Using ProVerif to Analyze Protocols with Diffie-Hellman Exponentiation

Ralf Küsters and Tomasz Truderung
University of Trier, Germany
{*kuesters,truderung*}*@uni-trier.de*

***Abstract*—ProVerif is one of the most successful tools for cryptographic protocol analysis. However, dealing with algebraic properties of operators such as the exclusive OR (XOR) and Diffie-Hellman exponentiation has been problematic. Recently, we have developed an approach which enables ProVerif, and related tools, to analyze a large class of protocols that employ the XOR operator. In this work, we adapt this approach to the case of Diffie-Hellman exponentiation.**

The core of our approach is to reduce the derivation problem for Horn theories modulo algebraic properties of Diffie-Hellman exponentiation to a purely syntactical derivation problem for Horn theories. The latter problem can then be solved by tools such as ProVerif. Our reduction works for a large class of Horn theories, allowing to model a wide range of intruder capabilities and protocols. We implemented our reduction and, in combination with ProVerif, applied it in the automatic analysis of several state-of-the-art protocols that use Diffie-Hellman exponentiation.

While the general idea of our approach follows the one for XOR in our previous work, the reduction itself and the proof of soundness and completeness of our reduction are entirely different from the XOR case. Surprisingly, the reduction for Diffie-Hellman exponentiation is more efficient than the one for XOR.

I. Introduction

ProVerif [5] is one of the most successful tools for the analysis of cryptographic protocols. It relies on the Horn theory based approach, in which protocols and intruders are modeled as Horn theories. Protocol analysis is w.r.t. an unbounded number of protocol sessions that may run concurrently and without putting a bound on the size of messages an intruder can generate. Verifying security properties, such as secrecy, boils down to solving the derivation problem for Horn theories. However, dealing with algebraic properties of operators such as the exclusive OR (XOR) and Diffie-Hellman exponentiation (DH) has been problematic (see the related work).

Recently, we have proposed a method for dealing with XOR in the Horn theory based approach [21]. The idea was to reduce the derivation problem modulo XOR to a purely syntactical derivation problem. The latter problem could then be solved by tools such as ProVerif, which otherwise cannot deal with XOR. The reduction works for a large class of Horn theories with XOR, allowing to model a wide range of protocols and intruder capabilities.

The goal of this work is to adapt this approach to DH, and by this, obtain a practical method for the automatic analysis of protocols that use DH, where the analysis is w.r.t. an unbounded number of sessions, without putting a bound on the message size, and taking a relatively rich set of algebraic properties for DH into account, including commutativity of exponents and inverses. More precisely, the contribution of our work is as follows.

Contribution of this Work. We introduce an expressive class of (unary) Horn theories, called exponent-ground Horn theories. A Horn theory is exponent-ground, if for every Horn clause in this theory the terms occurring in the clause are exponent-ground. A term t is exponent-ground if, roughly speaking, all subterms occurring in exponents are ground, i.e., do not contain variables. However, we allow for (non-exponent-ground) clauses which enable the intruder to perform exponentiation and compute inverses for arbitrary messages.

Our approach will allow us to deal with all cryptographic protocols and intruder capabilities that can be modeled as exponent-ground Horn theories. Note that clauses which do not contain the exponentiation or inverse symbol are exponent-ground by definition. The algebraic properties that we consider for DH are more accurate than those in other works for cryptographic protocol analysis w.r.t. an unbounded number of sessions. In these works, inverses are not considered (see the related work). We do not explicitly consider a product operator in exponents, unlike works on protocol analysis w.r.t. a *bounded* number of sessions. However, we show that our way of modeling DH corresponds to the one where products may only occur in exponents (see, e.g., [10]). This relationship appears to be of independent interest.

Our main technical result is that the derivation problem for exponent-ground Horn theories can be reduced to a purely syntactical derivation problem, i.e., a derivation problem where the algebraic properties of DH do not have to be considered anymore. Now, the syntactical derivation problem can be solved by highly efficient tools, such as ProVerif, for which dealing with DH is otherwise problematic or impossible. Surprisingly, unlike the case of XOR, our reduction is efficient.

This work was partially supported by the *Deutsche Forschungsgemeinschaft* (DFG) under Grant KU 1434/5-1 and 1434/4-2, the SNF under Grant 200021-116596, and the Polish Ministry of Science and Education under Grant 3 T11C 042 30. The second author is on leave from University of Wrocław, Poland.

We implemented our reduction, and using ProVerif, applied our two step approach—first reduce the problem, then run ProVerif on the result of the reduction—to the analysis of several state-of-the-art cryptographic protocols that employ DH. The experimental results demonstrate that our approach is practical and rather robust. The implementation is available at [20].

Just as in case of XOR, we note that a potential alternative to our approach is to perform unification modulo DH instead of syntactical unification in a resolution algorithm for solving the derivation problem. Whether or not this approach is practical is an open problem. The main difficulty is that unification modulo DH is much more inefficient than syntactical unification; it is NP-complete rather than linear and, in general, there does not exist a (single) most general unifier.

Related Work. As mentioned, in previous work [21], we have successfully applied the approach of reducing the derivation problem modulo an equational theory to the syntactical derivation problem in case of XOR. However, the reduction for DH and also the proof of soundness and completeness of this reduction are entirely different from the one for XOR. While the reduction for XOR suffers from an exponential blow up, the one for DH is efficient.

ProVerif has been used before to analyze protocols that employ DH (see, e.g., [7], [6], [1]). However, the only algebraic property of DH considered was commutativity of exponents, without taking inverses into account. Also, a fixed basis for exponentiation was assumed. Hence, attacks which exploit a richer set of algebraic properties, such as the one considered in our setting, are not captured. Also, protocols that explicitly use the inverse operator cannot be modeled in the works by Blanchet et al.

Meadows et al. [23], [22], [14] used their tool, the (Maude-)NRL Analyzer, for the analysis of protocols with DH. They too only considered the commutativity property of exponentiation, leaving out inverses.

In [15], Goubault-Larrecq et al. study decidability of a certain class of Horn theories with DH. However, inverses are not considered. They present an automatic analysis of one protocol w.r.t. a *bounded* number of sessions. Decidability results for classes of Horn theories modulo Abelian Groups were obtained by Verma et al. (see, e.g., [31]).

Automatic analysis w.r.t. a *bounded* number of sessions for protocols with DH is studied in [10], [9], [25], [29], [11]. An implementation based on [10] is presented in [30].

Unification modulo equational theories related to DH is examined, for example, in [24], [17].

Manual analysis based on the Protocol Composition Logic for protocols that use DH was carried out by Roy et al. (see, e.g., [28]).

Structure of this Work. In the following section, we recall the Horn Theory based approach. Exponent-ground Horn theories are introduced in Section III. In that section, we also establish an important property of these theories. The reduction, along with a proof of soundness and completeness, is presented in Section IV. Our implementation [20] and the experimental results are discussed in Section V. We conclude in Section VI, with further details provided in the appendix.

II. The Horn Theory Based Approach

In this section, we introduce Horn theories modulo the Diffie-Hellman exponentiation operator and illustrate, by means of a running example, how these theories can be used to model cryptographic protocols. We also relate our algebraic theory for Diffie-Hellman exponentiation to those previously proposed in the literature.

A. Horn Theories

Let Σ be a finite signature and V be a set of variables. The set of terms over Σ and V is defined as usual. By $\mathrm{var}(t)$ we denote the set of variables that occur in the term t.

To model cryptographic protocols, Σ typically contains constants (*atomic messages*, such as principal names, nonces, keys, or pre-agreed group generators for Diffie-Hellman exponentiation), unary function symbols, such as $\mathsf{hash}(\cdot)$ (*hashing*) and $\mathsf{pub}(\cdot)$ (*public key*), and binary function symbols, such as $\langle\cdot,\cdot\rangle$ (*pairing*), $\mathsf{mac}.(\cdot)$ (MAC), $\{\cdot\}.$ (*symmetric encryption*), $\{\!|\cdot|\!\}.$ (*public key encryption*), and $\mathsf{sig}.(\cdot)$ (*digital signature*). The signature Σ may also contain any other free function symbols.

We assume Σ to contain the binary function symbol $\uparrow$ (*Diffie-Hellman exponentiation*) and the unary symbol $\cdot^{-1}$ (*inverse*). Instead of $\uparrow(t,s)$, we use the infix notation $t\uparrow s$, which intuitively means that t is taken to the s-th power. We often write $t_0\uparrow t_1\uparrow\cdots\uparrow t_n$ instead of $(\cdots(t_0\uparrow t_1)\uparrow\cdots)\uparrow t_n$, i.e., our convention is that $\uparrow$ is left-associative.

For a given term, we define the set of its *subterms* in the usual way. For example, $x\uparrow b$ is a subterm of $t=(x\uparrow b)\uparrow c$, but $x\uparrow c$ is not a subterm of t.

Ground terms, i.e. terms without variables, are called *messages*. A *substitution* is a finite set of pairs of the form $\sigma=\{t_1/x_1,\ldots,t_n/x_n\}$, where $t_1,\ldots,t_n$ are terms and $x_1,\ldots,x_n$ are variables. The set $\mathrm{dom}(\sigma)=\{x_1,\ldots,x_n\}$ is called the domain of σ. The application $s\sigma$ of σ to a term/atom/set of terms s is defined as usual.

We model algebraic properties for Diffie-Hellman exponentiation (DH) by the congruence relation $\sim$ on terms induced by the following equational theory:

$$(x\uparrow y)\uparrow z = (x\uparrow z)\uparrow y \tag{DH1}$$

$$(x\uparrow y)\uparrow y^{-1} = x \tag{DH2}$$

$$(x^{-1})^{-1} = x \tag{DH3}$$

For example, let $t_1^{ex} = a\uparrow b\uparrow c^{-1}\uparrow b^{-1}\uparrow(d^{-1})^{-1}$, $t_2^{ex} = a\uparrow c^{-1}\uparrow d$, and $t_3^{ex} = a\uparrow d\uparrow c^{-1}$. Then, we have that $t_1^{ex}\sim t_2^{ex}\sim t_3^{ex}$.

We say that a term is *reduced* if, modulo (DH1), the equations (DH2) and (DH3), when interpreted as reductions from left to right, cannot be applied. Clearly, every term can be turned into a reduced form and this form is uniquely determined modulo (DH1). Moreover, we have that $t \sim s$ if and only if the reduced forms of t and s are equal modulo (DH1). We write $t \doteq s$, if t and s are equal modulo (DH1). For example, we obtain t_2^{ex} and t_3^{ex} by reducing t_1^{ex}. It holds that $t_2^{ex} \doteq t_3^{ex}$.

A term is *standard*, if its head symbol is neither $\uparrow$ nor $\cdot^{-1}$; otherwise it is *non-standard*. A term is *pure*, if $\uparrow$ and $\cdot^{-1}$ do not occur in it.

We write $t \uparrow s^{(n)}$, for $n \geq 0$, as an abbreviation for $t \uparrow s \uparrow \cdots \uparrow s$, where s is repeated n times. Similarly, $t \uparrow s^{(-n)}$ stands for $t \uparrow s^{-1} \uparrow \cdots \uparrow s^{-1}$, where s^{-1} is repeated n times. We stress that the expressions $t \uparrow s^{(n)}$ and $t \uparrow s^{(-n)}$ are merely abbreviations; our formal syntax does not contain integers.

For a unary predicate q and a (ground) term t, we call $q(t)$ a *(ground) atom*. A *Horn theory* T is a finite set of *Horn clauses*, each of the form $a_1, \ldots, a_n \to a_0$, where $a_0, \ldots, a_n$ are atoms. If $n = 0$, i.e., the left-hand side of the clause is always true, we call the Horn clause a_0 a *fact*. We write $q(t) \sim q'(t')$ if $q = q'$ and $t \sim t'$.

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

Similarly, we write $T \vdash_{DH} a$ if there exists a *derivation of a from T modulo DH*, i.e., there exists a sequence $b_1, \ldots, b_l$ of ground atoms such that $b_l \sim a$ and for every $i \in \{1, \ldots, l\}$ there exists a substitution σ and a Horn clause $a_1, \ldots, a_n \to a_0$ in T such that $a_0\sigma \sim b_i$ and for every $j \in \{1, \ldots, n\}$ there exists $k \in \{1, \ldots, i-1\}$ with $a_j\sigma \sim b_k$. We define $\pi(i)$, $\pi_{<i}$, and $|\pi|$ as above. Also, as above, we say that π is a *derivation (modulo DH) for* $T \vdash_{DH} a$.

Given T and a, we call the problem of deciding whether $T \vdash a$ (or $T \vdash_{DH} a$) is true, the *deduction problem (modulo DH)*.

B. Modeling Protocols by Horn theories

Following [5], we now illustrate how Horn theories can be used to analyze cryptographic protocols (with DH). The Horn theory based approach allows to analyze protocols, w.r.t. an unbounded number of sessions running concurrently and without putting a bound on the message size, in a fully automatic and sound way. However, the method may produce false attacks and analysis tools may not terminate.

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

Figure 1. Intruder rules for standard cryptographic primitives.

A Horn theory for modeling protocols and the (Dolev-Yao) intruder typically uses only the predicate I. The fact $\mathrm{I}(t)$ means that the intruder may be able to obtain the term t. The fundamental property is that if $\mathrm{I}(t)$ cannot be derived from the set of clauses, then the protocol preserves the secrecy of t. The Horn theory consists of three sets of Horn clauses: the initial intruder facts, the intruder rules, and the protocol rules. The set of *initial intruder facts* represents the initial intruder knowledge and may contain, for instance, names of principals, public keys, and a pre-agreed group generator for DH. The clauses in this set are facts, e.g., $\mathrm{I}(a)$ (the intruder knows the name a) and $\mathrm{I}(\mathsf{pub}(sk_a))$ (the intruder knows the public key of a, with sk_a being the corresponding private key). The set of *intruder rules* represents the intruders ability to derive new messages. For the cryptographic primitives mentioned in Section II-A, the set of intruder rules contains the clauses depicted in Figure 1. Additionally, to model the intruder's ability to perform Diffie-Hellman exponentiation and to compute the inverse operation on arbitrary messages, the set of intruder rules also contains the two rules given in Figure 2. The theory containing these two rules is called T_{DH}.

We stress that the rules presented in Figure 1 are only examples; this theory can be extended to capture other cryptographic primitives. As long as these rules are exponent-ground, which, for example, is the case if the operators $\uparrow$ and $\cdot^{-1}$ do not occur in these rules (see the next section for the definition of exponent-ground), our results apply.

The set of *protocol rules* represents the actions performed in the actual protocol. The i-th protocol step of a principal is described by a clause of the form $\mathrm{I}(r_1), \ldots, \mathrm{I}(r_i) \to \mathrm{I}(s_i)$ where the terms r_j, for $j \in \{1, \ldots, i\}$, describe the (patterns of) messages the principal has received in the previous $(i-1)$ steps plus the (pattern of the) message in the i-th step. The

$$\mathrm{I}(x), \mathrm{I}(y) \to \mathrm{I}(x \uparrow y) \tag{1}$$

$$\mathrm{I}(x) \to \mathrm{I}(x^{-1}) \tag{2}$$

Figure 2. Theory T_{DH} — intruder rules for exponentiation and inverse.

term $\mathrm{I}(s_i)$ is the (pattern of) the i-th output message of the principal.

Given a protocol P, we denote by T_P the Horn theory that comprises all three sets of clauses, mentioned above, *except* for the clauses of T_{DH}. We will call T_P *the theory of* P.

Now, given a protocol P and a message m, the fact that $T_P \cup T_{DH} \vdash_{DH} \mathrm{I}(m)$ does *not* hold means that an intruder cannot get hold of m even when interacting with an unbounded number of sessions of the protocol and employing algebraic properties of DH. In other words, $T_P \cup T_{DH} \nvdash_{DH} \mathrm{I}(m)$ means that P guarantees the secrecy of the message m (in the symbolic model considered).

C. Running Example

We use the SIGMA-BASIC protocol proposed by Krawczyk [19] as our running example. SIGMA is a family of key exchange protocols that serve as the basis for the signature-based modes of the IKE protocol (version 1 and 2). The intended run of SIGMA-BASIC is as follows:

$$\begin{array}{lll} \text{(P1)} & A \to B: & g\uparrow N \\ \text{(P2)} & B \to A: & g\uparrow M,\ B,\ \mathsf{sig}_{k_B}(\langle g\uparrow N, g\uparrow M\rangle),\ \mathsf{mac}_K(B) \\ \text{(P3)} & A \to B: & A,\ \mathsf{sig}_{k_A}(\langle g\uparrow M, g\uparrow N\rangle),\ \mathsf{mac}_K(A) \end{array}$$

where A and B are agent names, g is a group generator, N and M are nonces generated by A and B, respectively, k_A and k_B are private keys of A and B, respectively, and $K = \mathsf{hash}(g\uparrow N\uparrow M)$ is a key derived from N and M. One of the main security properties this protocol is supposed to achieve is that the value $g\uparrow N\uparrow M$ is secret, i.e., if A and B are honest participants, the intruder should not be able to derive $g\uparrow N\uparrow M$, as this value is used to derive the session key shared between A and B.

To illustrate how this protocol can be modeled in terms of Horn theories, let P be a set of participant names and $\mathsf{H} \subseteq \mathsf{P}$ be a set of names of honest participants. Following [12], it is easy to see that as far as secrecy properties are concerned, it suffices to consider the case $\mathsf{P} = \{a, b\}$ and $\mathsf{H} = \{a\}$. (If no attack is found with these sets of participants, then there is no attack even if bigger sets of (honest and dishonest) participants are taken into account.) In the following, k_a, for $a \in \mathsf{P}$, denotes the private key of a, n_{ab} denotes the nonce generated by $a \in \mathsf{P}$ and sent to $b \in \mathsf{P}$ in message (P1), and m_{ba} denotes the nonce generated by b and sent to a in message (P2).

The initial intruder knowledge is given by the following set of facts.

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

The intruder rules are those depicted in Figure 1 (the intruder rules in Figure 2 are taken into account separately; see below). The first step of the protocol performed by an honest principal is modeled by the facts

$$\mathrm{I}(g\uparrow n_{ab}) \tag{3}$$

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

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

$$\mathrm{I}(x) \to \mathrm{I}(\langle g\uparrow m_{ba},\ b,\ \mathsf{sig}_{k_b}(\langle x, g\uparrow m_{ba}\rangle)),\ \mathsf{mac}_{\mathsf{hash}(x\uparrow m_{ba})}(b)\rangle) \tag{4}$$

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

$$\mathrm{I}(\langle y,\ b,\ \mathsf{sig}_{k_b}(\langle g\uparrow n_{ab}, y\rangle),\ \mathsf{mac}_{\mathsf{hash}(y\uparrow n_{ab})}(b)\rangle) \to \mathrm{I}(\langle a,\ \mathsf{sig}_{k_a}(\langle y, g\uparrow n_{ab}\rangle),\ \mathsf{mac}_{\mathsf{hash}(y\uparrow n_{ab})}(a)\rangle) \tag{5}$$

for all $a \in \mathsf{H}$ and $b \in \mathsf{P}$. The set of Horn clauses defined above is denoted by T_{SB}.

The protocol is supposed to guarantee secrecy of $g\uparrow n_{ab}\uparrow m_{ba}$, for all $a, b \in \mathsf{H}$. Formally, this means that $T_{SB} \cup T_{DH} \vdash_{DH} \mathrm{I}(g\uparrow n_{ab}\uparrow m_{ba})$ should *not* hold for any $a, b \in \mathsf{H}$. We note that even though in this running example, the inverse operator is not explicitly used in the specification of the protocol, the intruder still can make use of this operator and its algebraic properties as introduced in Section II-A, potentially leading to successful attacks which otherwise would have been overlooked.

D. Remark on Algebraic Properties of DH

In other works on the automatic analysis of cryptographic protocols with DH (see, e.g., [8], [10], [29], [25]), different representations of terms were chosen and in some cases (apparently) more complex algebraic properties for DH were considered. Even among these existing works there are differences and the settings cannot be compared directly. In section we show that there is a strong relationship between our way of modeling DH and those models proposed for protocol analysis w.r.t. a bounded number of sessions where products of terms are considered and these products are restricted to appear in exponents only. For the sake of concreteness, we compare our work with the one by Chevalier et al. [10]. Nevertheless, our comparison also sheds some light on the relationship of our work with the other mentioned works.

In [10], terms representing exponentiation are of the form $\exp(t_0, t_1^{z_1} \times \cdots \times t_n^{z_n})$, where $t_0, \ldots, t_n$ are terms, $z_1, \ldots, z_n$ are integers, and $\times$ is a product operator. The equational theory

for such terms proposed in [10] is the following one:

$$\begin{array}{cc} (t_1 \times t_2) = (t_2 \times t_1) & (t_1 \times t_2) \times t_3 = t_1 \times (t_2 \times t_3) \\ t^z \times t^{z'} = t^{z+z'} & t \times 1 = t \\ t^1 = t & \exp(t,1) = t \\ t^0 = 1 & \exp(\exp(t,t'),t'') = \exp(t, t' \times t'') \\ 1^z = 1 & \end{array}$$

where z and z' are integers and '$+$' is addition of integers. We denote the congruence relation on terms induced by this apparently more complex equational theory by $\equiv$.

The intruder rule for DH considered in [10] is

$$\mathrm{I}(x_0), \ldots, \mathrm{I}(x_n) \;\rightarrow\; \mathrm{I}(\exp(x_0, x_1^{z_1} \times \cdots \times x_n^{z_n})) \qquad (6)$$

for variables $x_0, \ldots, x_n$ and integers $z_1, \ldots, z_n$.

To compare this way of modeling DH with our modeling of DH, in what follows we assume that our signature Σ contains the (free) constant 1. We call our terms *KT-terms* and those in [10] *CKRT-terms*.

We turn a CKRT-term t into a KT-term $\mathrm{KT}(t)$ as follows: First t is normalized as defined in [10], i.e., the equations given above are applied exhaustively from left to right modulo commutativity and associativity of '$\times$'. Let t' denote the resulting term, which is uniquely determined modulo commutativity and associativity of '$\times$'. Now, $\mathrm{KT}(t)$ is obtained from t' by replacing every term of the form $\exp(t_0, t_1^{z_1} \times \cdots \times t_n^{z_n})$ by $t_0 \uparrow t_1^{(z_1)} \uparrow \cdots \uparrow t_n^{(z_n)}$. (Recall that $t \uparrow s^{(n)}$ is an abbreviation defined in Section II-A).

It is easy to see that (the KT-term) $\mathrm{KT}(t)$ is a CKRT-term if all occurrences of $\cdot\uparrow\cdot$ are replaced by $\exp(\cdot,\cdot)$. (Note that $\mathrm{KT}(t)$ does not contain subterms of the form $(s^{-1})^{-1}$ and subterms of the form s^{-1} only occur in a context of the form $s' \uparrow s^{-1}$.) Hence, by abuse of notation, we can interpret $\mathrm{KT}(t)$ as a CKRT-term. Now, clearly we have:

Lemma 1. *Let t be a CKRT-term. Then, $\mathrm{KT}(t) \equiv t$.*

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

Lemma 2. *Let t and t' be CKRT-terms. Then, $t \equiv t'$ if and only if $\mathrm{KT}(t) \sim \mathrm{KT}(t')$.*

Let T be a Horn theory over CKRT-terms. We write $T \vdash_{\equiv} a$, if there exists a derivation of a from T modulo $\equiv$. This is defined analogously to $T \vdash_{DH} a$. Given a Horn theory T over CKRT-terms, we translate it into a Horn theory $\mathrm{KT}(T)$ over KT-terms as follows: (a) Horn clauses of the form (6) are removed, (b) for each of the remaining Horn clauses of the form $s_1, \ldots, s_n \rightarrow s_0$ in T, the theory $\mathrm{KT}(T)$ contains $\mathrm{KT}(s_1), \ldots, \mathrm{KT}(s_n) \rightarrow \mathrm{KT}(s_0)$, where for an atom $p(t)$, we define $\mathrm{KT}(p(t))$ to be the atom $p(\mathrm{KT}(t))$, and (c) the clauses in T_{DH} (see Figure 2) are added to $\mathrm{KT}(T)$.

The following theorem shows that our modeling of DH is at least as accurate as the one in [10]. The proof of this theorem is given in the appendix.

Theorem 1. *Let T be a Horn theory of CKRT-terms and let a be a CKRT-atom. Then, $T \vdash_{\equiv} a$ implies $\mathrm{KT}(T) \vdash_{DH} \mathrm{KT}(a)$.*

III. Exponent-ground Theories

In Section IV, we show how to reduce the deduction problem modulo DH to the one without DH, the latter problem can then be solved using tools such as ProVerif. The reduction works for a large class of Horn theories, namely exponent-ground theories. In this section, exponent-ground theories are introduced and a theorem that is the key to the reduction presented in Section IV is shown.

A term t is *well-formed* if every suberm of t of the form s^{-1} only occurs in a context of the form $s' \uparrow s^{-1}$ for some s'. For example, if a, b, c are constants, then a^{-1} and $a^{-1} \uparrow b$ are not well-formed, but $\langle\langle c \uparrow b^{-1}, c \uparrow b\rangle, c \uparrow a \uparrow b^{-1}\rangle$ is.

A term is *exponent-ground* if it is well-formed and for each of its subterms of the form $t \uparrow s$ it is true that s is of the form c or c^{-1}, where c is a pure, ground term. A Horn clause $p_1(t_1), \ldots, p_n(t_n) \rightarrow p_0(t_0)$ is called exponent-ground if the terms $t_0, \ldots, t_n$ are. A Horn theory is exponent-ground if each clause in this theory is exponent-ground. Also, a derivation $p_1(t_1), \ldots, p_n(t_n)$ is exponent-ground if each term $t_1, \ldots, t_n$ is.

We also need a more fine-grained notion: C-exponent-ground. Let C be a finite set of pure, ground terms. By C^{-1} we denote the set $\{c^{-1} : c \in \mathsf{C}\}$ and by C^* we denote the set $\mathsf{C} \cup \mathsf{C}^{-1}$. A term is C-*exponent-ground* if it is well-formed and for each of its subterms of the form $t \uparrow s$ we have that $s \in \mathsf{C}^*$. This notion is extended to Horn clauses, Horn theories, and derivations in the obvious way. Obviously, we have:

Lemma 3. *If a term, a Horn clause, a Horn theory or a derivation S is exponent-ground, then S is C-exponent-ground for some finite set C of pure, ground terms.*

Example 1. Let a, b, and c be constants and x, y, and z be variables. Then the term $a \uparrow x$ is not exponent-ground, while the term $\langle x \uparrow \mathsf{hash}(a), y \uparrow b^{-1} \uparrow c\rangle$ is exponent-ground. The latter term is C-exponent-ground for the set $\mathsf{C} = \{\mathsf{hash}(a), b, c\}$.

Example 2. T_{SB}, the Horn theory defined in Section II-C for modeling the SIGMA-BASIC protocol, is exponent-ground. This theory is C-exponent-ground for the set

$$\mathsf{C} = \{n_{ab} : a \in \mathsf{H}, b \in \mathsf{P}\} \cup \{m_{ab} : a \in \mathsf{H}, b \in \mathsf{P}\}.$$

Example 3. The Horn theory T_{DH} (see Figure 2) is not exponent-ground. In fact, clause (1) is not exponent-ground, because it contains a variable y as an exponent. Clause (2) is not well-formed, and by this, is also not exponent-ground. (In our approach, these clauses are dealt with separately.)

Now, we state and prove the main result of this section. It says that when a C-exponent-ground atom a can be derived modulo DH from a C-exponent-ground theory T and using the theory T_{DH} (formally $T \cup T_{DH} \vdash_{DH} a$), then there exists a

$$\mathrm{I}(x), \mathrm{I}(c) \to \mathrm{I}(x \uparrow c), \qquad \text{for each } c \in \mathsf{C} \tag{7}$$
$$\mathrm{I}(x), \mathrm{I}(c) \to \mathrm{I}(x \uparrow c^{-1}) \qquad \text{for each } c \in \mathsf{C} \tag{8}$$

Figure 3. Theory T_{DH}^{C} — a variant of intruder rules for exponentiation.

C-exponent-ground derivation of a. However, to obtain such a C-exponent-ground derivation we need to replace T_{DH} by T_{DH}^{C}, where T_{DH}^{C} is defined in Figure 3.

Theorem 2. *Let T be a C-exponent-ground Horn theory and a be a C-exponent-ground atom. If $T \cup T_{DH} \vdash_{DH} a$, then there exists a C-exponent-ground derivation for $T \cup T_{DH}^{\mathsf{C}} \vdash_{DH} a$. Moreover, the substitutions applied in this derivation are C-exponent-ground too.*

The rest of this section is devoted to the proof of Theorem 2. The idea of the proof is the following. We define a function δ_{C} which turns terms into C-exponent-ground terms. Then, we show that by applying δ_{C} to a (not necessarily C-exponent-ground) derivation for $T \cup T_{DH} \vdash_{DH} a$, we obtain a C-exponent-ground derivation for $T \cup T_{DH}^{\mathsf{C}} \vdash_{DH} a$.

In what follows, we first define the function δ_{C}. Then, before proving the theorem, several lemmas are stated and proved.

The function δ_{C} from terms to terms is simple. It is defined inductively as follows:

$$\begin{aligned}
\delta_{\mathsf{C}}(x) &= x && \text{for a variable } x \\
\delta_{\mathsf{C}}(t \uparrow s) &= \delta_{\mathsf{C}}(t) \uparrow s && \text{if } s \in \mathsf{C}^* \\
\delta_{\mathsf{C}}(t \uparrow s) &= \delta_{\mathsf{C}}(t) && \text{if } s \notin \mathsf{C}^* \\
\delta_{\mathsf{C}}(t^{-1}) &= \delta_{\mathsf{C}}(t) && \\
\delta_{\mathsf{C}}(f(t_1, \ldots, t_n)) &= f(\delta_{\mathsf{C}}(t_1), \ldots, \delta_{\mathsf{C}}(t_n)) && \text{for } f \notin \{\uparrow, \cdot^{-1}\}.
\end{aligned}$$

For instance, let $\mathsf{C} = \{a, b\}$. Then, $\delta_{\mathsf{C}}(x^{-1} \uparrow a^{-1} \uparrow y \uparrow b) = x \uparrow a^{-1} \uparrow b$. (Recall that the symbol $\uparrow$ is left-associative, i.e., $x^{-1} \uparrow a^{-1} \uparrow y \uparrow b$ stands for $((x^{-1} \uparrow a^{-1}) \uparrow y) \uparrow b$.)

The function δ_{C} is extended to atoms, sets of terms, Horn clauses, Horn theories, and derivations in the obvious way. When C is fixed and known from the context, we simply write δ instead of δ_{C}. The following lemma, which summarizes some basic properties of δ_{C}, is easy to prove.

Lemma 4. *For any set C of pure, ground terms and for every term t we have:*

1) $\delta_{\mathsf{C}}(\delta_{\mathsf{C}}(t)) = \delta_{\mathsf{C}}(t)$.
2) $\delta_{\mathsf{C}}(t)$ *is C-exponent-ground.*
3) $\delta_{\mathsf{C}}(t) = t$ *iff t is C-exponent-ground.*

We say that a term t is *exponent-reduced* if every subterm s of t which occurs as an exponent, i.e., in a context of the form $s' \uparrow s$, is reduced. For example, if a, b, c are constants, then $(a \uparrow b) \uparrow b^{-1}$ is exponent-reduced, but $a \uparrow ((b \uparrow c) \uparrow c^{-1})$ is not. Note that every reduced term is exponent-reduced. The following lemma states that δ preserves the equivalence relation $\sim$ on exponent-reduced terms. The proof is given in the appendix.

Lemma 5. *For all exponent-reduced terms t and s, if $t \sim s$, then $\delta(t) \sim \delta(s)$.*

For the proof of Theorem 2, we also need the following lemma (see the appendix for the proof).

Lemma 6. *Let C be a set of pure, ground terms. Let t be a C-exponent-ground term, and θ be a substitution. Then, $\delta(t\theta) = t\,\delta(\theta)$.*

We are now ready to prove Theorem 2. Assume that $\pi = b_1, \ldots, b_l$ is a derivation for $T \cup T_{DH} \vdash_{DH} a$; in particular, $b_l \sim a$. Without loss of generality, we can assume that the atom a and all b_i are reduced, i.e., the terms in a and b_i are reduced. We will show that $\delta(\pi)$ is a derivation for $T \cup T_{DH}^{\mathsf{C}} \vdash_{DH} a$. This then completes the proof, because $\delta(\pi)$ is C-exponent-ground by Lemma 4.

Because $b_l \sim a$ and both b_l and a are reduced, by Lemma 5, we know that $\delta(b_l) \sim \delta(a)$. We also know, by Lemma 4, that $\delta(a) = a$ as a is C-exponent-ground. Thus, we have $\delta(b_l) \sim a$. Hence, to prove that $\delta(\pi)$ is a derivation for $T \cup T_{DH}^{\mathsf{C}} \vdash_{DH} a$, we only need to show for every $i \in \{1, \ldots, l\}$ that $\delta(b_i)$ can be obtained from $\{\delta(b_1), \ldots, \delta(b_{i-1})\}$ by applying one of the Horn clauses in $T \cup T_{DH}^{\mathsf{C}}$. So, let $i \in \{1, \ldots, l\}$. We will consider three cases, depending on whether b_i, in the derivation π, is obtained using a (C-exponent-ground) clause of T or a clause of T_{DH}. It is easy to check that in all three cases the substitution applied to the Horn clause in $T \cup T_{DH}^{\mathsf{C}}$ is C-exponent-ground.

Case 1: *b_i is obtained by applying some C-exponent-ground clause*: It follows that there exists a clause $a_1, \ldots, a_n \to a_0$ in T such that $a_0, \ldots, a_n$ are C-exponent-ground, and that there exists a substitution θ such that $a_0\theta \sim b_i$ and for every $j \in \{1, \ldots, n\}$ there exists $k_j \in \{1, \ldots, i-1\}$ with $a_j\theta \sim b_{k_j}$. We define $k_0 = i$. It is easy to check that, since a_j is C-exponent-ground and θ is reduced, the atom $a_j\theta$ is exponent-reduced. Hence, by Lemma 5, it is true that $\delta(a_j\theta) \sim \delta(b_{k_j})$, for all $j \in \{0, \ldots, n\}$. With Lemma 6, we obtain that $a_j\delta(\theta) \sim \delta(b_{k_j})$. So, we can apply the same clause $a_1, \ldots, a_n \to a_0$ in T with the substitution $\delta(\theta)$ to $\delta(b_{k_1}), \ldots, \delta(b_{k_n})$ and obtain $\delta(b_{k_0}) = \delta(b_i)$.

Case 2: *b_i is obtained by applying* (2): In this case b_i is of the form $I(t)$ and, for some $j < i$, the atom b_j is of the form $\mathrm{I}(s)$ with $t \sim s^{-1}$. Since t and s are reduced and thus both t and s^{-1} are exponent-reduced, we can use Lemma 5 to obtain $\delta(t) \sim \delta(s^{-1}) = \delta(s)$. Hence, $\delta(b_i) = \mathrm{I}(\delta(t)) \sim \mathrm{I}(\delta(s)) = \delta(b_j)$.

Case 3: *b_i is obtained by applying* (1): In this case, b_i is of the form $\mathrm{I}(t)$ and there are atoms $\mathrm{I}(s)$ and $\mathrm{I}(r)$ amongst $b_1, \ldots, b_{i-1}$ such that $t \sim s \uparrow r$. We want to show that $\mathrm{I}(\delta(t))$ can be obtained from $\mathrm{I}(\delta(s))$ and $\mathrm{I}(\delta(r))$.

Since s and r are reduced, we can observe that $s\uparrow r$ is exponent-reduced. So, by Lemma 5, we have $\delta(t) \sim \delta(s\uparrow r)$ and so it is enough to show that $\mathrm{I}(\delta(s\uparrow r))$ can be obtained from $\mathrm{I}(\delta(s))$ and $\mathrm{I}(\delta(r))$. Let us consider three subcases: (a) If $r \notin \mathsf{C}^*$, then $\delta(s\uparrow r) = \delta(s)$, so it is nothing to prove. (b) If $r \in \mathsf{C}$, then $\delta(r) = r$ and therefore $\delta(s\uparrow r) = \delta(s)\uparrow r = \delta(s)\uparrow\delta(r)$. Hence, $\mathrm{I}(\delta(s\uparrow r))$ can be obtained from $\mathrm{I}(\delta(s))$ and $I(\delta(r))$ using (7). (c) If $r \in \mathsf{C}^{-1}$, then $r = \delta(r)^{-1}$ and therefore $\delta(s\uparrow r) = \delta(s)\uparrow r = \delta(s)\uparrow\delta(r)^{-1}$. Hence, $\mathrm{I}(\delta(s\uparrow r))$ can be obtained from $\mathrm{I}(\delta(s))$ and $\mathrm{I}(\delta(r))$ using (8). (Note that $\delta(r) \in \mathsf{C}$.)

IV. The Reduction

In this section, we show how to construct from a C-exponent-ground theory T, a theory T_C such that for every C-exponent-ground atom b we have $T \cup T_{DH} \vdash_{DH} b$ if and only if $T_\mathsf{C} \vdash b'$, where b' is an encoding of b. In other words, we reduce the derivation problem modulo DH to a completely syntactical derivation problem. This syntactical derivation problem can then be dealt with by tools such as *ProVerif*. In this way, cryptographic protocols that employ DH can be analyzed w.r.t. an unbounded number of sessions and without a bound on the message size by tools that a priori cannot deal with DH.

In what follows, we first introduce a representation for C-exponent-ground terms that we will use in our reduction. The reduction itself is then presented in Section IV-B, along with the main theorem stating soundness and completeness of our reduction. We prove the main theorem in Section IV-C.

A. Encoding of C-exponent-ground Terms

Let Σ be a signature with $\uparrow, \cdot^{-1}$. Let $\mathsf{C} = \{c_1, \ldots, c_m\}$ be a set of pure, ground terms, for some m. From Section III we know that we only need to consider C-exponent-ground terms in derivations. We now introduce a representation for such terms, which will be used in our reduction.

The encoding of C-exponent-ground terms will be over the signature $\Sigma^{\mathsf{exp}} = (\Sigma \setminus \{\uparrow, \cdot^{-1}\}) \cup \{0, \mathsf{succ}, \mathsf{prev}, \mathsf{exp}\}$, where 0 is a constant, succ and prev are unary function symbols, and exp is a function symbol of arity $m+1$.

The symbols 0, succ, and prev will be used to encode integers: $\mathsf{succ}^n(0) = \mathsf{succ}(\mathsf{succ}(\cdots\mathsf{succ}(0)\cdots))$, with succ repeated n-times, will represent n, and analogously, $\mathsf{prev}^n(0)$ will represent $-n$. Terms of one of these forms are called *integer terms*. For an integer term t, we will denote by $t2i(t)$ the integer represented by t. Conversely, for an integer n, we will denote by $i2t(n)$ the integer term representing n. Note that $t2i(i2t(n)) = n$.

By definition of C-exponent-ground terms, we know that a C-exponent-ground term t with head symbol $\uparrow$ is equivalent ($\sim$) to a term of the form $s\uparrow c_1^{(n_1)}\uparrow\cdots\uparrow c_m^{(n_m)}$ for some integers $n_1, \ldots, n_m$. The idea is to represent t as the term $\mathsf{exp}(s, i2t(n_1), \ldots, i2t(n_m))$ over Σ^{exp}, i.e., the $(i+1)$-st argument of exp encodes the number of occurrences of c_i/c_i^{-1}. Before we define the encoding of C-exponent-ground terms formally, we need to introduce some notation.

For an integer term t, let $\mathrm{incr}(t) = t'$, if $t = \mathsf{prev}(t')$, and $\mathrm{incr}(t) = \mathsf{succ}(t)$, otherwise. Similarly, $\mathrm{decr}(t) = t'$, if $t = \mathsf{succ}(t')$, and $\mathrm{decr}(t) = \mathsf{prev}(t)$, otherwise. Obviously, we have $\mathrm{incr}(t) = i2t(t2i(t)+1)$ and $\mathrm{decr}(t) = i2t(t2i(t)-1)$.

For $i \in \{1, \ldots, m\}$, we define $\mathrm{incr}_i(\mathsf{exp}(t_0, \ldots, t_m))$ as t_0, if $t_i = \mathsf{prev}(0)$ and $t_j = 0$, for all $j \neq i$. Otherwise, we define:

$$\mathrm{incr}_i(\mathsf{exp}(t_0, \ldots, t_m)) = \mathsf{exp}(t_0, \ldots, t_{i-1}, \mathrm{incr}(t_i), t_{i+1}, \ldots, t_m).$$

In other words, applying incr_i corresponds to exponentiation with c_i.

Similarly, we define $\mathrm{decr}_i(\mathsf{exp}(t_0, \ldots, t_m))$ as t_0, if $t_i = \mathsf{succ}(0)$ and $t_j = 0$, for all $j \neq i$. Otherwise, we define:

$$\mathrm{decr}_i(\mathsf{exp}(t_0, \ldots, t_m)) = \mathsf{exp}(t_0, \ldots, t_{i-1}, \mathrm{decr}(t_i), t_{i+1}, \ldots, t_m).$$

Furthermore, if t is not of the form $\mathsf{exp}(t_0, t_1, \ldots, t_m)$, then we define $\mathrm{incr}_i(t) = \mathrm{incr}_i(\mathsf{exp}(t, 0, \ldots, 0))$ and $\mathrm{decr}_i(t) = \mathrm{decr}_i(\mathsf{exp}(t, 0, \ldots, 0))$.

Now, for a C-exponent-ground term t over Σ, we define its encoding $\ulcorner t \urcorner$, which is a term over Σ^{exp}, recursively as follows:

$$\begin{aligned}
\ulcorner x \urcorner &= x && \text{for a variable } x,\\
\ulcorner f(t_1, \ldots, t_n) \urcorner &= f(\ulcorner t_1 \urcorner, \ldots, \ulcorner t_n \urcorner) && \text{for } f \neq \uparrow \text{ and } f \neq \cdot^{-1}\\
\ulcorner t\uparrow c_i \urcorner &= \mathrm{incr}_i(\ulcorner t \urcorner),\\
\ulcorner t\uparrow c_i^{-1} \urcorner &= \mathrm{decr}_i(\ulcorner t \urcorner).
\end{aligned}$$

Note that $\ulcorner\cdot\urcorner$ does not need to be defined for the case $\ulcorner t^{-1}\urcorner$, since we only consider C-exponent-ground terms. For instance, for $\mathsf{C} = \{c_1, c_2, c_3\}$, we have

$$\ulcorner f(x\uparrow c_2\uparrow {c_2}^{-1})\uparrow c_1\uparrow {c_3}^{-1}\urcorner = \mathsf{exp}(f(x), \mathsf{succ}(0), 0, \mathsf{prev}(0)).$$

For an atom $a = p(t)$, we define $\ulcorner a \urcorner = p(\ulcorner t \urcorner)$.

From the definition of $\ulcorner\cdot\urcorner$ it easily follows that C-exponent-ground terms that are equivalent modulo DH have the same representation (see the appendix for the proof).

Lemma 7. *For C-exponent-ground terms t and s, if $t \sim s$, then $\ulcorner t \urcorner = \ulcorner s \urcorner$.*

B. Computing T_C

Let T be a C-exponent-ground Horn theory and C be defined as in Section IV-A. We now show how to construct T_C from T and state our main result. The theory T_C is depicted in Figure 4. An explanation of the clauses of T_C follows.

The clauses (9)–(11) allow to derive any integer term. The clauses (12)–(13) allow to switch between t and $\mathsf{exp}(t, 0, \ldots, 0)$.

The clauses (14) are meant to simulate the clauses of T_{DH}^{C} for C-exponent-ground terms. If the intruder knows c_i, then he is allowed to exponentiate the term

$$\mathrm{I}(0) \tag{9}$$
$$\mathrm{I}(x) \;\to\; \mathrm{I}(\mathsf{succ}(x)) \tag{10}$$
$$\mathrm{I}(x) \;\to\; \mathrm{I}(\mathsf{prev}(x)) \tag{11}$$
$$\mathrm{I}(x) \;\to\; \mathrm{I}(\mathsf{exp}(x,0,\ldots,0)) \tag{12}$$
$$\mathrm{I}(\mathsf{exp}(x,0,\ldots,0)) \;\to\; \mathrm{I}(x) \tag{13}$$
$$\mathrm{I}(c_i),\ \mathrm{I}(y),\ \mathrm{I}(\mathsf{exp}(x_0,x_1,\ldots,x_m)) \;\to\; \mathrm{I}(\mathsf{exp}(x_0,\ldots,x_{i-1},y,x_{i+1},\ldots,x_m)) \quad \text{for each } c_i \in \mathsf{C} \tag{14}$$

$$\mathsf{E}(t, c_i, \mathrm{incr}_i(t)) \qquad \text{for each } c_i \in \mathsf{C} \text{ and } t \in A_i^+ \tag{15}$$
$$\mathsf{E}(t, c_i, \mathrm{decr}_i(t)) \qquad \text{for each } c_i \in \mathsf{C}^{-1} \text{ and } t \in A_i^- \tag{16}$$

where E is a new predicate not occurring in T, and A_i^+ and A_i^- are defined as follows:

$$A_i^+ = \{x, \mathsf{exp}(x_0,x_1,\ldots,x_m), \mathsf{exp}(x_0,\ldots,x_{i-1},\mathsf{prev}(x_i),x_{i+1},\ldots,x_m), \ulcorner x \uparrow {c_i}^{-1} \urcorner\},$$
$$A_i^- = \{x, \mathsf{exp}(x_0,x_1,\ldots,x_m), \mathsf{exp}(x_0,\ldots,x_{i-1},\mathsf{succ}(x_i),x_{i+1},\ldots,x_m), \ulcorner x \uparrow c_i \urcorner\}$$

$$\mathsf{E}(\ulcorner\theta(t_1')\urcorner, d_1, y_1), \ldots, \mathsf{E}(\ulcorner\theta(t_k')\urcorner, d_k, y_k),\ \ulcorner\theta(r_1)\urcorner, \ldots, \ulcorner\theta(r_n)\urcorner \;\to\; \ulcorner\theta(r_0)\urcorner, \quad \text{for each clause } r_1,\ldots,r_n \to r_0 \text{ in } T \tag{17}$$

where $t_1 = t_1' \uparrow d_1, \ldots, t_k = t_k' \uparrow d_k$ are all the non-ground, non-standard subterms of $r_0,\ldots,r_n$, and θ replaces every t_i by a fresh variable y_i. Applied to a term t, θ replaced the terms t_i by y_i in a top-down manner, i.e., bigger terms t_i are replaced first. Note that, since T is C-exponent-ground, it holds that $d_1,\ldots,d_k \in \mathsf{C} \cup \mathsf{C}^{-1}$.

Figure 4. The theory T_{C}.

with $c_i^{(n)}$ for some integer n. For example, if the intruder knows $\mathsf{exp}(t,\mathsf{prev}(0),\mathsf{succ}(0),0,\ldots)$, which represents $t \uparrow {c_1}^{-1} \uparrow c_2$, and the term $c_1 \in \mathsf{C}$, then the intruder can, for instance, use (14) with $y = \mathsf{succ}(\mathsf{succ}(0))$ to derive $\mathsf{exp}(t,\mathsf{succ}(\mathsf{succ}(0)),\mathsf{succ}(0),0,\ldots)$, which represents $t \uparrow c_1^{(2)} \uparrow c_2$ (the clauses (9)–(11) are used to derive $\mathsf{succ}(\mathsf{succ}(0))$. Two remarks are in order: First, our reduction also works if the clauses in (14) modeled exponentiation with c_i and ${c_i}^{-1}$ only, rather than multiple exponentiations with $c_i/{c_i}^{-1}$. However, (14) works better in combination with ProVerif. Second, the clauses (14) can be applied even if y is substituted by a *non*-integer term. However, as we will show, this neither spoils soundness nor completeness of our reduction.

The set of facts in (15) and (16) define the (new) predicate E, which expresses exponentiation for C-exponent-ground terms, as stated in the following lemma (which is easy to prove):

Lemma 8. *Let t be a ground term, $c \in \mathsf{C}$, and assume that t is C-exponent-ground. Then, $E(\ulcorner t \urcorner, c, \ulcorner t \uparrow c \urcorner)$ is an instance of* (15) *and $E(\ulcorner t \urcorner, c^{-1}, \ulcorner t \uparrow c^{-1} \urcorner)$ is an instance of* (16)*.*

Let us consider the following example of how E looks like for a given set C.

Example 4. For $\mathsf{C} = \{a,b\}$, the clauses given by (16) are:

$$\mathsf{E}(x,\ a^{-1},\ \mathsf{exp}(x,\mathsf{prev}(0),0)),$$
$$\mathsf{E}(x,\ b^{-1},\ \mathsf{exp}(x,0,\mathsf{prev}(0))),$$
$$\mathsf{E}(\mathsf{exp}(x_0,x_1,x_2),\ a^{-1},\ \mathsf{exp}(x_0,\mathsf{prev}(x_1),x_2)),$$
$$\mathsf{E}(\mathsf{exp}(x_0,x_1,x_2),\ b^{-1},\ \mathsf{exp}(x_0,x_1,\mathsf{prev}(x_2))),$$
$$\mathsf{E}(\mathsf{exp}(x_0,\mathsf{succ}(x_1),x_2),\ a^{-1},\ \mathsf{exp}(x_0,x_1,x_2)),$$
$$\mathsf{E}(\mathsf{exp}(x_0,x_1,\mathsf{succ}(x_2)),\ b^{-1},\ \mathsf{exp}(x_0,x_1,x_2)),$$
$$\mathsf{E}(\mathsf{exp}(x,\mathsf{succ}(0),0),\ a^{-1},\ x),$$
$$\mathsf{E}(\mathsf{exp}(x,0,\mathsf{succ}(0)),\ b^{-1},\ x).$$

Now, for $t = c \uparrow b$, the fact

$$\mathsf{E}(\ulcorner t \urcorner, b^{-1}, \ulcorner t \uparrow b^{-1} \urcorner) = \mathsf{E}(\mathsf{exp}(c,0,\mathsf{succ}(0)),\ b^{-1},\ c)$$

is an instance of the last of these clauses.

Now, let us consider the clauses given by (17) for some clause $A = (r_1,\ldots,r_n \to r_0)$ in T. Let us denote the clause in T_{C} resulting from A by A^*. First, we can observe, that if A does not contain the exponentiation symbol, then $A^* = A$. But if A contains some term of the form $t \uparrow d$ with a non-ground term t, then this term is replaced by a fresh variable y and the relation between t, d, and y is captured by adding $\mathsf{E}(t,d,y)$ to the clause. Similar steps are applied recursively to the remaining non-ground, non-standard subterms of A, including subterms of t. All terms are encoded using $\ulcorner . \urcorner$ to obtain terms over Σ^{exp}. The clauses (17) are further illustrated by the following examples.

Example 5. Suppose that the theory T contains a clause $A = \big(\mathrm{I}(x) \to \mathrm{I}(\mathsf{hash}(x{\uparrow}a))\big)$. The clause obtained from (17) for clause A is

$$\mathsf{E}(x,a,y_1),\ \mathrm{I}(x) \to \mathrm{I}(\mathsf{hash}(y_1)).$$

Because A contains only one non-ground, non-standard term $t_1 = t_1'{\uparrow}d_1$ with $t_1' = x$ and $d_1 = a$, the substitution θ replaces $t_1 = x{\uparrow}a$ by y_1. Now, suppose that T contains a clause

$$B = \big(\mathrm{I}(x) \to \mathrm{I}(\langle x{\uparrow}a^{-1}{\uparrow}b,\ c{\uparrow}b\rangle)\big).$$

There are two non-ground, non-standard subterms in this clause: $t_1 = x{\uparrow}a^{-1}$ and $t_2 = (x{\uparrow}a^{-1}){\uparrow}b$. Hence, θ replaces t_1 by y_1 and t_2 by y_2. Note also that $t_1' = x$ and $t_2' = t_1$. Thus, we have $\theta(t_2') = y_1$. According to (17) we obtain the following clause for B:

$$\mathsf{E}(x,a^{-1},y_1),\ \mathsf{E}(y_1,b,y_2),\ \mathrm{I}(x) \to \mathrm{I}(\langle y_2,\ \mathsf{exp}(c,0,\mathsf{succ}(0))\rangle).$$

Example 6. Recall that the theory T_{SB} for modeling the SIGMA-BASIC protocol is exponent-ground, and so the reduction described here can be applied to it. As an example, let us consider clause (4) of this theory, that is:

$$\mathrm{I}(x) \to \mathrm{I}(\langle g{\uparrow}m_{ba},\ b,\ \mathsf{sig}_{k_b}(\langle x, g{\uparrow}m_{ba}\rangle)),\ \mathsf{mac}_{\mathsf{hash}(x{\uparrow}m_{ba})}(b)\rangle).$$

According to (17) we obtain the following clause for (4), where $u = \ulcorner g{\uparrow}m_{ba}\urcorner$:

$$\mathsf{E}(x,m_{ba},y_1),\ \mathrm{I}(x) \to \mathrm{I}(\langle u,\ b,\ \mathsf{sig}_{k_b}(\langle x,u\rangle)),\ \mathsf{mac}_{\mathsf{hash}(y_1)}(b)\rangle) \quad (18)$$

From the above, we immediately obtain:

Proposition 1. *Given a* C*-exponent-ground Horn theory T, the theory T_{C} can be constructed efficiently. In particular, the size of T_{C} is polynomial in the size of T.*

Now, we state soundness and completeness of our reduction, the main technical result of this paper. To obtain soundness, we need to assume that T is *non-trivial*, i.e., there exists a ground term u such that $T \cup T_{DH} \vdash_{DH} \mathrm{I}(u)$. Obviously, this condition is satisfied for every reasonable theory.

Theorem 3. *Let T be a non-trivial,* C*-exponent-ground theory over Σ and $b = p(t)$ be a* C*-exponent-ground atom over Σ, with p being a predicate occurring in T. Then, $T \cup T_{DH} \vdash_{DH} b$ if and only if $T_{\mathsf{C}} \vdash \ulcorner b\urcorner$.*

Note that $T_{\mathsf{C}} \vdash \ulcorner b\urcorner$ is a purely syntactical derivation problem that can be solved by tools such as ProVerif or with the help of theorem provers.

C. Proof of Theorem 3

We begin the proof of Theorem 3 with a lemma that, together with Theorem 2, immediately establishes completeness of our reduction, i.e., $T \cup T_{DH} \vdash_{DH} b$ implies $T_{\mathsf{C}} \vdash \ulcorner b\urcorner$.

Lemma 9. *If there is a* C*-exponent-ground derivation for $T \cup T_{DH}^{\mathsf{C}} \vdash_{DH} b$ obtained using* C*-exponent-ground substitutions, then $T_{\mathsf{C}} \vdash \ulcorner b\urcorner$.*

Proof: Let $\pi = b_1,\ldots,b_l$ be a C-exponent-ground derivation for $T \cup T_{DH}^{\mathsf{C}} \vdash_{DH} b$ obtained using C-exponent-ground substitutions. The proof proceeds by induction on the length of π. For $l = 0$, nothing is to show. Now, let $\pi_{<l} = b_1,\ldots,b_{l-1}$. We know that b ($\sim b_l$) can be derived modulo $\sim$ form $\pi_{<l}$ by applying a clause from $T \cup T_{DH}^{\mathsf{C}}$, using a C-exponent-ground substitution σ. To complete the proof, it is enough to show that $\ulcorner b\urcorner$ can syntactically be derived from $\ulcorner \pi_{<l}\urcorner$ using T_{C}. We consider two cases:

Case 1: b is obtained using a clause of T_{DH}^{C}: So, $b = \mathrm{I}(t)$, for some C-exponent-ground term t such that the set $\pi_{<l}$ contains atoms $\mathrm{I}(r)$, for some C-exponent-ground r, and $\mathrm{I}(c_i)$, for $c_i \in \mathsf{C}$, such that $t \sim r{\uparrow}c_i$ or $t \sim r{\uparrow}c_i^{-1}$. The atom $\mathrm{I}(\ulcorner t\urcorner)$ can be obtained from $\mathrm{I}(\ulcorner r\urcorner)$ and $\mathrm{I}(\ulcorner c_i\urcorner) = \mathrm{I}(c_i)$ using the clauses (9)–(14): If the reduced form of r is standard, then first clause (12) is used. Then, clause (14) is used with an appropriate integer term y, derived by (9)–(11). Finally, if the reduced form of t is standard, then clause (13) is applied.

Case 2: b is obtained using some (C-exponent-ground) clause $r_1,\ldots,r_n \to r_0$ of T: In this case, there exists some C-exponent-ground substitution σ such that $b \sim \sigma(r_0)$ and all $\sigma(r_1),\ldots,\sigma(r_n)$ belong to $\pi_{<l}$ (modulo $\sim$). We will obtain $\ulcorner b\urcorner$ using the clause (17) for $r_1,\ldots,r_n \to r_0$. Let us denote this clause by $R \to S$. Let t_i, t_i', d_i, y_i, and θ be defined as in (17).

We define a substitution σ^*, which will be applied to $R \to S$ to obtain $\ulcorner b\urcorner$, as follows: $\sigma^*(x) = \ulcorner\sigma(x)\urcorner$, for $x \in \mathrm{var}(r_0,\ldots,r_n)$, and $\sigma^*(y_i) = \ulcorner\sigma(t_i)\urcorner$. It is easy to show by induction on the size of terms that, for each subterm u of $r_0,\ldots,r_n$ which is not of the form w^{-1}, we have $\sigma^*(\ulcorner\theta(u)\urcorner) = \ulcorner\sigma(u)\urcorner$: If u is standard, then the claim immediately follows by the induction hypothesis. If u is a ground, non-standard term, then both $\sigma^*(\ulcorner\theta(u)\urcorner)$ and $\ulcorner\sigma(u)\urcorner$ are equal to $\ulcorner u\urcorner$. Finally, if u is a non-ground and non-standard, i.e. $u \in \{t_1,\ldots t_k\}$, then $\theta(u) = y_i$ for some i. Now, the claim follows immediately from the definition of σ^*.

As a result, we obtain $\sigma^*(\ulcorner\theta(r_i)\urcorner) = \ulcorner\sigma(r_i)\urcorner$, for $i \in \{0,\ldots,n\}$; in particular, $\sigma^*(\ulcorner\theta(r_i)\urcorner) \in \ulcorner\pi_{<l}\urcorner$ for every $i \in \{1,\ldots,n\}$ and we obtain $\sigma^*(\ulcorner\theta(r_0)\urcorner) = \ulcorner\sigma(r_0)\urcorner = \ulcorner b\urcorner$ by applying $R \to S$ with σ^*, where the latter equality follows from Lemma 7. It remains to prove that $\sigma^*(\mathsf{E}(\ulcorner\theta(t_i')\urcorner, d_i, y_i))$ can be derived from T_{C}.

We have $\sigma^*(\mathsf{E}(\ulcorner\theta(t_i')\urcorner, d_i, y_i)) = \mathsf{E}(\sigma^*(\ulcorner\theta(t_i')\urcorner), d_i, \sigma^*(y_i))$, which by the above, is equal to $\mathsf{E}(\ulcorner\sigma(t_i')\urcorner, d_i, \ulcorner\sigma(t_i)\urcorner)$ and therefore to $\mathsf{E}(\ulcorner\sigma(t_i')\urcorner, d_i, \ulcorner\sigma(t_i'){\uparrow}d_i\urcorner)$. By Lemma 8, this fact

is an instance of (15) or (16), depending on whether d_i belongs to C or to C^{-1}. Hence, $\sigma^*(\mathsf{E}(\ulcorner\theta(t_i')\urcorner, d_i, y_i))$ can be derived from T_C. ∎

We now turn to the soundness of our reduction, i.e., we prove that $T_\mathsf{C} \vdash \ulcorner b \urcorner$ implies $T \cup T_{DH} \vdash_{DH} b$. Here we use the non-triviality of T, i.e., there exists a ground term u such that $T \cup T_{DH} \vdash_{DH} \mathrm{I}(u)$. For the proof of soundness we need to introduce some notation. The proof also uses several lemmas.

In Section IV-A, we defined the function $t2i(\cdot)$ on integer terms. Now, we extend the domain of $t2i(\cdot)$ to all terms as follows:

$$\begin{aligned} t2i(0) &= 0, \\ t2i(\mathsf{succ}(t)) &= t2i(t)+1, \\ t2i(\mathsf{prev}(t)) &= t2i(t)-1, \\ t2i(t) &= 0, \quad \text{for } t \text{ not of the form } 0, \mathsf{succ}(t'), \\ &\qquad \text{or } \mathsf{prev}(t') \end{aligned}$$

We also define a mapping $\llcorner\cdot\lrcorner$ from terms over Σ^{exp} to terms over Σ:

$$\begin{aligned} \llcorner x \lrcorner &= x, \quad \text{for a variable } x \\ \llcorner 0 \lrcorner &= u \\ \llcorner \mathsf{succ}(t) \lrcorner &= u \\ \llcorner \mathsf{prev}(t) \lrcorner &= u \\ \llcorner \mathsf{exp}(t, s_1, \ldots, s_m) \lrcorner &= \llcorner t \lrcorner \uparrow c_1^{(t2i(s_1))} \uparrow \cdots \uparrow c_m^{(t2i(s_m))} \\ \llcorner f(t_1, \ldots, t_n) \lrcorner &= f(\llcorner t_1 \lrcorner, \ldots, \llcorner t_n \lrcorner), \end{aligned}$$

where f is neither 0, succ, prev, nor exp, and u is defined as above, i.e., we have $T \cup T_{DH} \vdash_{DH} \mathrm{I}(u)$. For an atom $p(t)$, we define $\llcorner p(t) \lrcorner$ as $p(\llcorner t \lrcorner)$.

The relationship between $\ulcorner\cdot\urcorner$ and $\llcorner\cdot\lrcorner$ is captured by the following lemma (see the appendix for the proof).

Lemma 10. *Let t be a C-exponent-ground term over Σ. Then, $t \sim \llcorner \ulcorner t \urcorner \lrcorner$.*

The following lemma states a basic property of the predicate E, which occurs in T_C. Recall that this predicate is defined by a set of facts in T_C, and thus $\mathsf{E}(t,d,s)$ can be derived from T_C if and only if $\mathsf{E}(t,d,s)$ is an instance of some fact $\mathsf{E}(t',d',s')$ in T_C.

Lemma 11. *Let t, d, and s be ground terms over Σ^{exp}. If $\mathsf{E}(t,d,s)$ can be derived from T_C, then $d \in \mathsf{C} \cup \mathsf{C}^{-1}$ and $\llcorner s \lrcorner \sim \llcorner t \lrcorner \uparrow d$.*

The proof of this lemma can easily be carried out by case distinction, which we do not present here. We only illustrate the lemma by one concrete example: In Example 4, $\mathsf{E}(\mathsf{exp}(x_0,x_1,x_2),\ a^{-1},\ \mathsf{exp}(x_0,\mathsf{prev}(x_1),x_2))$ is a fact of T_C. Consider the substitution $\sigma = \{g/x_0, g/x_1, 0/x_2\}$, for a constant g. Then, $\mathsf{E}(\mathsf{exp}(g,g,0),\ a^{-1},\ \mathsf{exp}(g,\mathsf{prev}(g),0))$ is an instance of T_C. We indeed have $\llcorner \mathsf{exp}(g,\mathsf{prev}(g),0) \lrcorner \sim g \uparrow a^{-1} \sim \llcorner \mathsf{exp}(g,g,0) \lrcorner \uparrow a^{-1}$. (In this case, we even have syntactical equality.)

The following lemma is the main lemma for proving soundness of our reduction.

Lemma 12. *Let $a = p(t)$ be an atom, such that p occurs in T. Then, $T_\mathsf{C} \vdash a$ implies $T \cup T_{DH} \vdash_{DH} \llcorner a \lrcorner$.*

Proof: Let $\pi = a_1, \ldots, a_l$ be a (syntactic) derivation for $T_\mathsf{C} \vdash a$. The proof proceeds by induction on the length of π. For $l = 0$, nothing is to show. For the induction step we show that $\llcorner a_l \lrcorner$ can be derived from $\llcorner \pi_{<l} \lrcorner$, where $\pi_{<l} = a_1, \ldots, a_{l-1}$ and $\llcorner \pi_{<l} \lrcorner$ is the sequence of atoms obtained from $\pi_{<l}$ by removing all atoms of the form $\mathsf{E}(\ldots)$ and by replacing all the remaining atoms a_i by $\llcorner a_i \lrcorner$.

By assumption, a, and hence, a_l, is not of the form $\mathsf{E}(\ldots)$, as E is a predicate symbol that does not occur in T. Therefore, it suffices to consider the following cases:

Case 1. a_l is obtained using (9)–(11): So, a_l is of the form $\mathrm{I}(0)$, $\mathrm{I}(\mathsf{succ}(t))$, or $\mathrm{I}(\mathsf{prev}(t))$. Therefore, $\llcorner a_l \lrcorner = \mathrm{I}(u)$. By definition of u, we have $T \cup T_{DH} \vdash_{DH} \mathrm{I}(u)$.

Case 2. a_l is obtained using (12) *or* (13): It is enough to note that $\llcorner t \lrcorner = \llcorner \mathsf{exp}(t, 0, \ldots, 0) \lrcorner$. So, if the left-/right-hand side can be derived from $\llcorner \pi_{<l} \lrcorner$, then so can the right-/left-hand side.

Case 3. a_l is obtained using (14): Therefore, the atom a_l is of the form $\mathrm{I}(\mathsf{exp}(s_0, \ldots, s_{i-1}, s_i', s_{i+1} \ldots, s_m))$ such that $\mathrm{I}(\mathsf{exp}(s_0, \ldots, s_m))$, $\mathrm{I}(c_i)$, and $\mathrm{I}(s_i')$ occur in $\pi_{<l}$. We set $b = \mathrm{I}(\mathsf{exp}(s_0, \ldots, s_m))$. Thus, $\llcorner b \lrcorner = \mathrm{I}(\llcorner \mathsf{exp}(s_0, \ldots, s_m) \lrcorner)$, and $\llcorner \mathrm{I}(c_i) \lrcorner = \mathrm{I}(c_i)$ are elements of $\llcorner \pi_{<i} \lrcorner$.

If $t2i(s_i') > t2i(s_i)$, then, to derive $\llcorner a_l \lrcorner$ from $\llcorner b \lrcorner$ and $\mathrm{I}(c_i)$, the clause $\mathrm{I}(x), \mathrm{I}(y) \to \mathrm{I}(x \uparrow y)$ is applied a number of times, namely $t2i(s_i') - t2i(s_i)$ times. If $t2i(s_i') < t2i(s_i)$, then first the clause $\mathrm{I}(x) \to \mathrm{I}(x^{-1})$ is applied to $\mathrm{I}(c_i)$ and then the clause $\mathrm{I}(x), \mathrm{I}(y) \to \mathrm{I}(x \uparrow y)$ is applied a number of times, namely $t2i(s_i) - t2i(s_i')$ times. Otherwise, if $t2i(s_i') = t2i(s_i)$, then $\llcorner a_l \lrcorner$ is simply a repetition of $\llcorner b \lrcorner$.

Case 4. a_l is obtained using (17): Let $r_1, \ldots, r_n \to r_0$, t_i, t_i', d_i, y_i, and θ be as in (17). Assume that to obtain a_l, (17) was instantiated with a substitution σ. Hence, $a_l = \sigma(\ulcorner\theta(r_0)\urcorner)$. Furthermore, all the $\sigma(\ulcorner\theta(r_i)\urcorner)$, for $i \in \{1, \ldots, n\}$, and $\mathsf{E}(\sigma(\ulcorner\theta(t_i')\urcorner), d_i, \sigma(y_i))$, for $i \in \{1, \ldots, k\}$, are in $\pi_{<l}$. Therefore, $\llcorner \sigma(\ulcorner\theta(r_i)\urcorner) \lrcorner$, for $i \in \{1, \ldots, n\}$, are in $\llcorner \pi_{<l} \lrcorner$ and, by Lemma 11, $\llcorner \sigma(y_i) \lrcorner \sim \llcorner \sigma(\ulcorner\theta(t_i')\urcorner) \lrcorner \uparrow d_i$.

Let $\sigma^*(x) = \llcorner \sigma(x) \lrcorner$. For each subterm t of $r_0, \ldots, r_n$ such that t is not of the form w^{-1}, we show, by induction on the size of t, that $\sigma^*(t) \sim \llcorner \sigma(\ulcorner\theta(t)\urcorner) \lrcorner$. We consider the following cases:

(a) $t = x$ is a variable: Then, $\ulcorner\theta(x)\urcorner = x$, and thus $\sigma^*(x) = \llcorner \sigma(\ulcorner\theta(x)\urcorner) \lrcorner$, by the definition of σ^*.
(b) $t = f(t_1, \ldots, t_n)$, for $f \neq \uparrow$: The claim easily follows by induction.

(c) $t = t' \uparrow d$ and t is ground: Then $\llcorner\sigma(\ulcorner\theta(t)\urcorner)\lrcorner = \llcorner\ulcorner t\urcorner\lrcorner$ and $\sigma^*(t) = t$. We know that $\llcorner\ulcorner t\urcorner\lrcorner \sim t$, by Lemma 10.

(d) $t = t_i = t_i' \uparrow d_i$: Then, we have $\sigma^*(t_i) = \sigma^*(t_i') \uparrow d_i \sim \llcorner\sigma(\ulcorner\theta(t_i')\urcorner)\lrcorner \uparrow d_i$, by the inductive hypothesis. As we have noticed, $\llcorner\sigma(\ulcorner\theta(t_i')\urcorner)\lrcorner \uparrow d_i \sim \llcorner\sigma(y_i)\lrcorner$. Therefore, $\sigma^*(t_i) \sim \llcorner\sigma(y_i)\lrcorner = \llcorner\sigma(\ulcorner\theta(t_i)\urcorner)\lrcorner$, by the definition of θ and $\ulcorner\cdot\urcorner$.

By the above, we have, in particular, that $\sigma^*(r_i) \sim \llcorner\sigma(\ulcorner\theta(r_i)\urcorner)\lrcorner$. (Note that r_i is not of the form w^{-1}, since it is C-exponent-ground). Recall that $\llcorner\sigma(\ulcorner\theta(r_i)\urcorner)\lrcorner$, for $i \in \{1,\ldots,n\}$, are in $\llcorner\pi_{<l}\lrcorner$, which means that we can apply the clause $r_1,\ldots,r_n \to r_0$ with σ^* to obtain $\sigma^*(r_0) \sim \llcorner\sigma(\ulcorner\theta(r_0)\urcorner)\lrcorner = \llcorner a_l \lrcorner$. ■

With the above, soundness of our reduction follows easily: Suppose that $T_{\mathsf{C}} \vdash \ulcorner b\urcorner$. By assumption, $b = p(t)$ where p occurs in T. Thus, Lemma 12 implies $T \cup T_{DH} \vdash_{DH} \llcorner\ulcorner b\urcorner\lrcorner$. By Lemma 10, $b \sim \llcorner\ulcorner b\urcorner\lrcorner$, and therefore, $T \cup T_{DH} \vdash_{DH} b$.

V. Implementation and Experiments

We have implemented our reduction, and together with ProVerif, tested it on a set of protocols which employ Diffie-Hellman exponentiation. As mentioned in the introduction, our implementation, along with the Horn theory models of the protocols discussed in this section, is available at [20].

In this section, we briefly discuss our implementation, which closely follows the reduction presented in Section IV, and our experimental results.

A. Implementation

We have implemented our reduction in SWI-Prolog (version 5.6.14). Our implementation essentially takes a Horn theory as input, modeling the protocol and the intruder, as described in Section II. More precisely, the input consists of (1) a declaration of all the functor symbols used in the protocol and by the intruder, (2) the initial intruder facts as well as the protocol and intruder rules, except for the DH-rules (DH1) to (DH3), which are assumed implicitly, (3) a statement which defines a secrecy goal. Moreover, options, which are handed over to ProVerif, may be added.

Our implementation then first checks whether the given Horn theory, say T, (part (2) of the input) is exponent-ground. If it is not, an error message is returned. If it is, a set C is computed such that the Horn theory is C-exponent-ground. Recall that such a set always exists if the Horn theory is exponent-ground. Also it is straightforward to compute C for an exponent-ground theory: all one has to do is to collect all the ground exponents occurring in the terms of T. Once C is computed, the reduction as described in Section IV is applied to T, i.e., T_{C} is computed. Now, T_{C} together with the rest of the original input is passed on to ProVerif. This tool then does the rest of the work, i.e., it checks the goals for T_{C}. This is possible since the resulting theory T_{C} is meant to be interpreted in a free algebra, i.e., no algebraic properties are associated with the function symbols.

protocol	correct	reduction time	ProVerif time
BADTH	no	0.02s	0.02s
STS	yes	0.02s	0.02s
STS-CA	no	0.02s	0.03s
ISO-KE	yes	0.02s	0.02s
SIGMA-BASIC	yes	0.02s	0.05s
SIGMA-I	yes	0.02s	0.04s
SIGMA-R	yes	0.02s	0.03s
JFKi	yes	0.03s	0.06s
JFKr	yes	0.03s	0.06s
KERBEROS	no	0.02s	0.09s
KERBEROS-fix	yes	0.02s	0.07s
SSH	yes	0.03s	0.05s
IKEv2-DS	no	0.04s	0.14s
IKEv2-DS-fix	yes	0.03s	0.07s
IKEv2-MAC	yes	0.03s	0.04s
IKEv2-Child	yes	0.03s	0.06s
A-GDH.2-1S	yes	0.03s	4.44s
A-GDH.2-2S	no	0.03s	7.65s

Figure 5. Experimental Results.

B. Experiments

We tested our implementation on a set of exponent-ground protocols. The results, obtained by running our implementation on a 2,4 Ghz Intel CoreTM 2 Duo E6700 processor with 2GB RAM, are depicted in Figure 5, where we list both the time of the reduction and the time ProVerif needed for the analysis of the output of the reduction. In the 'correct' column, we indicate, whether the tool proves the secrecy properties we have specified. As can be seen from Figure 5, many of the protocols that we analyzed are important practical protocols. Along the lines of the specification of our running example, the analysis of these protocols is performed w.r.t. a bounded number of parties (from two to five) and an unbounded number of sessions (except for A-GDH.2). As mentioned in Section II-C, for checking secrecy-like properties, bounding the number of parties is a safe simplification in the sense that no attacks are excluded.

The performance of our tool (including running ProVerif on the result of our reduction) is very good: In fact, in almost all the cases (except for the protocol A-GDH.2) the total running time—reduction time plus the time ProVerif needed—is far less than one second. These experiments demonstrate that our reduction applied together with ProVerif constitutes a quite efficient and robust automatic analysis method for protocols that employ Diffie-Hellman exponentiation.

Let us now briefly discuss the protocols and security properties that we analyzed in more detail.

BADTH is a key exchange protocol discussed in [19], which uses digital signatures for authenticated Diffie-Hellman key exchange. While this protocol guarantees secrecy of session keys, it allows for an attack resulting in the following situation: Honest A successfully completes a protocol run and thinks to have established the session key s with honest B. However, B thinks to have established the *same* session key s with dishonest C (although C does not necessarily know s). This failure of the so-called *consistency requirement*, which constitutes a specific authentication property, was first shown in [13].

We model consistency requirements as follows: We introduce new atoms of the form $\mathsf{completedI}(a,b,S)$ and $\mathsf{completedR}(b,a,S)$. The former means that the initiator a believes to have completed a session with the responder b and to have established secret S with b. The atom $\mathsf{completedR}(b,a,S)$ is the same from the point of view of the responder b. Given these atoms, consistency requirements are expressed by clauses of the form:

$$\mathsf{completedI}(a,b,S),\ \mathsf{completedR}(b,c,S) \rightarrow \mathsf{Sec},$$

where Sec is a constant that should be kept secret, i.e., not derivable. As expected, our tool, when applied to BADTH detects the failure of the consistency requirement sketched above.

STS stands for STS-Basic, a protocol designed by Diffie et al. [13]. This protocol is meant to fix the problem explained for BADTH above. In addition to signatures, it uses encryption with the key derived from the Diffie-Hellman key exchange in order to prove knowledge of this key. Our tool proved this protocol secure w.r.t. secrecy properties and consistency requirements.

In [19] it is shown that STS is vulnerable to a consistency attack, if parties can register public keys without proving knowledge of the corresponding private key (which, as pointed in [19], is quite common). We model this *improper key registration* in STS-CA and reproduce the known attack with our tool.

ISO-KE is a protocol introduced in [16]. It is similar to STS. Our tool proved this protocol secure w.r.t. secrecy of session keys and consistency requirements as explained above. Again we model improper key registration.

SIGMA [19] is a family of protocols for authenticated key-exchange. This family serves as the basis for the signature-based modes of the IKE protocol (version 1 and 2). We have analyzed three variants of SIGMA: SIGMA-BASIC, SIGMA-I, and SIGMA-R. We used SIGMA-BASIC as our running example (see Section II-C). SIGMA-I and SIGMA-R extend SIGMA-BASIC in that they aim to provide identity protection of the initiator and the responder, respectively. Our tool proved these protocols secure w.r.t. the secrecy of session keys and consistency requirements, again allowing for improper key registration. However, checking identity protection is beyond the kind of security properties we can analyze with our tool.

JFK is a well-known key exchange protocol proposed in [2]. Here JFKi and JFKr stand for two versions of this protocol aiming to achieve initiator and responder identity protection, respectively. We proved these protocols secure w.r.t. security properties similar to those studied for the SIGMA protocols.

KERBEROS stands for the Kerberos 5 protocol (intra-real) with DHINIT [26], [27], [33]. Our modeling of this protocol follows the one of [28]. Our tool reproduces a known attack on this protocol reported in [28]. In this attack certain authentication requirements, which in this case can be modeled as reachability properties, are violated. However, our tool was able to establish other authentication and secrecy properties, where again the authentication properties that we considered were modeled as reachability properties. KERBEROS-fix is a version of Kerberos with a fix proposed in [28], which prevents the attack mentioned above. Our tool proves this protocol secure w.r.t. the security properties also analyzed for KERBEROS.

SSH refers to the SSH Transport Layer Protocol [32]. We have modeled this protocol along the lines of the AVISPA library [3], [4], of course w.r.t. an unbounded number of sessions. Our tool establishes secrecy and weak authentication properties for this protocol.

Finally, IKEv2-Child, IKEv2-DS, and IKEv2-MAC are sub-protocols of the IKEv2 protocol [18]. Our tool reproduces a known attack on IKEv2-DS. A fix of this protocol and the other two protocols are shown to be secure by our tool w.r.t. secrecy properties and weak authentication properties. The modeling of these sub-protocols of IKE follow the one of the AVISPA library.

We finally note that we also tried to analyze the A-GDH.2 protocol, a group protocol based on Diffie-Hellman key exchange. While we could apply our reduction easily (the protocol is exponent-ground), ProVerif, when running on the result of this reduction, did not terminate. This appears to be due to clauses of the form $\mathrm{I}(x),\ \ldots\ \rightarrow\ \mathrm{I}(x{\uparrow}c),\ \ldots$, where $x{\uparrow}c$ does not occur in the context of a more complex term. However, we were able to analyze A-GDH.2 w.r.t. a model that (safely) approximates a *bounded* number of protocol sessions. In this modeling of A-GDH.2 we used a technique inspired by the one sometimes used in the process calculus mode of ProVerif when encoding phases. We proved secrecy properties for A-GDH.2 in case of one session with four honest participants (A-GDH.2-1S) and discovered an attack that occurs if A-GDH.2 is run in a setting with two sessions (A-GDH.2-2S); A-GDH.2 is known to be flawed in this setting.

VI. Conclusion and Future Work

We presented a method for reducing the derivation problem modulo DH for Horn theories to a purely syntactical derivation problem. The reduction works for a large class of Horn theories, namely exponent-ground Horn theories, and hence, can be applied in the analysis of all protocols and intruder capabilities that can be modeled as exponent-ground Horn theories. We implemented our reduction and, in combination with ProVerif, successfully applied it in the automatic analysis of several state-of-the-art protocols that use Diffie-Hellman Exponentiation. This presents the first practical method for automatic protocol analysis w.r.t. an unbounded number of sessions that achieves this level of accuracy in terms of the algebraic properties covered for DH. In particular, our method yields more precise analysis results and covers a wider range of protocols than previous approaches.

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

In this work, we concentrated on secrecy properties, although we also analyzed simple authentication properties. It would be interesting to extend our approach to other security properties, such a stronger authentication properties and observational equivalence. We note that the translation of processes to Horn theories, as done by ProVerif for these properties, often leads to non-exponent-ground theories. It seems difficult to avoid this effect.

References

[1] M. Abadi, B. Blanchet, and C. Fournet. Just Fast Keying in the Pi Calculus. In D. Schmidt, editor, *Programming Languages and Systems: Proceedings of the 13th European Symposium on Programming (ESOP'04)*, volume 2986 of *Lecture Notes on Computer Science*, pages 340–354. Springer Verlag, 2004.

[2] W. Aiello, S. M. Bellovin, M. Blaze, J. Ioannidis, O. Reingold, R. Canetti, and A. D. Keromytis. Efficient, DoS-resistant, secure key exchange for internet protocols. In V. Atluri, editor, *ACM Conference on Computer and Communications Security*, pages 48–58. ACM, 2002.

[3] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In K. Etessami and S. Rajamani, editors, *Computer Aided Verification, 17th International Conference (CAV 2005)*, volume 3576 of *Lecture Notes in Computer Science*, pages 281–285. Springer-Verlag, 2005.

[4] The AVISPA Library of Protocols. http://avispa-project.org/library/.

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

[6] B. Blanchet, M. Abadi, and C. Fournet. Automated Verification of Selected Equivalences for Security Protocols. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005)*, pages 331–340. IEEE Computer Society, 2005.

[7] B. Blanchet, M. Abadi, and C. Fournet. Automated verification of selected equivalences for security protocols. *Journal of Logic and Algebraic Programming*, 75(1):3–51, 2008.

[8] M. Boreale and M. Buscemi. On the Symbolic Analysis of Low-Level Cryptographic Primitives: Modular Exponentiation and the Diffie-Hellman Protocol. In *Proceedings of the Workshop on Foundations of Computer Security (FCS 2003)*, 2003.

[9] M. Boreale and M. G. Buscemi. Symbolic analysis of cryptoprotocols based on modular exponentiation. In B. Rovan and P. Vojtás, editors, *Mathematical Foundations of Computer Science 2003, 28th International Symposium (MFCS 2003)*, volume 2747 of *Lecture Notes in Computer Science*, pages 269–278. Springer, 2003.

[10] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the Security of Protocols with Diffie-Hellman Exponentiation and Products in Exponents. In P. Pandya and J. Radhakrishnan, editors, *FSTTCS 2003: Foundations of Software Technology and Theoretical Computer Science*, volume 2914 of *Lecture Notes in Computer Science*, pages 124–135. Springer, 2003. A full version of this paper was published in ACM Transactions on Computational Logic (TOCL), 9(4), 2008.

[11] Y. Chevalier and M. Rusinowitch. Hierarchical Combination of Intruder Theories. In F. Pfenning, editor, *Term Rewriting and Applications, 17th International Conference, RTA 2006, Proceedings*, volume 4098 of *Lecture Notes in Computer Science*, pages 108–122. Springer, 2006.

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

[13] W. Diffie, P. C. van Oorschot, and M. J. Wiener. Authentication and Authenticated Key Exchanges. *Designes Codes and Cryptography*, 2(2):107–125, 1992.

[14] S. Escobar, C. Meadows, and J. Meseguer. State Space Reduction in the Maude-NRL Protocol Analyzer. In S. Jajodia and J. López, editors, *Computer Security – ESORICS 2008, 13th European Symposium on Research in Computer Security, Proceedings*, volume 5283 of *Lecture Notes in Computer Science*, pages 548–562. Springer, 2008.

[15] J. Goubault-Larrecq, M. Roger, and K. Verma. Abstraction and Resolution Modulo AC: How to Verify Diffie-Hellman-like Protocols Automatically. *Journal of Logic and Algebraic Programming*, 64(2):219–251, 2005.

[16] ISO/IEC IS 9798-3, Entity authentication mechanisms — Part 3: Entity authentication using assymetric techniques, 1993.

[17] D. Kapur, P. Narendran, and L. Wang. Analyzing protocols that use modular exponentiation: Semantic unification techniques. In R. Nieuwenhuis, editor, *Proceedings of the 14th International Conference on Rewriting Techniques and Applications (RTA 2003)*, volume 2706 of *Lecture Notes in Computer Science*, pages 165–179. Springer, 2003.

[18] C. Kaufman (Editor). Internet Key Exchange (IKEv2) Protocol. draft-ietf-ipsec-ikev2-17.txt, 2004. http://tools.ietf.org/html/draft-ietf-ipsec-ikev2-17.

[19] H. Krawczyk. SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols. In D. Boneh, editor, *Advances in Cryptology – CRYPTO 2003, 23rd Annual International Cryptology Conference*, volume 2729 of *Lecture Notes in Computer Science*, pages 400–425. Springer, 2003.

[20] R. Küsters and T. Truderung. DH-ProVerif Implementation. Available at http://www.infsec.uni-trier.de/tools.html.

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

[22] C. Lynch and C. Meadows. Sound Approximations to Diffie-Hellman Using Rewrite Rules. In J. Lopez, S. Qing, and E. Okamoto, editors, *Information and Communications Security, 6th International Conference (ICICS 2004)*, volume 3269 of *Lecture Notes in Computer Science*, pages 262–277. Springer, 2004.

[23] C. Meadows. Extending formal cryptographic protocol analysis techniques for group protocols and low-level cryptographic primitives. In P. Degano, editor, *Proceedings of the First Workshop on Issues in the Theory of Security (WITS'00)*, pages 87–92, 2000.

[24] C. Meadows and P. Narendran. A Unification Algorithm for the Group Diffie-Hellman Protocol. In *Workshop on Issues in the Theory of Security (WITS 2002)*, 2002.

[25] J. Millen and V. Shmatikov. Symbolic Protocol Analysis with Products and Diffie-Hellman Exponentiation. In *Proceedings of the 16th IEEE Computer Security Foundations Workshop (CSFW 16)*, pages 47–61. IEEE Computer Society, 2003.

[26] B. C. Neuman and T. Ts'o. Kerberos: An Authentication Service for Computer Networks. *IEEE Communications*, 32(9):33–38, 1994.

[27] C. Neuman, T. Yu, S. Hartman, and K. Raeburn. The Kerberos Network Authenitcation Service (V5). RFC 4120, 2005.

[28] A. Roy, A. Datta, and J. C. Mitchell. Formal Proofs of Cryptographic Security of Diffie-Hellman-Based Protocols. In G. Barthe and C. Fournet, editors, *Trustworthy Global Computing*, volume 4912 of *Lecture Notes in Computer Science*, pages 312–329. Springer, 2007.

[29] V. Shmatikov. Decidable Analysis of Cryptographic Protocols with Products and Modular Exponentiation. In D. Schmidt, editor, *13th European Symposium on Programming (ESOP 2004)*, volume 2986 of *Lecture Notes in Computer Science*, pages 355–369. Springer, 2004.

[30] M. Turuani. The CL-Atse Protocol Analyser. In F. Pfenning, editor, *Term Rewriting and Applications, 17th International Conference, RTA 2006, Proceedings*, volume 4098 of *Lecture Notes in Computer Science*, pages 277–286. Springer, 2006.

[31] K. Verma, H. Seidl, and T. Schwentick. On the complexity of equational horn clauses. In *Proceedings of the 20th International Conference on Automated Deduction (CADE 2005)*, volume 3328 of *Lecture Notes in Computer Science*, pages 337–352. Springer-Verlag, 2005.

[32] T. Ylonen. The Secure Shell (SSH) Transport Layer Protocol. RFC 4253, 2006.

[33] L. Zhu and B. Tung. Public key cryptography for initial authentication in kerberos (PKINIT). RFC 4556, 2006.

APPENDIX

A. Proof of Lemma 2

By Lemma 1, it suffices to show: $\mathrm{KT}(t) \equiv \mathrm{KT}(t')$ iff $\mathrm{KT}(t) \sim \mathrm{KT}(t')$. Note that to obtain $\mathrm{KT}(t)$, the term t is first normalized. Therefore, it is easy to see that both $\mathrm{KT}(t) \equiv \mathrm{KT}(t')$ and $\mathrm{KT}(t) \sim \mathrm{KT}(t')$ mean that $\mathrm{KT}(t)$ and $\mathrm{KT}(t')$ coincide up to (DH1). Now, the lemma easily follows.

B. Proof of Theorem 1

To prove Theorem 1, let us assume that π is a derivation of a from T modulo $\equiv$. We will show, by induction on the length of π, that there exists a derivation $\tilde{\pi}$ from $\mathrm{KT}(T)$ modulo $\sim$ such that whenever b is an element of π (modulo $\equiv$), then $\mathrm{KT}(b)$ is an element of $\tilde{\pi}$ (modulo $\sim$). In particular, this will imply $\mathrm{KT}(T) \vdash_{DH} \mathrm{KT}(a)$.

For $|\pi| = 0$, nothing is to show. So, suppose that $|\pi| > 0$. Let c be the last element of π, i.e. $\pi = \pi' c$. By the inductive hypothesis, there exists a derivation $\tilde{\pi}'$ from $\mathrm{KT}(T)$ modulo $\sim$ such that for every b in π' we have that $\mathrm{KT}(b)$ is in $\tilde{\pi}'$.

Now, if c is obtained using one of the Horn clauses in (6), i.e., c is of the form $\mathrm{I}(t_0 \uparrow t_1^{z_1} \times \cdots \times t_n^{z_n})$, for some integers $z_1, \ldots, z_n$ and some terms $t_0, \ldots, t_n$ such that $\mathrm{I}(t_0), \ldots, \mathrm{I}(t_n)$ occur in π' (modulo $\equiv$), then $\mathrm{I}(\mathrm{KT}(t_i))$ are in $\tilde{\pi}'$ and one can apply the rules of T_{DH} (possible a number of times) to obtain $\mathrm{KT}(c) \sim \mathrm{I}(\mathrm{KT}(t_0) \uparrow \mathrm{KT}(t_1)^{(z_1)} \uparrow \cdots \uparrow \mathrm{KT}(t_n)^{(z_n)})$. Therefore, we can extend $\tilde{\pi}'$ by a sequence $\tilde{\pi}''$ of atoms such that $\tilde{\pi} = \tilde{\pi}' \tilde{\pi}'' \, \mathrm{KT}(c)$ is a derivation.

If c is obtained using some other Horn clause in T of the form $s_1, \ldots, s_n \to s_0$ with a substitution σ, then one can use the clause $\mathrm{KT}(s_1), \ldots, \mathrm{KT}(s_n) \to \mathrm{KT}(s_0)$ in $\mathrm{KT}(T)$ with the substitution $\mathrm{KT}(\sigma)$, where $\mathrm{KT}(\sigma)(x) = \mathrm{KT}(\sigma(x))$ for all $x \in \mathrm{dom}(\sigma)$, to obtain $\mathrm{KT}(c)$: If b is an element of π' such that $b \equiv s_i\sigma$, for $i \in \{1, \ldots, n\}$, then $\mathrm{KT}(b)$ is in $\tilde{\pi}'$ by the induction hypothesis. By Lemma 1, we obtain $s_i \equiv \mathrm{KT}(s_i)$,

and $\sigma(x) \equiv \mathrm{KT}(\sigma(x))$. Since $\equiv$ is a congruence relation, it follows that $s_i\sigma \equiv \mathrm{KT}(s_i)\,\mathrm{KT}(\sigma)$. Now, since $b \equiv s_i\sigma$, Lemma 2 implies that $\mathrm{KT}(s_i)\,\mathrm{KT}(\sigma) \sim \mathrm{KT}(b)$. Analogously, it follows that $\mathrm{KT}(s_0)\,\mathrm{KT}(\sigma) \sim \mathrm{KT}(c)$. Hence, by applying $\mathrm{KT}(s_1),\ldots,\mathrm{KT}(s_n) \to \mathrm{KT}(s_0)$ with $\mathrm{KT}(\sigma)$ we can derive $\mathrm{KT}(c)$. Therefore, we can define $\tilde{\pi} = \tilde{\pi}'\,\mathrm{KT}(c)$.

C. Proof of Lemma 5

It is easy to see that it suffices to prove the lemma for the special case when s is a reduced form of t, because then we obtain the following: Let r be a reduced form of t. Since $t \sim s$, we know that r is also a reduced form of s. Hence, by the above, $\delta(t) \sim \delta(r)$ and $\delta(r) \sim \delta(s)$. By transitivity of $\sim$, this yields $\delta(t) \sim \delta(s)$.

So, let us assume that t is an exponent-reduced term and s is a reduced form of t. In the following, $u \downarrow_r w$ means that w is a reduced form of u. We proceed by induction on the size of t and consider the following cases:

1) The head symbol of t is neither $\cdot^{-1}$ nor $\uparrow$: Then the head symbol of s is the same as the one of t. If t is a constant or a variable (and so is s), then $\delta(t) = t = s = \delta(s)$. If t is of the form $f(t_1,\ldots,t_n)$, then s must also be of the form $f(s_1,\ldots,s_n)$ with $t_i \downarrow_r s_i$. By the induction hypothesis, we know that $\delta(t_i) \sim \delta(s_i)$. Hence, $\delta(t) = f(\delta(t_1),\ldots,\delta(t_n)) \sim f(\delta(s_1),\ldots,\delta(s_n)) = \delta(s)$.
2) The head symbol of t is $\cdot^{-1}$, i.e. $t = u^{-1}$: If u is of the form r^{-1}, then $r \downarrow_r s$. Hence, by the definition of δ and the induction hypothesis, $\delta(t) = \delta(r) \sim \delta(s)$. If u is not of the form r^{-1}, then s must be of the form w^{-1} with $u \downarrow_r w$. By the induction hypothesis, $\delta(u) \sim \delta(w)$ and, since $\delta(t) = \delta(u)$ and $\delta(s) = \delta(w)$, we have $\delta(t) \sim \delta(s)$.
3) The head symbol of t is $\uparrow$: Then we can write t as $t_0 \uparrow t_1 \uparrow \cdots \uparrow t_n$, where the head symbol of t_0 is not $\uparrow$. Let us consider two subcases:

   Case (a). There are $i,j \in \{1,\ldots,n\}$ such that $t_i \sim {t_j}^{-1}$. For simplicity of the presentation, we assume that $i = n-1$ and $j = n$ (it is easy to drop this assumption). Then, for $t' = t_0 \uparrow t_1 \uparrow \cdots \uparrow t_{n-2}$, we also have $t' \downarrow_r s$. Because t is exponent-reduced it follows that $t_i \doteq {t_j}^{-1}$ or ${t_i}^{-1} \doteq t_j$. We can conclude that $t_i \in \mathsf{C}^*$ if and only if $t_j \in \mathsf{C}^*$. Now, it is easy to see that $\delta(t) \sim \delta(t')$. By the induction hypothesis, we have $\delta(t') \sim \delta(s)$, which implies $\delta(t) \sim \delta(s)$.

   Case (b). There are no $i,j \in \{1,\ldots,n\}$ such that $t_i \sim {t_j}^{-1}$: Then, modulo (DH1), s must be of the form $s_0 \uparrow s_1 \uparrow \cdots \uparrow s_n$, where the head symbol of s_0 is not $\uparrow$, $t_0 \downarrow_r s_0$, and $t_i \doteq s_i$ for every $i \in \{1,\ldots,n\}$. The latter holds since by the definition of exponent-reduced, all t_i, for $i \in \{1,\ldots,n\}$, are reduced. Now, $\delta(t) = \delta(t_0) \uparrow t_{i_1} \uparrow \cdots \uparrow t_{i_k}$, where $t_{i_1},\ldots,t_{i_k}$ are exactly these elements of $t_1,\ldots,t_n$ which are in C^*. Similarly to Case (a), we have that $t_i \in \mathsf{C}^*$ if and only if $s_i \in \mathsf{C}^*$ for every $i \in \{1,\ldots,n\}$. Hence, $\delta(s) \doteq \delta(s_0) \uparrow s_{i_1} \uparrow \cdots \uparrow s_{i_k}$. By the induction hypothesis, we know that $\delta(t_0) \sim \delta(s_0)$. It follows that $\delta(t) \sim \delta(s)$.

D. Proof of Lemma 6

The proof is by induction on the structure of t. If t is a standard term, the statement easily follows by induction. We do not need to consider the case $t = s^{-1}$, since t is assumed to be C-exponent-ground. Let us assume that $t = s \uparrow s'$. Since t is C-exponent-ground, it follows that $s' \in \mathsf{C}^*$ and s is C-exponent-ground. Hence, $\delta(t\theta) = \delta(s\theta \uparrow s') = \delta(s\theta) \uparrow s'$. By the induction hypothesis, we have $\delta(s\theta) = s\delta(\theta)$. Thus, $\delta(s\theta) \uparrow s' = s\delta(\theta) \uparrow s' = t\delta(\theta)$.

E. Proof of Lemma 7

Assume that $t \sim s$. It follows that there exists a term r which is a reduced form of both t and s. Hence, we can obtain r from t (and from s) by applying to t (to s) equations (DH2) and (DH3) from left to right and equation (DH1) from left to right and from right to left, a number of times. More precisely, it is easy to see that (DH3) cannot be applied, since t and s are C-exponent-ground and because the above transformations with (DH2) and (DH1) preserve C-exponent-groundness. Moreover, it is not hard to verify that every such transformation with (DH2) and (DH1) preserves $\ulcorner\cdot\urcorner$, i.e., if u' is obtained by one such transformation from a C-exponent-ground term u, then $\ulcorner u'\urcorner = \ulcorner u\urcorner$. Now, it is easy to conclude that $\ulcorner t\urcorner = \ulcorner r\urcorner = \ulcorner s\urcorner$.

F. Proof of Lemma 10

The proof is by structural induction on t. In case t is standard, the statement follows immediately by the induction hypothesis. So, assume that t is non-standard. Let t' be a reduced form of t. Then, $t' \doteq t_0 \uparrow c_1^{(k_1)} \uparrow \cdots \uparrow c_m^{(k_m)}$, for some integers $k_1,\ldots,k_m$, and a C-exponent-ground term t_0. It is easy to see that $\ulcorner t'\urcorner = \mathsf{exp}(\ulcorner t_0\urcorner, i2t(k_1),\ldots,i2t(k_m))$.

By definition of $\llcorner\cdot\lrcorner$ and the fact that $t2i(i2t(k)) = k$, we obtain $\llcorner\ulcorner t'\urcorner\lrcorner = \llcorner\ulcorner t_0\urcorner\lrcorner \uparrow c^{(k_1)} \uparrow \cdots \uparrow c_m^{(k_m)}$. The induction hypothesis yields that $\llcorner\ulcorner t_0\urcorner\lrcorner \sim t_0$, and therefore, $\llcorner\ulcorner t'\urcorner\lrcorner \sim t_0 \uparrow c^{(k_1)} \uparrow \cdots \uparrow c_m^{(k_m)}$. Hence, $\llcorner\ulcorner t'\urcorner\lrcorner \sim t'$. Since $t \sim t'$, Lemma 7 implies that $\ulcorner t\urcorner = \ulcorner t'\urcorner$ and so $\llcorner\ulcorner t\urcorner\lrcorner = \llcorner\ulcorner t'\urcorner\lrcorner$. Consequently, we obtain $t \sim t' \sim \llcorner\ulcorner t\urcorner\lrcorner$.