '05: Proceedings of the 18th IEEE Computer Security Foundations Workshop (CSFW'05), IEEE Computer Society, Washington, DC, USA, 2005, pp. 62–76.

[24] Y. Chevalier, M. Rusinowitch, Combining intruder theories., in: L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, M. Yung (Eds.), ICALP, Vol. 3580 of Lecture Notes in Computer Science, Springer, 2005, pp. 639–651.

[25] H. Comon-Lundh, Intruder theories (ongoing work), in: Proc. 7th International Conference on Foundations of Software Science and Computation Structures (FOSSACS'04), Vol. 2987 of LNCS, Springer-Verlag, Barcelona (Spain), 2004, pp. 1–4.

[26] N. Durgin, P. Lincoln, J. Mitchell, A. Scedrov, Undecidability of bounded security protocols, in: Proc. Workshop on Formal Methods and Security Protocols (FMSP'99), Trento (Italy), 1999.

[27] J. Millen, V. Shmatikov, Constraint solving for bounded-process cryptographic protocol analysis, in: Proc. 8th ACM Conference on Computer and Communications Security (CCS'01), ACM Press, 2001, pp. 166–175.

[28] M. Tatebayashi, N. Matsuzaki, D. B. Newman, Key distribution protocol for digital mobile communication systems, in: Proc. 9th Annual International Cryptology Conference (CRYPTO'89), Vol. 435 of LNCS, Springer-Verlag, Santa Barbara (California, USA), 1989, pp. 324–333.

[29] D. Prawitz, Natural Deduction, A Proof-Theoretical Study, Almquist and Wiksell, 1965.

[30] D. B. and Sebastian Mödersheim and Luca Viganò, Ofmc: A symbolic model checker for security protocols, International Journal of Information Security 4 (3) (2005) 181–208, published online December 2004.

[31] M. Turuani, The CL-Atse protocol analyser, in: F. Pfenning (Ed.), 17th International Conference on Term Rewriting and Applications, Vol. 4098 of Lecture Notes in Computer Science, Springer, Seattle, WA, USA, 2006, pp. 277–286.

[32] N. Dershowitz, J.-P. Jouannaud, Rewrite systems, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, Vol. B - Formal Models and Semantics, Elsevier Science Publishers and The MIT Press, 1990, Ch. 6, pp. 243–320.

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

[34] P. Lafourcade, D. Lugiez, R. Treinen, Intruder deduction for AC-like equational theories with homomorphisms, Research Report LSV-04-16, LSV, ENS de Cachan (Nov. 2004).

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

[36] L. Dickson, Finiteness of the odd perfect and primitive abundant numbers with n prime factors, American Journal Mathematical Society 35 (1913) 413–422.

[37] D. Caucal, On the regular structure of prefix rewriting., Theoretical Comput. Sci. 106 (1) (1992) 61–86.

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

[39] P. Lafourcade, D. Lugiez, R. Treinen, Intruder deduction for AC-like equational theories with homomorphisms, in: J. Giesl (Ed.), Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05), Vol. 3467 of Lecture Notes in Computer Science, Springer-Verlag, Nara, Japan, 2005, pp. 308–322.

[40] P. Lafourcade, Intruder deduction for the equational theory of *exclusive-or* with commutative and distributive encryption, in: Proceedings of the 1st International Workshop on Security and Rewriting Techniques (SecReT'06), Venice, Italy, 2006.

A Proofs for Section 10

Definition 16 *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 9 *The prefix rewrite system of Section 10 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 10 *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 lemmas 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 $\{a_i\}_{v_i} - \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, and a sequence of strings $h_i \in U^*$, $i = 1, \ldots, n$, such that
(a) $\{a\}_{x_1 x_2 y_3} = \{a_1\}_{v_1 h_1}$
(b) $\{b_i\}_{w_i h_i} = \{a_{i+1}\}_{v_{i+1} h_{i+1}}$ for $i = 1, \ldots, n-1$
(c) $\{b_n\}_{w_n h_n} = \{b\}_{y_1 y_2}$
and such that for some i the longest common suffix of y_3 and h_i is ϵ.

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 $a = b$, $x_1 = y_1$, $x_2 = y_2$, $x_3 = \epsilon$,
or there exists a sequence of binary terms $\{a_i\}_{v_i} - \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, and a sequence of strings $h_i \in U^*$, $i = 1, \ldots, n$, such that
(a) $\{a\}_{x_1 x_2} = \{a_1\}_{v_1 h_1}$
(b) $\{b_i\}_{w_i h_i} = \{a_{i+1}\}_{v_{i+1} h_{i+1}}$ for $i = 1, \ldots, n-1$
(c) $\{b_n\}_{w_n h_n} = \{b\}_{y_1 y_2 x_3}$
and such that for some i the longest common suffix of x_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

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

since then

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

(2) The rewrite rule is of the form

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

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

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

since then

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

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 9 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

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

By induction hypothesis, there exists a sequence of binary terms $\{a_i\}_{v_i} - \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, 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) $\{b_i\}_{w_ih_i} = \{a_{i+1}\}_{v_{i+1}h_{i+1}}$ for $i = 1, \ldots, n-1$
(3) $\{b_n\}_{w_nh_n} = \{b\}_{y_1y_2}$

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

(1) $\{a\}_{x_1x_2z_3} = \{a_1\}_{v_1k_1}$
(2) $\{b_i\}_{w_ik_i} = \{a_{i+1}\}_{v_{i+1}k_{i+1}}$ for $i = 1, \ldots, n$
(3) $\{b_{n+1}\}_{w_{n+1}k_{n+1}} = \{c\}_{z_1z_2}$

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

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

Then there exists u such that

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

We conclude by choosing

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

since

$$\begin{aligned} &\{a\}_{x_1x_2z_3} = \{a\}_{x_1x_2y_3} = \{a_1\}_{v_1h_1} = \{a_1\}_{v_1k_1} \\ &\{b_i\}_{w_ik_i} = \{b_i\}_{w_ih_i} = \{a_{i+1}\}_{v_{i+1}h_{i+1}} = \{a_{i+1}\}_{v_{i+1}k_{i+1}} \quad (i = 1, \ldots, n-1) \\ &\{b_n\}_{w_nk_n} = \{b\}_{y_1y_2} = \{b\}_{ru} = \{a_{n+1}\}_{v_{n+1}k_{n+1}} \\ &\{b_{n+1}\}_{w_{n+1}k_{n+1}} = \{c\}_{su} = \{c\}_{z_1z_2} \end{aligned}$$

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

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

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

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

We conclude by choosing

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

since

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

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 $ax_1 \# x_2 \perp \mapsto^* by_1 \# y_2 \overline{y_3} \perp$. Otherwise, we proceed by induction on n.

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

If $n \geq 2$ then there exists a sequence of binary terms $\{a_i\}_{v_i} - \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, 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 $\{b\}_{y_1y_2} = \{b_n\}_{w_nh_n}$ Hence,

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

(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 $h_i' = h_is$. Hence, we also have 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 $i = 1, \ldots, n-2$
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 $\{b_{n-1}\}_{w_{n-1}h_{n-1}} = \{a_n\}_{v_nh_n}$, that is $w_{n-1}h_{n-1}'s = v_nh_n$. 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 $v_n = v_n^1s$ and obtain that

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

□