A Unification Algorithm for Analysis of Protocols with Blinded Signatures

Deepak Kapur[1]*, Paliath Narendran[2]**, and Lida Wang[2]***

[1] Department of Computer Science, University of New Mexico, Albuquerque, NM 87131 USA, kapur@cs.unm.edu.

[2] Department of Computer Science, SUNY at Albany, Albany, NY 12222 USA, dran@cs.albany.edu, lidawang@cs.albany.edu.

Abstract. Analysis of authentication cryptographic protocols, particularly finding flaws in them and determining a sequence of actions that an intruder can take to gain access to the information which a given protocol purports not to reveal, has recently received considerable attention. One effective way of detecting flaws is to hypothesize an insecure state and determine whether it is possible to get to that state by a legal sequence of actions permitted by the protocol from some legal initial state which captures the knowledge of the principals and the assumptions made about an intruder's behavior. Relations among encryption and decryption functions as well as properties of number theoretic functions used in encryption and decryption can be specified as rewrite rules. This, for example, is the approach used by the NRL Protocol Analyzer, which uses narrowing to reason about such properties of cryptographic and number-theoretic functions.

Following [14], a related approach is proposed here in which equation solving modulo most of these properties of cryptographic and number-theoretic functions is done by developing new unification algorithms for such theories. A new unification algorithm for an equational theory needed to reason about protocols that use the Diffie-Hellman algorithm is developed. In this theory, multiplication forms an Abelian group; exponentiation function distributes over multiplication, and exponents can commute. This theory is useful for analyzing protocols which use blinded signatures. It is proved that the unification problem over this equational theory can be reduced to the unification problem modulo the theory of Abelian groups with commuting homomorphisms with an additional constraint. Baader's unification algorithm for the theory of Abelian groups with commuting homomorphisms, which reduces the unification problem to solving equations over the polynomial ring over the integers with the commuting homomorphisms serving as indeterminates, is generalized to give a unification algorithm over the theory of Abelian groups with commuting homomorphism with a *linear constraint*.

It is also shown that the unification problem over a (simple) extension of the equational theory considered here (which is also an extension of the equational theory considered in [14]) is undecidable.

* Research supported in part by the NSF grant nos. CCR-0098114 and CDA-9503064, the ONR grant no. N00014-01-1-0429, and a grant from the Computer Science Research Institute at Sandia National Labs.

** Research supported in part by NSF grant no. CCR-0098095 and ONR grant no. N00014-01-1-0430.

*** Research supported in part by NSF grant no. CCR-0098095.

1 Introduction

Search techniques for exploring the vast state space possibly generated by a given authentication cryptographic protocol have turned out to be an effective way to determine possible flaws in protocols. A typical state is the knowledge possessed by each of the principals interested in communicating among each other in a secure fashion, time clock often needed for generating nonces and/or timestamps, and most importantly, an intruder who can read, alter and delete traffic, send messages of its own, pretend to be any of the principals and who may have help from the principals, etc. Actions taken by principals and an intruder(s) lead to state changes. At the same time, certain relations between various cryptographic functions as well as properties of number theoretic functions used for public encryption and decryption must be honored by the states. For instance in a symmetric key system, it is common to introduce a rule saying that, if a principal knows a message M encrypted with a key K, and also knows the key K, then it can learn message M. In a public key based cryptosystem, encryption and decryption can be expressed in terms of two functions symbols e and d that obey the following identities: $e(Kpu, d(Kpr, M)) \rightarrow M$ and $d(Kpr, e(Kpu, M)) \rightarrow M$ where Kpu, Kpr are respectively the public and private keys for a principal.

A technical report by Clark and Jacob [7] is a comprehensive survey of authentication protocols. That report reviews numerous protocols proposed in the literature; its annotated bibliography also discusses various attacks on some of these protocols and how they have been fixed. The report briefly mentions various approaches for establishing correctness of authentication protocols. An interested reader should consult [7] for details. Below, we discuss a state-based approach for analyzing possible attacks on an authentication protocol; see [13, 12] for more details.

A typical scenario for analyzing a protocol is to hypothesize an insecure state in which the protocol is compromised possibly by an intruder knowing some information that the protocol professes not to reveal, and to work *backwards* to determine whether this state is reachable, by a sequence of actions of the principals and the intruder, from a given legal state. This, for example, is the approach used by the NRL Protocol Analyzer (NPA) [13], a software tool for cryptographic protocol analysis implemented in Prolog. NPA exploits the backtracking facility in Prolog in combination with equational unification for exploring the state space. It makes use of narrowing, a general purpose technique for equation solving, to identify states which are equivalent because of properties of encryption, decryption functions as well as properties of number theoretic functions used in authentication, encryption and decryption. For instance, most public key based cryptosystems use multiplication, exponentiation, and modulus operations on numbers. Relations are specified by terminating rewrite rules. In certain cases, NPA is even able to rule out the reachability of such insecure states.

(Narrowing is a general purpose technique for equation solving with respect to a given rewrite system; it thus requires that every property be oriented into a terminating rewrite rule. See [9], for instance.)

Although narrowing works well for certain properties relating encryption and decryption, unfortunately there are a number of other properties of cryptographic operations which cannot be handled. Certain relations, such as associativity and commutativity properties of arithmetic operations $+$, $*$, etc, cannot be oriented into terminating rewrite rules. Furthermore, narrowing is a general purpose method, whereas special purpose unification algorithms for certain theories capturing relations among cryptographic functions could perhaps be more efficient.

An approach for integrating unification algorithms for equational theories axiomatizing properties of cryptosystems including encryption, decryption and primitive number theoretic functions implementing them, is outlined in a recent paper by Meadows and Narendran [14]. The basic idea is to use a unification algorithm for an equational theory in place of narrowing using rewrite rules in the equational theory. A unification algorithm for an equational theory found useful in analyzing group Diffie-Hellman protocols was proposed in the paper and the complexity of the unifiability check for this equational theory was shown to be NP-complete.

This paper builds on [14]. The equational theory under consideration is assumed to be weaker than the theory considered in [14] with respect to the properties of the exponentiation function on numbers. in particular, the axiom $x^{y^z}, = x^{y \cdot z}$ relating exponentiation with multiplication is dropped. However, the exponents are assumed to commute, i.e., $x^{y^z} = x^{z^y}$ holds[1]. This theory is useful in cryptographic techniques such as Chaum's blinded signature [6], popular in anonymous electronic cash schemes, which also make direct use of properties such as distributivity of modular exponentiation over modular multiplication (i.e., $(x \cdot y)^z = (x^z) \cdot (y^z)$).

It is proved in this paper that the above distributivity axiom cannot be added to the theory considered in [14] without making the unification problem on the extended theory undecidable. The undecidability proof

[1] It is easy to see that this property follows from the axiom $x^{y^z} = x^{y \cdot z}$ because of commutativity of $\cdot$.

of the unification problem over the equational theory of *Cartesian Closed Categories* (CCC) considered in [15] can be adapted to be applicable for this theory.

A decision procedure for the unification problem for the equational theory of Abelian group for multiplication along with the distributivity axiom and the exponent commutativity axiom (and the properties of the unit 1) is also given. In this sense, the theory considered in this paper and the theory considered in [14] are two proper subsets with decidable unification problems of an equational theory with an undecidable unification problem.

The remainder of this paper is organized as follows. Section 2 is based on [14], providing a brief review of the Diffie-Hellman and group Diffie-Hellman algorithms, and discussing axioms relating number-theoretic properties of multiplication and exponentiation used. It is included for the sake of completeness; for details, an interested reader may consult [14].

Section 3 shows the undecidability of the equational theory considered in [14] along with the distributivity axiom of exponentiation over multiplication. The proof is essentially based on the undecidability proof given in [15] where Hilbert's tenth problem over natural numbers is shown to be an instance of the unification problem over the equational theory of cartesian closed categories. The main difference is that in the case below, it is possible to simulate negative numbers as well. Consequently, Hilbert's tenth problem over the integers is reduced to this unification problem.

Section 4 develops the necessary background to relate the unification problem over the equational theory of blinded signatures to the unification problem over the equational theory of Abelian groups with n commuting homomorphisms, called $AGnHC$ by Baader [2], with an additional condition, called a *linear constraint*.

Section 5 shows how Baader's algorithm for the unification problem over $AGnHC$ can be generalized so as to satisfy a linear constraint (which is similar to the linear constant restrictions discussed in [1]). A new way of defining admissible term orderings is introduced, which is used to compute a Gröbner basis of a polynomial ideal.

Section 6 concludes with some remarks on complexity of the algorithm and outlines some areas for further research.

2 Protocols based on Diffie-Hellman Algorithm

This section is borrowed from [14] where the motivation for this approach is outlined.

The Diffie-Hellman algorithm in its most basic form allows two principals to securely exchange a secret key without having any shared secret beforehand. Consider a prime number P and a generator x of the multiplicative group of $\mathbf{Z}_P$. Principal A generates a secret value N_A, and B generates a secret value N_B. The protocol then runs as follows:

1. $A \rightarrow B \ : \ x^{N_A} \bmod P$
   B then computes $(x^{N_A})^{N_B}$.
2. $B \rightarrow A \ : \ x^{N_B} \bmod P$
   A then computes $(x^{N_B})^{N_A}$, which is the same as $(x^{N_B})^{N_A}$, implying that the shared key is the same among the principals A and B.

In order to secure the above protocol against active eavesdroppers, it is necessary to include some form of authentication, usually provided by public-key signatures. Thus an equational theory that could be used to reason about Diffie-Hellman would need to take into account the relationship between exponentiation and $\cdot$, the commutativity of $\cdot$, and possibly identities obeyed by the signature algorithms used.

Interaction between exponentiation and $\cdot$ is captured by the distributivity axiom: $(x \cdot y)^z = (x^z) \cdot (y^z)$.

2.1 Some Equational Theories for Diffie-Hellman and Group Diffie-Hellman

Equational theories under consideration consist of the axioms of Abelian groups (A, C, U and Inv below), denoted by AG, along with (some) axioms for exponentiation. In contrast to the equational theory considered in [14], the axiom, $(x^y)^z = x^{y \cdot z}$, used in [14] is excluded; instead, axioms (Exp2, Exp3, Exp4) are used. The relationship between exp and $\cdot$ is captured by axiom Exp3.

Consider the following axioms for $\cdot$ and exponentiation (denoted by x^y):

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z \tag{A}$$
$$x \cdot y = y \cdot x \tag{C}$$
$$x \cdot 1 = x \tag{U}$$
$$x \cdot x^{-1} = 1 \tag{Inv}$$
$$x^1 = x \tag{Exp1}$$
$$1^x = 1 \tag{Exp2}$$
$$(x \cdot y)^z = (x^z) \cdot (y^z) \tag{Exp3}$$
$$x^{y^z} = x^{z^y} \tag{Exp4}$$
$$(x^y)^z = x^{y \cdot z} \tag{Exp5}$$

The first four axioms characterize Abelian groups; this theory is denoted by AG. The remaining axioms are about exponentiation as well as interactions between exponentiation and $\cdot$ (particularly, axioms (Exp4) and (Exp5)).

As stated earlier, [14] gave a unification algorithm for the equational theory consisting of AG and axioms (Exp1, Exp2, Exp5); it was proved that the unifiability check is NP-complete. It is easy to see that (Exp4) follows from AG + (Exp5).

It is shown in the next section that the unifiability check for AG and axioms (Exp1, Exp2, Exp3) and (Exp5) is undecidable. The rest of the paper focuses on the decidability of the unifiability check for AG plus axioms {Exp1, Exp2, Exp3, Exp4}.

3 Undecidability of Unifiability Check for the theory of AG and Exponentiation

Consider the equational theory $\mathcal{E}$ consisting of AG and axioms (Exp1, Exp2, Exp3 and Exp5). This theory has the following convergent rewrite system modulo associativity and commutativity (also called AC-convergent).

$$\begin{aligned}
(x^{-1})^{-1} &\rightarrow x \\
1^{-1} &\rightarrow 1 \\
x \ \cdot \ 1 &\rightarrow x \\
x \ \cdot \ x^{-1} &\rightarrow 1 \\
x \ \cdot \ (x^{-1} \ \cdot \ y) &\rightarrow y \\
(x \cdot y)^{-1} &\rightarrow x^{-1} \ \cdot \ y^{-1} \\
1^z &\rightarrow 1 \\
z^1 &\rightarrow z \\
(x^{-1})^y &\rightarrow (x^y)^{-1} \\
(x \cdot y)^z &\rightarrow (x^z) \cdot (y^z) \\
(x^y)^z &\rightarrow x^{(y \cdot z)}
\end{aligned}$$

It is easy to see that the normal form of a term using the above rewrite rules is either 1 or of the form $t_1 \cdot t_2 \cdot \ldots \cdot t_n$, where each t_i is x, x^{-1}, x^{e_i}, or $(x^{e_i})^{-1}$ for some variable or constant x, where e_i is a term in normal form wrt the above AC-convergent rewrite system that is different from 1. In what follows, a term of the form b^i where i is a positive integer, is an abbreviation of $\underbrace{b \cdot b \cdot \ldots \cdot b}_{i}$. If i is a negative integer then b^i is an abbreviation of $\underbrace{b^{-1} \cdot b^{-1} \cdot \ldots \cdot b^{-1}}_{(-i)}$. ($b^0 = 1$.)

Lemma 1. *Let $t_1, \ldots, t_n$ be terms in normal form modulo the above AC-convergent system, such that none of the t_i's has $\cdot$ as the outermost symbol. If s is the normal form of $t = t_1 \cdot \ldots \cdot t_n$, then s is either 1 or s can be written as $s_1 \cdot \ldots \cdot s_k$ where $\{s_1, \ldots, s_k\} \subseteq \{t_1, \ldots, t_n\}$ (in the multiset sense).*

Proof: If t is already in normal form, then the lemma holds. If for each $i\,(1 \leq i \leq n)$ $t_i = 1$, then s is 1. Otherwise since $t_i\,(1 \leq i \leq n)$ is in normal form and none of the t_is has $\cdot$ as the outermost symbol, the only possible reduction rules that can apply to t are:

$$\begin{aligned} x \ \cdot \ 1 &\rightarrow x \\ x \ \cdot \ x^{-1} &\rightarrow 1 \\ x \ \cdot \ (x^{-1} \ \cdot \ y) &\rightarrow y \end{aligned}$$

Each time any of these these rules is applied, some t_i will be gotten rid of. By induction on the number of reduction steps, we can complete the proof. □

Lemma 2. *For all i, the $\mathcal{E}$-unification problem*

$$x \cdot a^y =^? x^b \cdot a^{b^i}$$

is solvable iff $y \leftarrow b^m$ for some integer m.

Proof: The "if" part is straightforward. It is clear that for every $n \geq i$, $\{y \leftarrow b^n, x \leftarrow a^{b^{n-1}} \cdot a^{b^{n-2}} \cdot \cdots \cdot a^{b^{i+1}} \cdot a^{b^i}\}$ is indeed a solution. We prove the "only if" part by contradiction. Suppose the claim is not true, i.e., there is a solution with a y that is not of the form b^j where j is an integer.

Let $S = \{\ x \mid x$ is in normal form and there are y and i such that $a^y \cdot x =_{\mathcal{E}} x^b \cdot a^{b^i}$, i is an integer, y is not of the form b^j where j is an integer $\}$

Let x be the smallest term in the above set. Since x is in normal form, it must be that $x^b.a^{b^i}.i(a^y)$ reduces to x. Now two cases have to be considered:

(i) a^{b^i} is a part (factor) of x, i.e., $x =_{\mathcal{AC}(\cdot)} a^{b^i} \cdot z$ for some z. Then

$$a^y \cdot a^{b^i} \cdot z =_{\mathcal{E}} z^b \cdot a^{b^{i+1}} \cdot a^{b^i}$$

Canceling on both sides, we find that $z \in S$ and z is smaller than x.

(ii) the normal form of x^b contains $i(a^{b^i})$. Since x is in normal form, x must contain $i(a^{b^{i-1}})$, i.e., $x =_{\mathcal{AC}(\cdot)} i(a^{b^{i-1}}).z$. Then

$$a^y \cdot i(a^{b^{i-1}}) \cdot z =_{\mathcal{E}} z^b$$

Again, z is smaller than x, which is a contradiction.

□

Lemma 3. *Let b and c be free constants and j be an integer. Then the $\mathcal{E}$-unification problem*

$$x^c \cdot a^{b^j} =^? x^b \cdot a^u$$
$$z \cdot a^u =^? z^c \cdot a$$

is unifiable iff $u \leftarrow c^j$. (In other words, $x^c \cdot a^{b^j} =^? x^b \cdot a^{c^k}$ is unifiable if and only if $j = k$.)

Proof: By Lemma 2, the second equation is unifiable iff $u \leftarrow c^k$ where k is an integer. Suppose the equation $x^c \cdot a^{b^j} =^? x^b \cdot a^{c^k}$ is solvable. Replace b with c everywhere in the equation. Since b and c are free constants, it must be that $y^c \cdot a^{c^j} =_{\mathcal{E}} y^c \cdot a^{c^k}$ where y is x with b replaced everywhere with c. Now the cancellation properties can be applied to get the result. □

Lemma 4. *Let b and c be free constants and j and k be integers. Then $\mathcal{E}$-unification problem*

$$x^{c^k} \cdot a^{b^j} =^? x^b \cdot a^u$$
$$z \cdot a^u =^? z^c \cdot a$$

is unifiable iff $u \leftarrow c^{j*k}$.

Proof: By Lemma 2, the second equation is unifiable iff $u \leftarrow c^n$ where n is an integer. Suppose the equation $x^{c^k} \cdot a^{b^j} =^? x^b \cdot a^{c^n}$ is solvable. Replace b with c^k everywhere in the equation. Since b and c are free constants, it must be that $y^{c^k} \cdot a^{c^{j*k}} =_{\mathcal{E}} y^{c^k} \cdot a^{c^n}$ where y is x with b replaced everywhere with c^k. Now the cancellation properties can be applied to get the result.□

Lemma 4 shows that multiplication of two integers can be simulated. Consider, for instance, the equation $z = x * y$. If $x = b^i$ and $y = b^j$, then we can force z to be b^{ij} in the following way:

(i) Copy x to x' changing b's to c's; i.e., $x' = c^i$. This can be done using equations as given in the statement of Lemma 4.
(ii) Multiply x' and y to get $z' = c^{ij}$.
(iii) Copy z' to z changing c's to b's.

Thus the equations we get are

$$\begin{aligned} w_1 \cdot a^{x'} &=^? {w_1}^c \cdot a \\ {w_2}^c \cdot a^x &=^? {w_2}^b \cdot a^{x'} \\ {w_3}^{x'} \cdot a^y &=^? {w_3}^b \cdot a^{z'} \\ w_4 \cdot a^{z'} &=^? {w_4}^c \cdot a \\ {w_5}^c \cdot a^z &=^? {w_5}^b \cdot a^{z'} \end{aligned}$$

Simulating addition is easy, since if $x = b^i$ and $y = b^j$, then $(a^x)^y = a^{b^{i+j}}$.

Now we are ready to prove the undecidability of the equational theory $\mathcal{E}$ by reduction from Hilbert's tenth problem:

Theorem 1. *The unifiability check for the equational theory $\mathcal{E}$ is undecidable.*

Proof: Given a system of diophantine equations, we can construct a unification problem modulo $\mathcal{E}$ as outlined above. □

4 Unification over Equational Theory of Blinded Signatures

In this section, we consider a proper subset of the above equational theory. In particular, the axiom (Exp5) is replaced by a weaker axiom (Exp4). Let $\mathcal{E}_0$ consist of AG and axioms (Exp1, Exp2, Exp3, Exp4) denoted by *Exp*.

The theory $\mathcal{E}_0 \setminus \text{Exp4}$, denoted by $\mathcal{E}_0$', has the following AC-convergent rewrite system.

$$\begin{aligned}
(x^{-1})^{-1} &\to x \\
1^{-1} &\to 1 \\
x \ \cdot \ 1 &\to x \\
x \ \cdot \ x^{-1} &\to 1 \\
(x \cdot y)^{-1} &\to x^{-1} \ \cdot \ y^{-1} \\
1^{z} &\to 1 \\
z^{1} &\to z \\
(x^{-1})^{y} &\to (x^{y})^{-1} \\
(x \cdot y)^{z} &\to (x^{z}) \cdot (y^{z})
\end{aligned}$$

Exp4 cannot be oriented into a terminating rewrite rule.

In the next sections, we show that the equational unification problem for $\mathcal{E}_0$ is equivalent to the unification problem modulo the theory of Abelian groups with *n commuting* homomorphisms, denoted by $AGnHC$, but with an additional constraint. The theory $AGnHC$ is a *monoidal* theory; further, it is shown in [16, 3] that $AGnHC$ is *unitary* with respect to unification without constants and it is also unitary with respect to unification with constants[2]. In Section 5 of [2], Baader showed that the unification problem of $AGnHC$ reduces to solving linear equations over the polynomial ring $\mathsf{Z}[h_1, \ldots, h_n]$, where h_1, ... , h_n are the commuting homomorphisms of $AGnHC$, treated as indeterminates in the polynomial ring. We generalize Baader's algorithm by adding an additional key step to ensure that a given *linear constraint* is satisfied by the unifier, so as to apply it to the equational unification problem for $\mathcal{E}_0$. This is discussed in this section and the next section.

4.1 Unification over $\mathcal{E}_0$ as a Combination of Theories

Consider a set S_0 of equations whose unifiability needs to be checked wrt $\mathcal{E}_0$. Assuming S_0 is unifiable, given any unifier θ of S_0, θ may substitute 1 for certain variables in S_0; also many variables in S_0 may get identical substitutions. Given that there are only finitely many variables in S_0,

[2] A theory is unitary if a minimal complete set of unifiers always exists and its cardinality is at most one.

there are only finitely many such partial unifiers for S_0 in which some of the variables in S_0 get either 1 or another variable as a substitution. After applying such a partial unifier on S_0, simplifying the result by the rewrite rules of the associated AC-convergent system, and deleting trivial equations (i.e., equations which are in the equational theory of $\mathcal{E}_0$), we get a set S_1 of equations. If S_1 is empty, then the above partial unifier is a unifier of S_0. If S_1 is not empty, then the following steps are applied. We will assume that the equations in S_1 have been normalized using the AC-convergent rewrite system for $\mathcal{E}'_0$ $(= \mathcal{E}_0 - \text{Exp4})$ discussed above. Thus, a unifier of S_1 cannot substitute for any variable x, a normalized term t properly containing x (occur-check).

Any unification problem S_1 over $\mathcal{E}_0$ can be simplified using variable abstraction (by introducing new symbols) to a *simple* $\mathcal{E}_0$-unification problem, say S_2; this is defined precisely below.

Definition 1. *An $\mathcal{E}_0$-unification problem S over Σ is called an* AG-*unification problem if each equation in S is of the form $x =^? t$, where x is a variable and t is a term over the signature of AG such that $t \neq_{AG} 1$.*

Definition 2. *An $\mathcal{E}_0$-unification problem S on Σ is called an* exponent *$\mathcal{E}_0$-unification problem if every equation in S is of the form $x =^? y^z$ where x and y are variables and z is a variable or a free constant. Also if z is a variable, z is called an* exponent variable, *otherwise it is called an* exponent constant.

Definition 3. *An $\mathcal{E}_0$-unification problem S on Σ is called a* simple *$\mathcal{E}_0$-unification problem if $S = S_1 \bigcup S_2$ where S_1 is an AG-unification problem and S_2 is an exponent $\mathcal{E}_0$-unification problem.*

Let $Var(S)$ denote the set of all variables in S.

It is easy to see that using abstraction, any $\mathcal{E}_0$-unification problem can be transformed into a simple $\mathcal{E}_0$-unification problem. For example, consider $S_1 = \{w =^? (x^{(y^{u \cdot v})^{-1} . z^{u' \cdot v'}})^{-1}\}$. Using abstractions, the above equation in S_1 is transformed to S_2:

$$\begin{aligned}\{&1.\ w =^? z_1^{-1}, \quad 2.\ z_1 =^? x^{z_2}, \quad 3.\ z_2 =^? z_3^{-1} \cdot z_4, \quad 4.\ z_3 =^? y^{z_5},\\ &5.\ z_5 =^? u \cdot v, \quad 6.\ z_4 =^? z^{z_6}, \quad 7.\ z_6 =^? u' \cdot v'\},\end{aligned}$$

where $z_1, z_2, z_3, z_4, z_5, z_6$ are new variables introduced to abstract alien subterms in S_1.

4.2 Relating $AG + EXP$ to $AGnHC$

Given a simple $\mathcal{E}_0$-unification problem S_2 on $\Sigma = \{\cdot, ^{-1}, 1, x^y\}$, for each equation of the form $x = y^\beta$ in S_2, we transform it into $x = h_\beta(y)$, where h_β is a homomorphism corresponding to the symbol β. Let $\mathcal{H}(S_2)$ denote the set of all homomorphisms introduced in this way. Let $\Sigma' = \{\cdot, ^{-1}, 1\} \bigcup \mathcal{H}(S_2)$, $\mathcal{E}' = AG \bigcup \{h_1(h_2(u)) = h_2(h_1(u)),\ h(u_1 \cdot u_2) = h(u_1) \cdot h(u_2)\}$ for all $h, h_1, h_2 \in \mathcal{H}(S_2)$. We call the transformed $\mathcal{E}'$-unification problem on Σ' as an *h-image* of S_2.

For the above example, its *h-image* T_2 is:

$$\{1.\ w =^? z_1^{-1}, \quad 2.\ z_1 =^? h_{z_2}(x), \quad 3.\ z_2 =^? z_3^{-1} \cdot z_4, \quad 4.\ z_3 =^? h_{z_5}(y),$$
$$5.\ z_5 =^? u \cdot v, \quad 6.\ z_4 =^? h_{z_6}(z), \quad 7.\ z_6 =^? u' \cdot v'\}.$$

The requirement that a normalized unifier for S_2 wrt $\mathcal{E}_0$ satisfy the occur-check for every variable x translates to a related requirement in $\mathcal{E}'$. A normalized unifier of the h-image T_2 of S_2 wrt $AGnHC$ should satisfy (i) the occur-check for every variable x, and in addition, (ii) the substitution for x must not properly include h_x, the homomorphism introduced for x when x appears as an exponent.

In order to show the equivalence of the unifiability check over $\mathcal{E}_0$ with the unifiability check over $AGnHC$, it is necessary to place restrictions on unifiers considered for $\mathcal{E}_0$ given that we have considered a priori equivalent substitutions for variables as well as 1 as the substitution for variables. This is done by solving the unifiability problem of T_2 wrt $AGnHC$ subject to *linear* constraints (including) $x \succ h_x$ for every homomorphism $h_x \in \mathcal{H}(S_2)$, i.e., a unifier θ of T_2 should satisfy the condition that for every $x \in Var(T_2)$, $\theta(x)$ *does not contain* any occurrence of h_x.

Definition 4. *Given a simple $\mathcal{E}_0$-unification problem S and its h-image T modulo $AGnHC$, a* linear constraint *is a total order $\succ$ over $Var(T) \cup \mathcal{H}(S)$ such that $x \succ_C h_x$ for all exponent variables x in S.*

Definition 5. *A substitution β whose domain is $Var(T)$ is said to satisfy a linear constraint C if and only if the following holds: for every $x \in Var(T)$, $\beta(x)$* does not contain *any of the function symbols below x in C. In other words, if $x \succ_C h_y$, then $\beta(x)$ does not contain any occurrence of h_y.*

Definition 6. *A unifier θ for a unification problem S is said to be a* discriminating *unifier if and only if the following hold for all variables in $Var(S)$:*

1. $\theta(u) \neq_{\mathcal{E}} 1$ *for all* u.
2. $\theta(v) =_{\mathcal{E}} \theta(w)$ *iff* $v = w$.

The following two theorems relate the unification problem S_2 over $\mathcal{E}_0$ to its h-image T_2 over $AGnHC$.

Theorem 2. *If a simple* $\mathcal{E}_0$*-unification problem* S_2 *has a discriminating unifier, then its h-image* T_2 *which is a* $\mathcal{E}'$*-unification problem on* Σ' *is unifiable. Furthermore, there is a linear constraint* C *that the unifier satisfies.*

Proof: Let θ be a discriminating unifier of a simple $\mathcal{E}_0$-unification problem S_2. Consider all the exponent equations in S_2 :

$$\begin{array}{ll} \{\ x_{u_1} =^? & {x_{v_1}}^{x_{w_1}} \\ \quad\vdots & \quad\vdots \\ \ \ x_{u_i} =^? & {x_{v_i}}^{x_{w_i}} \\ \quad\vdots & \quad\vdots \\ \ \ x_{u_k} =^? & {x_{v_k}}^{x_{w_k}}\} \end{array}$$

The unifier θ includes $\{x_{u_i} \leftarrow t_{u_i}, x_{v_i} \leftarrow t_{v_i}, x_{w_i} \leftarrow t_{w_i} (1 \leq i \leq k)\}$. Thus, $t_{u_i} =_{\mathcal{E}_0} {t_{v_i}}^{t_{w_i}} (1 \leq i \leq k)$.

For each t_{w_i}, we introduce a homomorphism $h_{t_{w_i}}$. Let $\mathcal{H}''(S)$ denote the set of homomorphisms introduced for all t_{w_i}. Let $\Sigma'' = \{\cdot, ^{-1}\} \bigcup \mathcal{H}''(S)$. Let $\mathcal{E}'' = AG \bigcup \{h_{t_{w_i}}(h_{t_{w_j}}(u)) = h_{t_{w_j}}(h_{t_{w_i}}(u)),\ h_{t_{w_i}}(u_1 \cdot u_2) = h_{t_{w_i}}(u_1) \cdot h_{t_{w_i}}(u_2)\}$ for all i $(1 \leq i \leq k)$. We also define a recursive function ***rep*** as follows:

$\boldsymbol{rep}(a) = a$ where a is a constant in Σ.
$\boldsymbol{rep}(A \cdot B) = \boldsymbol{rep}(A) \cdot \boldsymbol{rep}(B)$ where A, B are terms on Σ.
$\boldsymbol{rep}(A^x) = h_x(\boldsymbol{rep}(A))$ where x, A are terms on Σ.

It is easy to see that the function ***rep*** removes all occurrences of the exponent operator from terms over Σ. Since θ is a discriminating unifier for S_2, for each $t_{w_i}, t_{w_j} (i \neq j)$, we have $t_{w_i} \neq_{\mathcal{E}_0} t_{w_j}$. Also it is easy to show that $s =_{\mathcal{E}_0} t$ if and only if $\boldsymbol{rep}(s) =_{\mathcal{E}'} \boldsymbol{rep}(t)$ for any terms s, t over Σ. Therefore we should have:

$$\boldsymbol{rep}(t_{u_i}) =_{\mathcal{E}''} \boldsymbol{rep}({t_{v_i}}^{t_{w_i}}) =_{\mathcal{E}''} h_{t_{w_i}}(\boldsymbol{rep}(t_{v_i}))\ (1 \leq i \leq k) \text{ ——(1)}.$$

Now for each $h_{t_{w_i}} (1 \leq i \leq k)$, we introduce the homomorphism $h_{x_{w_i}}$ which is the same homomorphism we introduced for x_{w_i} in the *h-image*

T_2 of S_2. We also define function $\boldsymbol{rep'}$ as:

$$\begin{array}{l}
\boldsymbol{rep'}(a) = a \text{ where } a \text{ is a constant on } \Sigma'.\\
\boldsymbol{rep'}(A \cdot B) = \boldsymbol{rep'}(A) \cdot \boldsymbol{rep'}(B) \text{ where } A, B \text{ are terms on } \Sigma'.\\
\boldsymbol{rep'}(h_{t_{w_i}}(A)) = h_{x_{w_i}}(\boldsymbol{rep'}(A)) \text{ where } A \text{ is a term on } \Sigma.
\end{array}$$

Obviously $\boldsymbol{rep'}$ maps all $h_{t_{w_i}} (1 \leq i \leq k)$ to the corresponding $h_{x_{w_i}}$. So from (1) and the definition of $\mathcal{E}'$,

$$\boldsymbol{rep'}(\boldsymbol{rep}(t_{u_i})) =_{\mathcal{E}'} h_{x_{w_i}}(\boldsymbol{rep'}(\boldsymbol{rep}(t_{v_i}))).$$

That means T_2 is solvable and a unifier β for T_2 can be constructed as follows: $\beta(x) \leftarrow \boldsymbol{rep'}(\boldsymbol{rep}(\theta(x)))$ for every $x \in Var(S)$.

A linear constraint that β satisfies is constructed by comparing $\theta(x_i)$ for variables in $Var(T_2)$ using a simplification AC term ordering that is total on ground terms (e.g. [11]).[3]

□

Theorem 3. *Given a simple $\mathcal{E}_0$-unification problem S_2 on Σ and its h-image T_2 which is a $\mathcal{E}'$-unification problem on Σ', if T_2 has a solution which satisfies a linear constraint, then S_2 is solvable.*

Proof: Consider in S_2, all exponent equations

$$\begin{array}{lll}
\{ & x_{u_1} =^? & {x_{v_1}}^{x_{w_1}} \\
 & \vdots & \vdots \\
 & x_{u_i} =^? & {x_{v_i}}^{x_{w_i}} \\
 & \vdots & \vdots \\
 & x_{u_k} =^? & {x_{v_k}}^{x_{w_k}} \ \}.
\end{array}$$

The h-image T_2 for S_2 includes:

[3] Given a total AC-simplification ordering on ground terms $>$, add a new constant, say $\perp$, smaller than every other symbol. Now order the terms in the set

$$\{\theta(x_1), ..., \theta(x_n), \perp^{\theta(x_1)}, ..., \perp^{\theta(x_n)}\}$$

using $>$. All these terms will be distinct because we are considering a discriminating unifier. Note also that any term that contains $\theta(x_i)$ as an exponent is $> \perp^{\theta(x_i)}$. Replacing the $\theta(x_i)$'s by the (corresponding) x_i and replacing the $\perp^{\theta(x_i)}$'s by the corresponding h_{x_i}, we get a linear chain. Reversing the order gives C.

$$\begin{array}{rcl}\{\ x_{u_1} & =^? & h_{x_{w_1}}(x_{v_1}) \\ \vdots & & \vdots \\ x_{u_i} & =^? & h_{x_{w_i}}(x_{v_i}) \\ \vdots & & \vdots \\ x_{u_k} & =^? & h_{x_{w_k}}(x_{v_k})\}.\end{array}$$

Let β be a ground unifier of T_2 which satisfies a linear constraint C. From C, we can get a subconstraint C' on variables in $Var(T_2)$. Assume without loss of generality that $C' = x_n \succ \cdots \succ x_i \succ \cdots \succ x_1$. Now we will use induction on C' to form θ.

Let us first consider the first variable x_n in C'. Since x_n is the first variable, and $\beta(x_n)$ should not contain any item below x_n in C, it must be that $\beta(x_n)$ is composed of constants, and we define $\theta(x_n) = \beta(x_n)$.

Assume that we have already constructed all $\theta(x_{j'})$ $(j \leq j' \leq n)$. For variable x_{j-1}, the following cases arise:

1. $\beta(x_{j-1})$ is composed of constants. In this case, we define $\theta(x_{j-1}) = \beta(x_{j-1})$.
2. $\beta(x_{j-1})$ is composed of constants and some $h_{x_{w_i}}$ $(1 \leq i \leq k)$ where each $h_{x_{w_i}} \succ x_{j-1}$. Since $x_{w_i} \succ h_{x_{w_i}}$, we have $x_{w_i} \succ h_{x_{w_i}} \succ x_{j-1}$. By the induction hypothesis, we already know all these $\theta(x_{w_i})$. Therefore, we can define $\theta(x_{j-1}) = \boldsymbol{repp}(\beta(x_{j-1}))$ where the function $\boldsymbol{repp}$ is defined as:

$$\begin{array}{lcl}\boldsymbol{repp}(a) & = & a \text{ where } a \text{ is a constant on } \Sigma'. \\ \boldsymbol{repp}(A \cdot B) & = & \boldsymbol{repp}(A) \cdot \boldsymbol{repp}(B) \text{ where } A, B \text{ are terms on } \Sigma'. \\ \boldsymbol{repp}(h_{x_{w_i}}(A)) & = & \boldsymbol{repp}(A)^{\theta(x_{w_i})} \text{ where } A \text{ is a term on } \Sigma'.\end{array}$$

It can be shown that θ is a solution for S_2. Consider each exponent equation in T_2: $x_{u_i} =^? h_{x_{w_i}}(x_{v_i})$. Since $\beta(x_{u_i}) =_{\mathcal{E}'} h_{x_{w_i}}(\beta(x_{v_i}))$, we have $\theta(x_{u_i}) =_{\mathcal{E}_0} (\theta(x_{v_i}))^{\theta(x_{w_i})}$ by our definition of θ. □

In the next section, we show how Baader's algorithm for unifiability check for $AGnHC$ can be generalized to work with a linear constraint. This generalization is then used to solve the unifiability check over $\mathcal{E}$', and hence $\mathcal{E}_0$.

5 Unification over *AGnHC* with a Linear Constraint

In [2], Baader showed that the unifiers of a unification problem wrt *AGnHC*, where $h_1, \ldots, h_k$ are the commuting homomorphisms, correspond to the solutions of (nonhomogeneous) linear equations over the polynomial ring $Z[h_1, ..., h_k]$ with $h_1, \ldots, h_k$ as indeterminates in the polynomial ring. Let $NHE =$

$$\begin{aligned}\{p_{11}X_1 + \cdots + p_{1n}X_n &= p_1, \\ \vdots \qquad\qquad & \quad \vdots \\ p_{m1}X_1 + \cdots + p_{mn}X_n &= p_m\}\end{aligned}$$

be a set of linear equations where $p_{11}, \cdots, p_{1n}, \cdots, p_{m1}, \cdots, p_{mn}, p_1, \cdots, p_m$ are in $Z[h_1, ..., h_k]$.

Baader [2] gave an algorithm for solving such nonhomogeneous linear equations by first computing a *syzygy basis* for homogeneous linear equations over $Z[h_1, ..., h_k]$ using an algorithm for computing a *weak Gröbner basis* of a polynomial ideal and then computing a particular solution.[4]

Let SB denote a syzygy basis

$$\{(q_{11}, \cdots, q_{1n}), \ldots, (q_{w1}, \cdots, q_{wn})\}$$

for the set HE of the homogeneous equations

$$\begin{aligned}\{p_{11}X_1 + \cdots + p_{1n}X_n &= 0, \\ \vdots \qquad\qquad & \quad \vdots \\ p_{m1}X_1 + \cdots + p_{mn}X_n &= 0\}.\end{aligned}$$

Let $\pi = (q_1, \ldots, q_n)$ be a particular solution for the above set of nonhomogeneous equations obtained, for instance, using Baader's algorithm. From the particular solution, a most general unifier for the unification problem wrt *AGnHC* is computed (as stated above, *AGnHC* is unitary for unification without as well as with constants [16, 3]). The algorithm is nontrivial; we will not discuss the details here because of space limitations, but suggest the reader to refer to [2] for details.

Proposition 1. *$\pi' = (q'_1, \ldots, q'_n)$ is equivalent to π with respect to SB and hence, is also a particular solution iff there exist multipliers $b_1, \cdots, b_w$ such that $q_i - q'_i = b_1 q_{1i} + \cdots + b_w q_{wi}$ for each $1 \leq i \leq n$.*

[4] See Baader [2] for a definition of a weak Gröbner basis as well as syzygy basis.

Definition 7. *A linear constraint C on an extended alphabet $\Sigma' = \{Y_0, \ldots, Y_l, a_1, \ldots, a_l\}$, which includes $X_1, \ldots, X_n, h_1 \ldots, h_k$, is written as:*

$$Y_l \succ_C a_l \succ_C \ldots \succ_C a_2 \succ_C Y_1 \succ_C a_1 \succ_C Y_0,$$

where $\{X_1, \ldots, X_n\} \subseteq \{Y_0, \ldots, Y_l\}$ and $\{h_1, \ldots, h_k\} \subseteq \{a_1, \ldots, a_l\}$.

In the above, upper case symbols are used for variables, and lower case symbols are used for constants. Extra symbols are introduced for technical reasons so that between every two variables, there is a constant in the ordering.

Definition 8. *A solution β for the above set of nonhomogeneous linear equations satisfies a linear constraint C if and only if for every $Y \in \{Y_0, \ldots, Y_l\}$, $\beta(Y)$* does not contain *any of the symbols below Y in C. In other words, if $Y \succ_C a_j$ then $\beta(Y)$ does not contain any occurrence of a_j.*

Note that among variables, Y_l is the most constrained, since it cannot contain any of $a_1, \ \ldots \ , a_l$. On the other hand, from the point of view of constants, a_1 is the most constrained since it cannot appear in any variable other than Y_0.

5.1 Solutions Satisfying a Linear Constraint

Our goal is to find, among all particular solutions of the above set NHE of nonhomogeneous equations, a solution that satisfies the linear constraint C. In order to search for a particular solution that is equivalent to π and also satisfies C, an admissible ordering $\succ_t$ on terms (and polynomials) induced by C is defined in such a way that solutions satisfying C are minimal in this ordering.

Let $\tau_1, \ldots, \tau_n$ be new indeterminates. Consider the following set MSB of polynomials in $Z[h_1, \ldots, h_k, \tau_1, \ldots, \tau_n]$, constructed from SB:

$$\{q_{11}\tau_1 + \ldots + q_{1n}\tau_n, \ \ldots \ , \ q_{w1}\tau_1 + \ldots + q_{wn}\tau_n\}.$$

In addition, we include in MSB, additional polynomials $\{\tau_i\tau_j \mid 1 \leq i, j \leq n\}$ so that after simplification using these rules, every polynomial under consideration is *linear* in the τ_i's. Thus we only have to consider terms of the form $h_1^{d_1} \ldots h_k^{d_k}\tau_j$, where $d_1, \ldots d_k$ are nonnegative integers. Below we define an admissible ordering $\succ_t$ on such linear terms (in τ_i's) in $Z[h_1, \ldots, h_k, \tau_1, \ldots, \tau_n]$ induced by C. This term ordering $\succ_t$ is then extended to the simplified polynomials in $Z[h_1, \ldots, h_k, \tau_1, \ldots, \tau_n]$ which are linear in the τ_i's, in the usual way [5, 10].

The ordering $\succ_t$ is used to to construct a *strong* Gröbner basis $GMSB$ for the set MSB of polynomials [10]. The polynomial $\pi_p = q_1\tau_1 + \ldots q_n\tau_n$ corresponding to the particular solution π is then normalized using the Gröbner basis $GMSB$. Since the equivalence relation induced by MSB preserves solutions of NHE, the canonical (normal) form of π_p wrt $GMSB$ also corresponds to a particular solution. If this particular solution satisfies C (i.e., all terms are *good* in the sense defined below), then we get from the canonical form of π_p, a unifier for the unification problem wrt $AGnHC$ satisfying C. If the canonical form of π_p does not satisfy C, then the unification problem wrt $AGnHC$ does not have a solution satisfying C, since no polynomial in the equivalence class of π_p satisfies C (as every polynomial in the equivalence class of π_p is bigger than or equal to the normal form of π_p wrt $\succ_t$ whereas a polynomial corresponding to a solution satisfying C must be smaller wrt $\succ_t$).

In the following subsection, such an admissible ordering $\succ_t$ induced by a linear constraint C is defined on terms in $Z[h_1, \ldots, h_k, \tau_1, \ldots, \tau_n]$ which are linear in the τ_i's (i.e., whose degree in $\{\tau_1, \ldots, \tau_n\}$ is 1).

5.2 A New Way of Defining an Admissible Ordering on Terms

It is well-known that to construct a Gröbner basis of a polynomial ideal, a total admissible term ordering is needed. An admissible ordering must satisfy two properties:

1. For any term $t \neq 1$, $t \succ_t 1$, and
2. for any terms s, t, u, if $s \succ_t t$, then $u\ s \succ_t u\ t$.

Two commonly used admissible orderings in the Gröbner basis literature are the total degree ordering and the pure lexicographic ordering induced by a total ordering on indeterminates. Below we define an admissible ordering in a radically different way.

Consider any two terms s, t in $Z[h_1, \ldots, h_k, \tau_1, \ldots, \tau_n]$ which are linear in $\{\tau_1, \ldots, \tau_n\}$. Define $s \succ_t t$ iff $nf(s) >' nf(t)$, where the function nf stands for the normal form with respect to the reduction rules defined below in order to capture the linear constraint C. After defining nf and $>'$, we show that $\succ_t$ is admissible.

The term $nf(s)$ of a term s is over the extended alphabet $\Sigma_1 = \{a_1, ..., a_l, v_1 ..., v_l, a'_1, ..., a'_l, t_0, t_1, ..., t_l\}$, where a'_i is a copy of a_i distinguishing it from a_i, v_j's are introduced to represent badness in a term (there is one for every a_j), and t_j's are introduced to stand for Y_j's. Recall that $\{h_1, \ldots, h_k\} \subseteq \{a_1, \ldots, a_l\}$ and $\{X_1, \ldots, X_n\} \subseteq \{Y_0, \ldots, Y_l\}$; thus, corresponding to every X_i, there is a $Y_j = X_i$; similarly, corresponding to each τ_i, there is a $t_j = \tau_i$, i.e., $\{\tau_1, \ldots, \tau_n\} \subseteq \{t_0, \ldots, t_l\}$.

Below, *legal* term, *good* term, and *bad* term are defined on Σ_1 based on whether the term satisfies the linear constraint C.

Definition 9. *A term $s = {a_1}^{d_1}{a_2}^{d_2}...{a_l}^{d_l}\tau_i$ is called a* legal *term (only such terms appear in the polynomials in the basis MSB and in the computation of a Gröbner basis from MSB because of rules $\tau_i\tau_j \to 0$).*

A legal term $s = {a_1}^{d_1}{a_2}^{d_2}...{a_l}^{d_l}\tau_i$ is called a good *term if for each $1 \leq j \leq l$, $a_j \succ_C X_i$ in C, i.e., s satisfies the linear constraint C with respect to X_i.*

A legal term $s = {a_1}^{d_1}{a_2}^{d_2}...{a_l}^{d_l}\tau_i$ is called a bad *term if there exists a $1 \leq j \leq l$ such that it is not the case that $a_j \succ_C X_i$ in C, i.e., s does not satisfy the linear constraint C with respect to X_i.*
(A legal term that is not good, is bad.)

To capture the restrictions imposed by the linear constraint C on terms, we define the reduction rules on legal terms as:

$$\begin{aligned} a_i\, t_j &\to a_i'\, t_j \quad \text{if } a_i \succ_C Y_j. \\ a_i\, t_j &\to a_i'\, v_i t_j \text{ if } a_i \not\succ_C Y_j. \end{aligned}$$

So the normal form of a legal term ${a_1}^{d_1}{a_2}^{d_2}...{a_l}^{d_l}\tau_i$ with respect to the above rules is either

(i) $(a_1')^{d_1}(a_2')^{d_2}...(a_l')^{d_l}\tau_i$, or
(ii) $(a_1')^{d_1}(a_2')^{d_2}...(a_l')^{d_l}{v_{j_1}}^{d_{j_1}}...{v_{j_u}}^{d_{j_u}}...{v_{j_k}}^{d_{j_k}}\tau_i$,

where for each $1 \leq u \leq k$, it is not the case that $a_{j_u} \succ_C Y_i$.

Let $nf(t)$ denote the normal form of a term t by the above rules.

To compare the normal forms of s and t using the above reduction rules, we define the following lexicographic ordering $>'$ on symbols in Σ_1:

$$\begin{aligned} &a_1 >' \ldots >' a_l >' v_1 >' \ldots >' v_l >' t_l >' \ldots >' t_1 \\ &\quad >' t_0 >' a_1' >' \ldots >' a_l' \end{aligned}$$

This ordering is extended in a natural way to terms over Σ_1.

By the above reduction rules, the normal form of a bad term is greater than the normal form of a good term because only the normal form of a bad term has some v_j's which are greater than all a_i''s and t_i's.

Below, we sketch a proof that the ordering $\succ_t$ on legal terms in $Z[h_1, \cdots, h_l, t_1, \ldots, t_n]$, defined as

$$s \succ_t t \text{ iff } nf(s) >' nf(t)$$

is admissible.

For any $s \neq 1$, it is easy to see that $s \succ_t 1$.

The following lemma ensures that if $s \succ_t t$, then for any u, $u\,s \succ_t u\,t$, by proving that $nf(s) >' nf(t)$ iff $nf(u\,s) >' nf(u\,t)$.

Lemma 5. *Let Σ_1, $>'$, and nf be as defined above. Then $nf(s) >' nf(t)$ iff $nf(us) >' nf(ut)$ for all legal terms s, t, us and ut.*

To keep the main body of the report self-contained, a proof sketch of the lemma is given below. For the complete proof, the reader may consult the appendix.

Proof-sketch: (i) If $nf(s) >' nf(t)$ then $nf(u\ s) >' nf(u\ t)$: To prove this, it is enough to prove that for any symbol a_p in Σ_1, $nf(a_p\ s) >' nf(a_p\ t)$ if $nf(s) >' nf(t)$.

The key idea is this: since $nf(s) >' nf(t)$, multiplying by a_p on both sides could contribute either a'_p or $a'_p\ v_p$ to the normal forms. The only hard case is when $nf(s)$ contains t_i and $nf(t)$ contains t_j such that $Y_j \succ_C Y_i$ (i.e., Y_j is 'more constrained' than Y_i) and in addition, $a_p \succ_C Y_i$ and $a_p \not\succ_C Y_j$. Multiplying both sides by a_p will contribute a'_p to $nf(a_p\ s)$ and $a'_p\ v_p$ to $nf(a_p\ t)$. But since $nf(s) >' nf(t)$ there must be some a_q in s such that $a_q \not\succ_C Y_i$ and the power of a_q in s is larger than the power of a_q in t. Thus $nf(s)$ includes v_q whose power in s is larger than its power in $nf(t)$. But since $a_p \succ_C Y_i$, $v_q >' v_p$. Thus $nf(a_p\ s) >' nf(a_p\ t)$ (since the presence of v_q will "lexicographically nullify" the effect of including v_p).

(ii) If $nf(u\ s) >' nf(u\ t)$, then $nf(s) >' nf(t)$: This part is easier and similar. (Observe that $>'$ is a total ordering on terms.) □

6 Conclusion

We have presented a unification algorithm for analyzing cryptographic protocols using modular exponentiation and multiplication. This algorithm along with a related algorithm in [14] addresses theories, similar to theories arising in many cryptographic protocols including the RSA cryptosystem [18], one of the most popular public-key cryptosystems.

While the check for unifiability over the theory discussed in [14] is NP-complete, the check for unifiability for the theory discussed in this paper is likely to be worse; it can be easily shown to be EXPSPACE-hard. Unification algorithms for these theories are exponential in complexity (the algorithm in this paper is double-exponential given that the Gröbner basis algorithm used for solving syzygies is doubly-exponential). An interesting challenge is thus to identify special cases arising in cryptographic protocol analysis for which these unification algorithms are more efficient. As observed in [14], Pereira and Quisquater's algorithm [17] for analyzing the cliques protocol is not precisely a unification algorithm but it appears to be closely related, and it is possible that their approach could be applied to developing an efficient unification algorithm. Pereira and Quisquater were able to discover several security problems by solving a set of linear equations.

Also, it still remains to be seen in practice whether the integration of these unification algorithms into a software tool such as NRL Protocol Analyzer works more effectively than an approach based on narrowing implemented in it. (In the presence of associativity and commutativity (AC) properties of certain operations, it is unclear how simple narrowing is helpful unless an AC-unification algorithm is integrated into narrowing.)

As observed in [14], it will be necessary not only to develop algorithms for particular theories relevant to cryptographic protocol analysis, but to be able to combine them at will. Most protocols make use of several different forms of encryption, depending on the needs of the application.

References

1. F. Baader and K.U. Schultz. Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures. Proc. *11th Conference on Automated Deduction (CADE-11),* Saratoga Springs, NY, Springer LNAI 607, 1992, 50–65.
2. F. Baader. Unification in Commutative Theories, Hilbert's Basis Theorem, and Gröbner Bases. *J. ACM,* 40 (3), 1993, 477–503.
3. F. Baader and W. Nutt. Adding Homomorphisms to Commutative/Monoidal Theories, or: How Algebra Can Help in Equational Unification. Proc. *4th International Conference on Rewriting Techniques and Applications, RTA 91,* Springer LNCS 488, 1991, 124–135.
4. F. Baader and W. Snyder. *Unification Theory.* In: J.A. Robinson and A. Voronkov, editors, Handbook of Automated Reasoning. Elsevier Science Publishers, 2001.
5. B. Buchberger. Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory. In *Multidimensional Systems Theory* (N.K. Bose, ed.), Reichel, Dordrecht, 1985, 184–229.
6. D. Chaum. Security without Identification: Transaction Systems to Make Big Brother Obsolete. CACM 28 (10), 1985, 1030–1044.
7. J. Clark and J. Jacob. *A Survey of Authentication Protocol Literature: Version 1.0.* Unpublished Technical Report, Department of Computer Science, University of York, UK, Nov. 1997. Available at the URL: `www-users.cs.york.ac.uk/~jac/papers/drareviewps.ps`

8. M. Davis. *Computability and Unsolvability.* Dover Publications, 1982.
9. J.-M. Hullot. Canonical forms and unification. In *Proc. of the 5th conference on Automated Deduction (CADE-5),* Lecture Notes in Computer Science 87, 318–334.
10. A. Kandri-Rody and D. Kapur. Computing the Gröbner Basis of a Polynomial Ideal over Integers. Proc. *Third MACSYMA Users' Conference,* Schenectady, NY, July 1984, 436–451.
11. D. Kapur and G. Sivakumar. A Total, Ground Path Ordering for Proving Termination of AC-Rewrite Systems. Proc. *Rewriting Techniques and Applications, 8th International Conference, RTA-97,* Sitges, Spain (ed. Comon, H.), Springer LNCS 1231, June 1997, 142-156.
12. C. Meadows. Formal Verification of Cryptographic Protocols: A Survey. Proc. *AsiaCrypt 96,* 1996.
13. C. Meadows. The NRL Protocol Analyzer: An Overview. *J. Logic Programming,* 26(2), 1996, 113–131.
14. C. Meadows and P. Narendran. A Unification Algorithm for the Group Diffie-Hellman Protocol. Presented at the *Workshop on Issues in the Theory of Security (WITS 2002),* Portland, Oregon, Jan 2002.
15. P. Narendran, F. Pfenning, and R. Statman. On the Unification Problem for Cartesian Closed Categories. *Journal of Symbolic Logic,* 62 (2), June 97, 636–647.
16. W. Nutt. Unification in Monoidal Theories. Proc. *10th International Conference on Automated Deduction (CADE-10),* Kaiserslautern, West Germany, Springer LNCS 449, July 1990.
17. O. Pereira and J.-J. Quisquater. A Security Analysis of the Cliques Protocols Suites. Proc. *14th IEEE Computer Security Foundations Workshop,* June 2001, 73–81.
18. R.L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. CACM, 21 (2), 1978, 120–126.

Appendix

Lemma 5: Let the alphabet order on

$$\Sigma = \{a_1, \ldots, a_l, v_1, \ldots, v_l, a'_1, \ldots, a'_l, t_1, \ldots, t_l\}$$

be $a_1 > \ldots > a_l > v_1 > v_2 > \ldots > v_l > t_l > \ldots > t_1 > a'_1 > \ldots > a'_l$. Let $\succ$ be the pure lexicographic order on terms over Σ. Then $nf(s) \succ nf(t)$ iff $nf(ms) \succ nf(mt)$ for all legal terms s, t, ms and mt on Σ.

Proof: We first prove that if $nf(s) \succ nf(t)$ then $nf(ms) \succ nf(mt)$ To prove this, it is enough to prove that for any legal term s, t, character a_p on Σ, $nf(a_p s) \succ nf(a_p t)$ if $nf(s) \succ nf(t)$.

Without loss of generality, we assume that s and t are in the format of $a_{i_1}...a_{i_r}...a_{i_k}t_x$ $(1 \leq x \leq n,\ i_r < i_{r+1}\ (1 \leq r < k))$. Also we assume the normal forms of s and t are in the format of $a'_{i_1}...a'_{i_r}...a'_{i_k}t_x$ or $a'_{i_1}...a'_{i_r}...a'_{i_k}v_{z_1}...v_{z_r}...v_{z_q}t_x$ $(1 \leq x \leq n,\ i_r \leq i_{r+1},\ z_r \leq z_{r+1}\ (1 \leq r < k))$. Now we want to prove that $nf(a_p s) \succ nf(a_p t)$ if $nf(s) \succ nf(t)$. Since it is impossible that the normal form of a good term is pure lexicographically greater than the normal form of a bad term, we only need to consider the following three possibilities for s and t:

1. Both s and t are good terms. Since $nf(s) \succ nf(t)$, we will get the following reductions:

$$\begin{tikzcd}
s \arrow[d, equal] & & t \arrow[d, equal] \\
a_{i_1}...a_{i_k}t_x \arrow[d, "+"'] & & a_{j_1}...a_{j_l}t_y \arrow[d, "+"] \\
a'_{i_1}...a'_{i_k}t_x & \succ & a'_{j_1}...a'_{j_l}t_y
\end{tikzcd}$$

   Since $a'_{i_1}...a'_{i_k}t_x \succ a'_{j_1}...a'_{j_l}t_y$, it must be that $x \geq y$. After adding a_p to s and t, we would have the following three cases: (**Note**: since $x \geq y$, so it is not possible that $p > x$ and $p \leq y$. In other words, the case of $a_p s$ being a good term but $a_p t$ being a bad them is impossible).

   (a) $p > x$ and $p > y$, so both $a_p s$ and $a_p t$ are still good terms. The reductions would be:

$$\begin{tikzcd}
a_p s \arrow[d, equal] & a_p t \arrow[d, equal] \\
a_pa_{i_1}...a_{i_k}t_x \arrow[d, "+"'] & a_pa_{j_1}...a_{j_l}t_y \arrow[d, "+"] \\
a'_pa'_{i_1}...a'_{i_k}t_x \arrow[r, phantom, "\succ"] & a'_pa'_{j_1}...a'_{j_l}t_y
\end{tikzcd}$$

Clearly $nf(a_p s) \succ nf(a_p t)$.

(b) $p \leq x$ and $p > y$, so $a_p s$ is a bad term but $a_p t$ is a good term, the reductions would be:

$$\begin{tikzcd}
a_p s \arrow[d, equal] & a_p t \arrow[d, equal] \\
a_pa_{i_1}...a_{i_k}t_x \arrow[d, "+"'] & a_pa_{j_1}...a_{j_l}t_y \arrow[d, "+"] \\
a'_pa'_{i_1}...a'_{i_k}v_pt_x \arrow[r, phantom, "\succ"] & a'_pa'_{j_1}...a'_{j_l}t_y
\end{tikzcd}$$

$nf(a_p s) \succ nf(a_p t)$.

(c) $p \leq x$ and $p \leq y$, so both $a_p s$ and $a_p t$ are bad terms. The reductions would be:

$$\begin{tikzcd}
a_p s \arrow[d, equal] & a_p t \arrow[d, equal] \\
a_pa_{i_1}...a_{i_k}t_x \arrow[d, "+"'] & a_pa_{j_1}...a_{j_l}t_y \arrow[d, "+"] \\
a'_pa'_{i_1}...a'_{i_k}v_pt_x \arrow[r, phantom, "\succ"] & a'_pa'_{j_1}...a'_{j_l}v_pt_y
\end{tikzcd}$$

$nf(a_p s) \succ nf(a_p t)$.

2. s is a bad term and t is a good term. In this case, since $nf(s) \succ nf(t)$ the reductions would be:

$$\begin{tikzcd}
s \arrow[d, equal] & t \arrow[d, equal] \\
a_{i_1}...a_{i_k}t_x \arrow[d, "+"'] & a_{j_1}...a_{j_l}t_y \arrow[d, "+"] \\
a'_{i_1}...a'_{i_k}v_{z_1}...v_{z_r}...v_{z_q}t_x \arrow[r, phantom, "\succ"] & a'_{j_1}...a'_{j_l}t_y
\end{tikzcd}$$

(for each r ($1 \leq r \leq q$), there exists a u ($1 \leq u \leq k$) so that $i_u \leq x$ and $z_r = i_u$.)

After adding a_p to s and t, we will have the following cases:

(a) $p > y$. So a_pt is still a good term, and we will have the following possibilities:

i. $p > x$. The reductions would be:

$$\begin{tikzcd}
a_ps \arrow[d, equal] & a_pt \arrow[d, equal] \\
a_pa_{i_1}...a_{i_k}t_x \arrow[d, "+"'] & a_pa_{j_1}...a_{j_l}t_y \arrow[d, "+"] \\
a'_pa'_{i_1}...a'_{i_k}v_{z_1}...v_{z_r}...v_{z_q}t_x \arrow[r, phantom, "\succ"] & a'_pa'_{j_1}...a'_{j_l}t_y
\end{tikzcd}$$

(for each r $(1 \leq r \leq q)$, there exists a u $(1 \leq u \leq k)$ so that $i_u \leq x$ and $z_r = i_u$.)

$nf(a_ps) \succ nf(a_pt)$.

ii. $p \leq x$. The reductions would be:

$$\begin{tikzcd}
a_ps \arrow[d, equal] & a_pt \arrow[d, equal] \\
a_pa_{i_1}...a_{i_k}t_x \arrow[d, "+"'] & a_pa_{j_1}...a_{j_l}t_y \arrow[d, "+"] \\
a'_pa'_{i_1}...a'_{i_k}v_{z_1}...v_{z_r}...v_{z_q}v_pt_x \arrow[r, phantom, "\succ"] & a'_pa'_{j_1}...a'_{j_l}t_y
\end{tikzcd}$$

(for each $r(1 \leq r \leq q)$, there exists a $u(1 \leq u \leq k)$ so that $i_u \leq m$ and $z_r = i_u$.)

Thus $nf(a_ps) \succ nf(a_pt)$.

(b) $p \leq y$. So a_pt is a bad term, and we have the following possibilities:

i. $p > x$, so a_pt_x is a good term. The reductions would be:

$$\begin{tikzcd}
a_ps \arrow[d, equal] & a_pt \arrow[d, equal] \\
a_pa_{i_1}...a_{i_k}t_x \arrow[d, "+"'] & a_pa_{j_1}...a_{j_l}t_y \arrow[d, "+"] \\
a'_pa'_{i_1}...a'_{i_k}v_{z_1}...v_{z_r}...v_{z_q}t_x \arrow[r, phantom, "\succ"] & a'_pa'_{j_1}...a'_{j_l}v_pt_y
\end{tikzcd}$$

(for each r $(1 \leq r \leq q)$, there exists a u $(1 \leq u \leq k)$ so that $i_u \leq x$ and $z_r = i_u$.)

The reason $a'_pa'_{i_1}...a'_{i_k}v_{z_1}...v_{z_r}...v_{z_q}t_x \succ a'_pa'_{j_1}...a'_{j_l}v_pt_y$ is that since $p > x$ and for each r $(1 \leq r \leq q)$, $z_r \leq x$, so $z_r < p$, resulting $v_{z_r} > v_p$ according to our alphabet order on Σ. So $nf(a_p s) \succ nf(a_p t)$.

ii. $p \leq x$, the reductions would be:

$$\begin{tikzcd}
a_p s \arrow[d, equal] & & a_p t \arrow[d, equal] \\
a_p a_{i_1}...a_{i_k} t_x \arrow[d, "+"'] & & a_p a_{j_1}...a_{j_l} t_y \arrow[d, "+"] \\
a'_p a'_{i_1}...a'_{i_k} v_{z_1}...v_{z_r}...v_{z_q} v_p t_x \arrow[rr, phantom, "\succ"] & & a'_p a'_{j_1}...a'_{j_l} v_p t_y
\end{tikzcd}$$

(for each r ($1 \leq r \leq q$), there exists a u ($1 \leq u \leq k$) so that $i_u \leq m$ and $z_r = i_u$.)

Thus $nf(a_p s) \succ nf(a_p t)$.

3. Both s and t are bad terms. Since $nf(s) \succ nf(t)$, the reductions would be:

$$\begin{tikzcd}
s \arrow[d, equal] & & t \arrow[d, equal] \\
a_{i_1}...a_{i_k} t_x \arrow[d, "+"'] & & a_{j_1}...a_{j_l} t_y \arrow[d, "+"] \\
a'_{i_1}...a'_{i_k} v_{z_1}...v_{z_r}...v_{z_q} t_x \arrow[rr, phantom, "\succ"] & & a'_{j_1}...a'_{j_l} v_{w_1}...v_{w_r}...v_{w_g} t_y
\end{tikzcd}$$

(for each r ($1 \leq r \leq q$), there exists a u ($1 \leq u \leq k$) so that $i_u \leq x$ and $z_r = i_u$, $j_u \leq y$ and $j_u = w_r$.)

After adding a_p to s and t, we have the following sub-cases:

(a) $p > x$ and $p > y$. The reductions would be:

$$\begin{tikzcd}
a_p s \arrow[d, equal] & & a_p t \arrow[d, equal] \\
a_p a_{i_1}...a_{i_k} t_x \arrow[d, "+"'] & & a_p a_{j_1}...a_{j_l} t_y \arrow[d, "+"] \\
a'_p a'_{i_1}...a'_{i_k} v_{z_1}...v_{z_r}...v_{z_q} t_x \arrow[rr, phantom, "\succ"] & & a'_p a'_{j_1}...a'_{j_l} v_{w_1}...v_{w_r}...v_{w_g} t_y
\end{tikzcd}$$

(for each r ($1 \leq r \leq q$), there exists a u ($1 \leq u \leq k$) so that $i_u \leq x$ and $z_r = i_u$, $j_u \leq n$ and $j_u = w_r$.)

$nf(a_p s) \succ nf(a_p t)$.

(b) $p > x$ and $p \leq y$. The reductions would be:

$$\begin{tikzcd}
a_ps \arrow[d, equal] & a_pt \arrow[d, equal] \\
a_pa_{i_1}...a_{i_k}t_x \arrow[d, "+"'] & a_pa_{j_1}...a_{j_l}t_y \arrow[d, "+"] \\
a'_pa'_{i_1}...a'_{i_k}v_{z_1}...v_{z_r}...v_{z_q}t_x \arrow[r, phantom, "\succ"] & a'_pa'_{j_1}...a'_{j_l}v_{w_1}...v_{w_r}...v_{w_g}v_pt_y
\end{tikzcd}$$

(for each r ($1 \leq r \leq q$), there exists a u ($1 \leq u \leq k$) so that $i_u \leq x$ and $z_r = i_u$, $j_u \leq y$ and $j_u = w_r$.)

The reason $a'_pa'_{i_1}...a'_{i_k}v_{z_1}...v_{z_r}...v_{z_q}t_x \succ a'_pa'_{j_1}...a'_{j_l}v_{w_1}...v_{w_r}...v_{w_g}v_pt_y$ is that since $a'_{i_1}...a'_{i_k}$ $v_{z_1}...v_{z_r}...v_{z_q}t_x \succ a'_{j_1}...a'_{j_l}v_{w_1}...v_{w_r}...v_{w_g}t_y$ and $y > x$, so there must exists a r ($1 \leq r \leq q$) so that $z_r < w_r$ and $z_a = w_a$ for all a ($1 \leq a < r$), also since $p > x$, $z_r \leq x$, so $z_r < p$, resulting $v_{z_r} > v_p$, $a'_pa'_{i_1}...a'_{i_k}v_{z_1}...v_{z_r}...v_{z_q}t_x \succ a'_pa'_{j_1}...a'_{j_l}v_{w_1}...v_{w_r}...v_{w_g}v_pt_y$.

$nf(a_ps) \succ nf(a_pt)$.

(c) $p \leq m$ and $p > n$. The reductions would be:

$$\begin{tikzcd}
a_ps \arrow[d, equal] & a_pt \arrow[d, equal] \\
a_pa_{i_1}...a_{i_k}t_x \arrow[d, "+"'] & a_pa_{j_1}...a_{j_l}t_y \arrow[d, "+"] \\
a'_pa'_{i_1}...a'_{i_k}v_{z_1}...v_{z_r}...v_{z_q}v_pt_x \arrow[r, phantom, "\succ"] & a'_pa'_{j_1}...a'_{j_l}v_{w_1}...v_{w_r}...v_{w_g}t_y
\end{tikzcd}$$

(for each r ($1 \leq r \leq q$), there exists a u ($1 \leq u \leq k$) so that $i_u \leq x$ and $z_r = i_u$, $j_u \leq y$ and $j_u = w_r$.)

$nf(a_ps) \succ nf(a_pt)$.

(d) $p \leq x$ and $p \leq y$. The reductions would be:

$$\begin{tikzcd}
a_ps \arrow[d, equal] & a_pt \arrow[d, equal] \\
a_pa_{i_1}...a_{i_k}t_x \arrow[d, "+"'] & a_pa_{j_1}...a_{j_l}t_y \arrow[d, "+"] \\
a'_pa'_{i_1}...a'_{i_k}v_{z_1}...v_{z_r}...v_{z_q}v_pt_x \arrow[r, phantom, "\succ"] & a'_pa'_{j_1}...a'_{j_l}v_{w_1}...v_{w_r}...v_{w_g}v_pt_y
\end{tikzcd}$$

(for each r ($1 \leq r \leq q$), there exists a u ($1 \leq u \leq k$) so that $i_u \leq m$ and $z_r = i_u$, $j_u \leq n$ and $j_u = w_r$.)

$nf(a_ps) \succ nf(a_pt)$.

In the same way, we can prove that if $the\ normal\ form\ of\ ms \succ the\ normal\ form\ of\ mt$ then $the\ normal\ form\ of\ s \succ the\ normal\ form\ of\ t$. (omitted.)