An E-unification Algorithm for Analyzing Protocols that Use Modular Exponentiation

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. Modular multiplication and exponentiation are common operations in modern cryptography. Unification problems with respect to some equational theories that these operations satisfy are investigated. Two different but related equational theories are analyzed. A unification algorithm is given for one of the theories which relies on solving syzygies over multivariate integral polynomials with noncommuting indeterminates. For the other theory, in which the distributivity property of exponentiation over multiplication is assumed, the unifiability problem is shown to be undecidable by adapting a construction developed by one of the authors to reduce Hilbert's 10th problem to the solvability problem for linear equations over semi-rings. A new algorithm for computing strong Gröbner bases of right ideals over the polynomial ring $\mathsf{Z}<X_1, \ldots, X_n>$ is proposed; unlike earlier algorithms proposed by Baader as well as by Madlener and Reinert which work only for right admissible term orderings with the boundedness property, this algorithm works for *any* right admissible term ordering. The algorithms for some of these unification problems are expected to be integrated into Naval Research Lab.'s Protocol Analyzer (NPA), a tool developed by Catherine Meadows, which has been successfully used to analyze cryptographic protocols, particularly emerging standards such as the Internet Engineering Task Force's (IETF) Internet Key Exchange [11] and Group Domain of Interpretation [12] protocols. Techniques from several different fields – particularly symbolic computation (ideal theory and Gröbner basis algorithms) and unification theory — are thus used to address problems arising in state-based cryptographic protocol analysis.

1 Introduction

Modular arithmetic is the mainstay of many modern cryptographic algorithms. Arithmetical operations such as modular multiplication and exponentiation are

* 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.

part of several algorithms. For instance, the RSA and El Gamal algorithms use modular exponentiation for encryption/decryption, whereas the Diffie-Hellman and group Diffie-Hellman algorithms use it for key exchange. Abstracting equational properties of modular exponentiation and designing equational unification algorithms for the resulting theories for use in protocol analysis was initiated in [13]. Building on the work reported in [7], we have so far explored four related unification theories involving modular multiplication. These are described below.

The motivation for studying these equational theories comes from building a tool for analyzing authentication protocols against possible attack. The proposed unification algorithms are planned to be integrated into a software tool called the *NRL Protocol Analyzer* (*NPA*) developed by Catherine Meadows at the Naval Research Laboratory [10]. This tool employs a state-based approach for analyzing attacks on an authentication protocol, and has been used effectively to analyze, e.g., the Internet Engineering Task Force's (IETF) Internet Key Exchange [11], Group Domain of Interpretation [12] protocols, demonstrate known integrity flaws in the Encapsulating Security Protocol [19] (an Internet Standard), and discover a new attack on the Simmons Selective Broadcast Protocol [18]. Currently, the tool uses simple unification (over the empty theory) and a *narrowing* procedure for state exploration. Narrowing is used to find solutions with respect to (henceforth, abbreviated as *wrt*) a set of terminating rewrite rules capturing the semantics of primitive operations used in a protocol, thus simulating unification wrt the associated equational theory. However, if these primitive operations have the associativity and commutativity properties, which is often the case, then narrowing either does not work or is extremely inefficient.

To review the various equational theories discussed in this paper, consider the following set of axioms relating modular multiplication with modular exponentiation:

$$
\begin{array}{rclr}
x \cdot (y \cdot z) &=& (x \cdot y) \cdot z & A \\
x \cdot y &=& y \cdot x & C \\
x \cdot 1 &=& x & U \\
x \cdot x^{-1} &=& 1 & Inv \\
x^1 &=& x & Exp1 \\
1^x &=& 1 & Exp2 \\
(x \cdot y)^z &=& (x^z) \cdot (y^z) & Exp3 \\
(x^y)^z &=& (x^z)^y & Exp4 \\
(x^y)^z &=& x^{y \circ z} & Exp5 \\
x \circ (y \circ z) &=& (x \circ y) \circ z & A' \\
x \circ y &=& y \circ x & C' \\
x \circ 1 &=& x & U'
\end{array}
$$

The first four equations (A, C, U, Inv) characterize an Abelian group with $\cdot$ representing multiplication modulo a prime number p; for brevity, this axiomatization is referred by AG. The last three equations characterize an Abelian monoid with $\circ$ standing for multiplication modulo $p-1$. Operators $\cdot$ and $\circ$ share the unit 1. Modular exponentiation is denoted by $\hat{}$; $x \hat{} y$ is also written as x^y.

Note also that $Exp5$ implies a 'kind of' associativity for $\circ$, since

$$x^{(u\circ(v\circ w))} =_{Exp5} ((x^u)^v)^w =_{Exp5} x^{((u\circ v)\circ w)}.$$

We have investigated various theories that are obtained by keeping AG fixed and varying the theories that $\circ$ satisfies. Our main results so far are:

(i) The unification problem for the equational theory consisting of AG and axioms $(Exp1, Exp2, Exp3)$ is decidable; this theory is denoted as $\mathcal{E}_3$. This theory is a subset of the equational theories in (ii) and (iii) below. A unification algorithm for this problem is given by reducing it to that of finding constrained solutions of linear equations over $\mathsf{Z}{<}X_1, \dots, X_n{>}$, an algebraic structure similar to a polynomial ring over the integers, except that the indeterminates do not commute (i.e., the multiplication operation on the terms is not commutative — this structure is often called a *monoid semiring*). This latter problem is solved by computing *strong* Gröbner bases[1] of right ideals over $\mathsf{Z}{<}X_1, \dots, X_n{>}$. A new algorithm for computing strong Gröbner bases for right ideals over $\mathsf{Z}{<}X_1, \dots, X_n{>}$ is given; in contrast to the earlier algorithms proposed by Baader [1] for weak Gröbner bases and by Madlener and Reinert [9] for strong Gröbner bases assuming right admissible orderings with the *boundedness property*[2], the proposed algorithm computes strong Gröbner bases using any right admissible ordering.

Techniques employed here for solving linear equations over $\mathsf{Z}{<}X_1, \dots, X_n{>}$ are novel since the variables of the equations being solved have additional constraints of the form that certain indeterminates cannot appear in their solutions (these constraints are related to the occur check condition in simple unification, i.e., over the empty theory[3]). Such constraints are ensured by defining somewhat *unusual* kinds of admissible term orderings reflecting such constraints. These term orderings are very different from usual term orderings typically used in the literature for computing Gröbner bases of ideals, such as total degree and lexicographic, pure lexicographic, reverse lexicographic as well as block orderings.

(ii) When axiom $Exp4$ is added to $\mathcal{E}_3$, the unification problem for the extended equational theory remains decidable, as shown in [7]; we will denote this theory by $\mathcal{E}_{3,4}$. The unification algorithm uses an algorithm for computing strong Gröbner bases of polynomial ideals in $\mathsf{Z}[x_1, \cdots, x_n]$ developed by one

[1] A *weak* Gröbner basis of a (right) ideal simplifies polynomials in the (right) ideal to 0; however, polynomials equivalent with respect to the (right) ideal need not be simplified by the weak Gröbner basis to the *same* canonical form. In contrast, a *strong* Gröbner basis of a (right) ideal simplifies every polynomial to a unique canonical form such that equivalent polynomials with respect to the (right) ideal have the same canonical form; the canonical form of every polynomial in the (right) ideal is 0; see [6, 1, 9] for more details.

[2] An ordering has the boundedness property if and only if for every element, there are only finitely many elements smaller than the element in the ordering.

[3] and also the *linear constant restrictions* introduced by Baader and Schulz [3]

of the authors in 1984 [6]. Linear equations are solved under constraints in a way similar to (i) above. For details, the reader can refer to [7].

(iii) If the equational theory in (i) is extended by adding axioms $Exp5$, A', C' and U' instead of $Exp4$, then the unification problem for this equational theory, denoted by $\mathcal{E}_{3,5}$, is undecidable. (It is easy to see that $Exp4$ follows from C' and $Exp5$, thus implying that $Exp5$ is strictly stronger than $Exp4$ in this context.) The undecidability is mainly due to the distributivity property since if the distributivity axiom $Exp3$ is dropped from this equational theory—thus just keeping $Exp5$—then the unifiability problem is NP-complete [13].

The undecidability proof adapts an earlier proof showing the undecidability of the unification problem over the equational theory of isomorphisms over *Cartesian Closed Categories* developed by one of the authors [15]. Details can be found in [7].

(iv) If the equational theory in (i) is extended by adding axioms $Exp5$, A' and U' (in other words, $\cdot$ forms an Abelian group, whereas $\circ$ forms a *monoid*), denoted by $\mathcal{E}'_{3,5}$, the unification problem is still undecidable. This is shown by reducing Hilbert's tenth problem to the unification problem by adapting a construction developed by one of the authors [14] for showing undecidability of solving linear equations over polynomial semirings. In Section 2, we provide a sketch highlighting the main point; details can be found in [8].

The result mentioned in (ii) above was obtained earlier and is reported in [7] with detailed proofs. The undecidability result mentioned in (iii) is similar to the undecidability result reported in [7].

The paper thus brings together techniques from several different fields, particularly symbolic computation (ideal theory and Gröbner basis algorithms) and unification theory, to address problems arising in state-based cryptographic protocol analysis.

We state in the next section that unifiability for $\mathcal{E}'_{3,5}$ is undecidable. The rest of the paper focuses on the decidability of the unifiability check for $\mathcal{E}_3$. Section 3 relates unification over $\mathcal{E}_3$ to unification over $AGnH$ with an occur-check like condition. Section 4 discusses an algorithm for solving the unification problem over $AGnH$ with an occur-check-like condition. The algorithm uses a strong Gröbner basis algorithm to solve linear equations over a monoid semiring $\mathsf{Z}{<}X_1, \ldots, X_n{>}$. To ensure that the solution to the unification problem satisfy the occur-check like condition, unusual term orderings capturing this constraint are defined. The Gröbner basis algorithm is discussed in Section 5. It is shown that the algorithm computes a Gröbner basis for a right ideal using arbitrary right admissible term orderings including those which do not have the boundedness property. This algorithm is new since earlier algorithms worked only for admissible term orderings satisfying the boundedness property.

2 Undecidability of Unification over $\mathcal{E}'_{3,5}$

We reduce Hilbert's tenth problem to it by simulating numbers as well as multiplication over numbers. In the following, for a natural number k, $\bigcirc_k(b)$ denotes $\underbrace{b \circ b \ldots \circ b}_{k}$.

Theorem 1. *Unification over $\mathcal{E}'_{3,5}$ is undecidable.*

Proof Sketch: The following equations

$$
\begin{array}{ll}
1.\ {x_1}^b \cdot a =^? x_1 \cdot a^{\bigcirc_{k_1}(b)} & 6.\ {z_2}^c \cdot v_2 =^? y_2 \cdot z_2 \\
2.\ {x_2}^c \cdot a =^? x_2 \cdot a^{\bigcirc_{k_2}(c)} & 7.\ \qquad s_1 =^? v_2 \cdot v_2 \\
3.\ {y_1}^b \cdot v_1 =^? x_1 \cdot y_1 & 8.\ \qquad s_2 =^? s_1 \cdot v_1 \\
4.\ {y_2}^c \cdot v_1 =^? x_2 \cdot y_2 & 9.\ {x_3}^b \cdot a =^? x_3 \cdot a^{w_2} \\
5.\ {z_1}^b \cdot v_2 =^? y_1 \cdot z_1 & 10.\ {y_3}^b \cdot s_2 =^? x_3 \cdot y_3,
\end{array}
$$

force w_2 to be $\bigcirc_{k_1^2}(b)$, thus simulating squaring. Using the construction proposed in [14] with the above way of encoding squaring of numbers, Hilbert's tenth problem can be formulated as a unification problem in $\mathcal{E}'_{3,5}$. For a detailed proof, please refer to [8].

3 Relating Unification over $\mathcal{E}_3$ to Unification over $AGnH$

We consider the theory $\mathcal{E}_3$ consisting of AG and axioms $Exp1$, $Exp2$ and $Exp3$. We show that the equational unification problem for $\mathcal{E}_3$ is equivalent to the unification problem wrt the theory of Abelian groups with *noncommuting* homomorphisms, denoted by $AGnH$, but with an additional constraint. (The n stands for the number of homomorphisms.)

It is shown in [2] that the theory $AGnH$ is *unitary* with respect to unification without constants and it is also unitary with respect to unification with constants[4]. In Section 5 of [1], Baader showed that the unification problem of $AGnH$ reduces to solving linear equations over the polynomial ring $\mathsf{Z}{<}h_1, \ldots, h_n{>}$ where $h_1, \ldots, h_n$ are the noncommuting homomorphisms of $AGnH$, treated as indeterminates in the polynomial ring. He gave a unification algorithm for $AGnH$, which uses a weak Gröbner basis of a right ideal over $\mathsf{Z}{<}h_1, \ldots, h_n{>}$ for solving linear equations; as a result, Baader also proposed an algorithm for computing a weak Gröbner basis. We generalize Baader's algorithm by adding an additional key step to ensure that a given *linear constraint* (capturing an occur-check like condition) is satisfied by the unifier, so as to apply it to the equational unification problem for $\mathcal{E}_3$. This generalization needs a strong Gröbner basis of a right ideal over $\mathsf{Z}{<}h_1, \ldots, h_n{>}$. In a later section, we develop a new algorithm for computing a strong Gröbner basis of a right ideal over $\mathsf{Z}{<}h_1, \ldots, h_n{>}$.

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

3.1 Unification over $\mathcal{E}_3$ as a Combination of Theories

Let $\Sigma = \{\cdot, ^{-1}, 1, \hat{}\}$, the signature of $\mathcal{E}_3$.

Definition 1. *An $\mathcal{E}_3$-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.*

Definition 2. *An $\mathcal{E}_3$-unification problem S on Σ is called an* exponent $\mathcal{E}_3$-*unification problem, abbreviated as EXP, 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 $\mathcal{E}_3$-unification problem S on Σ is called a* simple $\mathcal{E}_3$-*unification problem if $S = S_1 \bigcup S_2$, where S_1 is an AG-unification problem and S_2 is an exponent $\mathcal{E}_3$-unification problem.*

It is easy to see that using abstraction, any $\mathcal{E}_3$-unification problem can be transformed into a simple $\mathcal{E}_3$-unification problem. For example, consider $S = \{w =^? (x^{(y^{u\cdot v})^{-1}\cdot z^{u'\cdot v'}})^{-1}\}$. Using abstractions, S is transformed to S':

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

where $z_1, z_2, z_3, z_4, z_5, z_6$ are new variables introduced for alien subterms in S.

3.2 Relating $AG + EXP$ to $AGnH$

Given a simple $\mathcal{E}_3$-unification problem S, for each equation of the form $x =^? y^w$ in S, we transform it into $x =^? h_w(y)$ where h_w is a homomorphism corresponding to the symbol w. Let $\mathcal{H}(S)$ denote the set of all homomorphisms introduced in this way, and, let $\Sigma_h = \{\cdot, ^{-1}, 1\} \cup \mathcal{H}(S)$. We call the transformed $AGnH$ problem a *h-image* of S.

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

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

It is not hard to see that the unifiability of T wrt $AGnH$ does not necessarily imply that the original problem S has a unifier wrt $\mathcal{E}_3$. For instance, the problem $\{x =^? a^x\}$ has no unifier wrt $\mathcal{E}_3$, whereas its h-image $\{x =^? h_x(a)\}$, when simply taken as a unification problem wrt $AGnH$ (with $n = 1$) is solvable. Thus something like an extended occur check must be enforced; this is done by solving the unifiability problem of T wrt $AGnH$ subject to *linear* constraints (including) $x \succ h_x$ for every homomorphism $h_x \in \mathcal{H}(S)$, i.e., a unifier θ of T should satisfy the condition that for every $x \in Var(T)$, $\theta(x)$ *does not contain* any occurrence of h_x, where $Var(T)$ denotes the set of all variables in T.

Definition 4. *Given a simple $\mathcal{E}_3$-unification problem S and its h-image T wrt $AGnH$, a* linear constraint C *is a total ordering $\succ_C$ 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 $\mathcal{E}_3$-unification problem S is said to be a* discriminating *unifier if and only if the following hold for all variables in $Var(S)$:*

1. *$\theta(u) \neq_{\mathcal{E}_3} 1$ for all u.*
2. *$\theta(v) =_{\mathcal{E}_3} \theta(w)$ iff $v = w$.*

Theorems 2 and 3 below relate the unification problem S wrt $\mathcal{E}_3$ to its h-image T wrt $AGnH$.

Theorem 2. *Given a simple $\mathcal{E}_3$-unification problem S and its h-image T wrt $AGnH$, if S has a discriminating unifier, then T is unifiable. Furthermore, there is a linear constraint C that the unifier of T satisfies.*

A detailed proof of this theorem is discussed in [8].

Theorem 3. *Given a simple $\mathcal{E}_3$-unification problem S and its h-image T wrt $AGnH$, if T has a solution which satisfies a linear constraint, then S is solvable.*

Proof. Consider all exponent equations $\{x_{u_1} =^? {x_{v_1}}^{x_{w_1}}, \ldots, x_{u_k} =^? {x_{v_k}}^{x_{w_k}}\}$ in S; similarly, also consider their corresponding equations $\{x_{u_1} =^? h_{x_{w_1}}(x_{v_1}), \ldots, x_{u_k} =^? h_{x_{w_k}}(x_{v_k})\}$ in the h-image T for S.

Let β be a ground unifier of T that satisfies a linear constraint C. From C, we can get a subconstraint C' on variables in $Var(T)$. Let $Var(T) = \{x_1, \ldots, x_n\}$. Assume without loss of generality that $C' = x_n \succ_C \cdots \succ_C x_i \succ_C \cdots \succ_C x_1$. Now we will use induction on C' to form θ. As the basis step, consider the first variable x_n in C'. Since x_n is the first variable, $\beta(x_n)$ should not contain any item below x_n in C, implying that $\beta(x_n)$ is composed of constants. Define $\theta(x_n) = \beta(x_n)$.

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

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

   $repp(a) = a$, where a is a constant on Σ_h.

   $repp(A \cdot B) = repp(A) \cdot repp(B)$, where A, B are terms on Σ_h.

   $repp(h_{x_{w_i}}(A)) = repp(A)^{\theta(x_{w_i})}$, where A is a term on Σ_h.

It can be shown that $s =_{AGnH} t$ iff $repp(s) =_{\mathcal{E}_3} repp(t)$. Therefore, $repp(\beta(x_{u_i}))$ $=_{\mathcal{E}_3} repp(h_{x_{w_i}}(\beta(x_{v_i})))$ for all $1 \leq i \leq n$. So $\theta(x_{u_i}) =_{\mathcal{E}_3} (\theta(x_{v_i}))^{\theta(x_{w_i})}$ for all $1 \leq i \leq n$. Thus θ is a solution for S. □

In the next section, we generalize Baader's algorithm for unifiability check for $AGnH$ to work with a linear constraint.

4 Unification over *AGnH* with a Linear Constraint

The unifiers of a unification problem wrt $AGnH$, where $h_1, \ldots, h_n$ are the non-commuting homomorphisms, correspond to the solutions of (nonhomogeneous) linear equations over the polynomial ring $\mathsf{Z}{<}h_1, \ldots, h_n{>}$ with $h_1, \ldots, h_n$ as indeterminates in the polynomial ring. The set of solutions of a homogeneous equation $p_1X_1 + \ldots + p_kX_k = 0$, where $p_1, \ldots, p_k \in \mathsf{Z}{<}h_1, \ldots, h_n{>}$and $X_1, \ldots, X_k$ are variables, is a finitely generated right $\mathsf{Z}{<}h_1, \ldots, h_n{>}$-semimodule. The nonhomogeneous equation $p_1X_1 + \ldots + p_kX_k = p_0$ has a solution iff p_0 is a member of the right ideal generated by $p_1, \ldots, p_k$. And, the membership problem for finitely generated right ideals is decidable.

Let NHE be a set of linear equations
$$\begin{aligned}\{p_{11}X_1 + \cdots + p_{1k}X_k &= p_1,\\ \vdots \qquad\qquad & \quad \vdots\\ p_{m1}X_1 + \cdots + p_{mk}X_k &= p_m\}\end{aligned}$$
where $p_{11}, \cdots, p_{1k}, \cdots, p_{m1}, \cdots, p_{mk}, p_1, \cdots, p_m$ are in $\mathsf{Z}{<}h_1, \ldots, h_n{>}$.

Baader [1] gave an algorithm for finding a solution for such nonhomogeneous linear equations. In his algorithm, he first computes a *weak Gröbner basis* G for a polynomial right ideal, then computes a right syzygy basis for homogeneous linear equations, and finally, a particular solution for the nonhomogeneous equations by using G. The algorithm is nontrivial; we will not discuss the details here but suggest the reader to refer to [1] for details. Let $\pi = (q_1, \ldots, q_k)$, where $\pi(X_i) = q_i$, be a particular solution for the above set NHE of nonhomogeneous equations obtained, for instance, using Baader's algorithm.

Let SB denote a right syzygy basis $\{(q_{11}, \cdots, q_{1k}), \ldots, (q_{w1}, \cdots, q_{wk})\}$ for the set HE of the homogeneous equations

$$\begin{aligned}\{p_{11}X_1 + \cdots + p_{1k}X_k &= 0,\\ \vdots \qquad\qquad & \quad \vdots\\ p_{m1}X_1 + \cdots + p_{mk}X_k &= 0\}.\end{aligned}$$

Proposition 1. *$\pi' = (q'_1, \ldots, q'_k)$ 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 $q_i - q'_i = q_{1i}b_1 + \cdots + q_{wi}b_w$ for each $1 \leq i \leq k$.*

4.1 Solutions Satisfying a Linear Constraint

We discuss how a solution satisfying a linear constraint C, if any, is searched for among all solutions of NHE.

Let $z_1, \ldots, z_k$ be new indeterminates. Consider the following set MSB of polynomials in $Z<h_1, \ldots, h_n, z_1, \ldots, z_k>$ constructed from SB above:

$$\{z_1\, q_{11} + \ldots + z_k\, q_{1k},\ \ldots\ ,\ z_1\, q_{w1} + \ldots + z_k\, q_{wk}\}.$$

Note that we only have to consider terms of the form $z_j\ \omega$, where $\omega \in (\mathcal{H}(S))^*$, where $\mathcal{H}(S) = \{h_1, \cdots, h_n\}$. Below, we define a right compatible ordering $\succ_t$ on such linear terms to capture linear constraint C in the following sense: if in a term $z_j\, \omega_j$, $h_i \succ_C X_j$ for each h_i appearing in ω_j, whereas in another term $z_{j'}\, \omega_{j'}$ we have some $h_{i'}$ in $\omega_{j'}$ such that $X_{j'} \succ_C h_{i'}$ then $z_j\, \omega_j \prec_t z_{j'}\, \omega_{j'}$.

The ordering $\succ_t$ is used to construct a strong Gröbner basis $GMSB$ for the set MSB of polynomials. The polynomial $\pi_p = z_1 q_1 + \ldots z_k q_k$ corresponding to the particular solution π above 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 π'. If this particular solution π' satisfies C (i.e., all terms are *good* in the sense defined in the next subsection), then we get a unifier for the unification problem wrt $AGnH$ satisfying C. If π' does not satisfy C, then the unification problem wrt $AGnH$ does not have a solution satisfying C, since a polynomial corresponding to a solution satisfying C must be smaller than the canonical form of π_p wrt $\succ_t$.

In the next subsection, we introduce how to define a term ordering $\succ_t$ that capture linear constraint C.

4.2 A Term Ordering Capturing Occur-Check Condition

Among all possible solutions of nonhomogeneous linear equations, to search a solution that satisfies a given linear constraint C, we define below a right admissible term ordering in a radically different way. This is in contrast to term orderings including total degree ordering, pure lexicographic ordering, reverse lexicographic ordering and block ordering, typically used in the literature on Gröbner basis computations.

A linear constraint C on an extended alphabet

$$\Delta = \{Y_0, \ldots, Y_l, a_1, \ldots, a_l\},$$

where $\{X_1, \ldots, X_k\} \subseteq \{Y_0, \ldots, Y_l\}$ and $\{h_1, \ldots, h_n\} \subseteq \{a_1, \ldots, a_l\}$, is written as:

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

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

Consider any two terms s, t in $\mathsf{Z}<h_1, \ldots, h_n, z_1, \ldots, z_k>$ which are linear in $\{z_1, \ldots, z_k\}$. Define $s \succ_t t$ iff $nf(s\#) >' nf(t\#)$, where nf stands for the normal form wrt the reduction rules defined below to capture the linear constraint C. After defining nf and $>'$, we show that $\succ_t$ is right admissible.

The term $nf(s\#)$ is over the extended alphabet

$$\Delta_1 = \{a_1, \ldots, a_l, v_1, \ldots, v_l, a_1', \ldots, a_l', z_1, \ldots, z_k, z_1', \ldots, z_k'\},$$

where the a_i's in Δ_1 are copies of the corresponding a_i's in Δ; further, corresponding to every X_i in Δ, we have z_i, z_i' in Δ_1.

Below, *legal* term, *good* term, and *bad* term are defined on $\{a_1, \ldots, a_l, z_1, \ldots, z_k\}$ based on whether the term satisfies the linear constraint C.

Definition 7. *A term $s = z_i\ s_a$, where $s_a \in \{a_1, \cdots, a_l\}^*$ 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).*
A legal term $s = z_i\ s_a$ is called a good *term if for each a_j appearing in s_a, $a_j \succ_C X_i$ in C, i.e., s_a satisfies the linear constraint C with respect to X_i.*
A legal term $s = z_i\ s_a$ is called a bad *term if there exists a_j appearing in s_a such that it is not the case that $a_j \succ_C X_i$ in C, i.e., s_a 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:

$$\begin{array}{ll}
z_i\ a_j \rightarrow a_i'\ z_j & \text{if } a_j \succ_C X_i. \\
z_i\ a_j \rightarrow v_j\ a_j'\ z_i & \text{if } X_i \succeq_C a_j. \\
z_i\ \# \rightarrow z_i'. & \\
a_i'\ z_j' \rightarrow z_j'\ a_i'. & \\
a_i'\ v_j \rightarrow v_j\ a_i'. & \\
v_i\ v_j \rightarrow v_j\ v_i & \text{if } X_i \succ_C X_j.
\end{array}$$

The normal form of a legal term $z_i\ s_a$ wrt the above rules is either

(i) ${z_i}'\ s_{a'}$, where $s_{a'}$ is obtained by systematically replacing every a_j in s_a by a_j'. or
(ii) $s_v\ {z_i}'\ s_{a'}$, where $s_{a'}$ is obtained from s_a as in (i) and s_v is a string of v_j's.

To compare $nf(s\#)$ with $nf(t\#)$ using the above reduction rules, we define the following ordering $>'$ on symbols in Δ_1:

$$v_1 >' \ldots >' v_l >' {z_k}' >' \ldots >' {z_1}' >' a_1' >' \ldots >' a_l'$$

This ordering is *extended* to terms over Δ_1 using pure lexicographic comparison.

Definition 8. *Given two terms $s = x_1 \ldots x_m,\ t = y_1 \ldots y_r$ on Δ_1 and the alphabet ordering on Δ_1 as defined above, we define $s >' t$ iff t is a prefix of s or there exists a* k *such that $x_k > y_k$ and $x_{k'} = y_{k'}$ for all $1 \leq k' < k$. Given two legal terms s, t on Δ_1, $s \succ_t t$ iff $nf(s\#) >' nf(t\#)$.*

By the above reduction rules, the normal form of a bad term $\succ_t$ the normal form of a good term because only the normal form of a bad term has some v_j's which are $>'$ all a_i''s and z_i''s. The ordering clearly does not have the boundedness property.

Below, we sketch a proof that the ordering $\succ_t$ on legal terms in $\mathsf{Z}{<}h_1, \cdots, h_n, z_1, \ldots, z_k{>}$ is right 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, $s\,u \succ_t t\,u$, by proving that $nf(s\#) >' nf(t\#)$ iff $nf(su\#) >' nf(tu\#)$.

Lemma 1. *Let Δ_1, $>'$, and nf be as defined above. Then $nf(s\#) >' nf(t\#)$ iff $nf(su\#) >' nf(tu\#)$ for all legal terms s, t, su and tu.*

Proof-sketch: (i) If $nf(s\#) >' nf(t\#)$ then $nf(su\#) >' nf(tu\#)$: To prove this, it is enough to prove that for any symbol a_p in Δ_1, $nf(sa_p\#) >' nf(ta_p\#)$ if $nf(s\#) >' nf(t\#)$.

Since $nf(s\#) >' nf(t\#)$, right multiplying by a_p on both s and t could contribute either a_p' or $v_p a_p'$ to the normal form of $sa_p\#$ and $ta_p\#$. The only hard case is when $nf(s\#)$ contains z_i and $nf(t\#)$ contains z_j such that $X_j \succ_C X_i$ (i.e., X_j is 'more constrained' than X_i) and in addition, $a_p \succ_C X_i$ and $a_p \not\succ_C X_j$. Right multiplying both sides by a_p will contribute a_p' to $nf(sa_p\#)$ and $v_p a_p'$ to $nf(ta_p\#)$. But since $nf(s\#) >' nf(t\#)$ there must be some a_q in s such that $a_q \not\succ_C X_i$ and the number of a_q's in s is larger than the number of a_q's in t. Thus $nf(s\#)$ includes v_q whose power in $nf(s\#)$ is larger than its power in $nf(t\#)$. But since $a_p \succ_C X_i$, $v_q >' v_p$. Thus $nf(sa_p\#) >' nf(ta_p\#)$ since the terms are compared lexicographically.

(ii) If $nf(su\#) >' nf(tu\#)$, then $nf(s\#) >' nf(t\#)$: This is easier since $>'$ is a total ordering on terms. For a detailed proof, see [8] □

5 A Gröbner Basis Algorithm for Right Ideals

Gröbner bases for polynomials over Z have been considered in for example [6], and more generally for polynomials over Euclidean rings. Here, we are interested in constructing Gröbner bases for finitely generated right ideals in $\mathsf{Z}{<}X_1, \ldots, X_n{>}$, where $X_1, \ldots, X_n$ are the noncommuting indeterminates. In [1], Baader gave an algorithm for constructing weak Gröbner bases for right ideals in $\mathsf{Z}{<}X_1, \ldots, X_n{>}$. But in his algorithm only right compatible orderings with the boundedness property can be used to construct weak Gröbner bases.[5] In [9], an algorithm for computing strong Gröbner bases for right ideals in $\mathsf{Z}{<}X_1, \ldots, X_n{>}$ is given but that too is for right admissible term orderings with the boundedness property. In this section, we give an algorithm which can be used to construct Gröbner bases for polynomial right ideals in $\mathsf{Z}{<}X_1, \ldots, X_n{>}$ based on any right admissible term ordering.

[5] Recall that an ordering has the boundedness property if and only if for every element, there are only finitely many elements less than it in the ordering.

For $m, n \in \mathsf{Z}$, we define the ordering $\succ_{\mathsf{Z}}$ as follows: $m \succ_{\mathsf{Z}} n$ if either $|m| > |n|$, or $|m| = |n|$ and $m < n$. Thus $-5 \succ_{\mathsf{Z}} 5$.

The set of terms we consider is the monoid $W_n = \{X_1, \ldots, X_n\}^*$.

Definition 9. *A total ordering $\succ$ on W_n is called right admissible iff it satisfies:*

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

A right admissible ordering $\succ$ on terms can be extended to monomials as follows: Let $a, b \in \mathsf{Z}$ and $s, t \in W_n$. Then $as \succ bt$ iff

1. $s \succ t$, or
2. $s = t$ and $a \succ_{\mathsf{Z}} b$.

Given an integer c, c induces an equivalence relation on Z as follows: $a =_c b$ iff there exists a k_1 such that $a = k_1 * c + b$ where $a, b, k_1 \in \mathsf{Z}$. Using this equivalence relation $=_c$, we define the *remainders* with respect to c as follows: Consider all equivalence classes induced by $=_c$. From each equivalence class, the smallest element wrt $\succ_{\mathsf{Z}}$ is a *remainder* of c. For example, the remainders of 5 are $0, 1, -1, 2, -2$; 3 is not a remainder of 5; similarly, the remainders of 4 are $0, 1, -1, 2$.

Let $\succ$ be a right admissible ordering and let $f = at + g$ be a polynomial in $\mathsf{Z}{<}X_1, \ldots, X_n{>}$ such that $t \in W_n$ is the greatest term in f with respect to $\succ$ and $a \in Z \backslash \{0\}$ is the coefficient of t in f. Then t is called the *head term* of f, denoted as $HT(f)$; a is called the *head coefficient* of f, denoted as $HC(f)$; $a\ t$ is called the *head monomial* of f, denoted as $HM(f)$, and $g = f - a\ t$ is called the *rest* of f, denoted as $R(f)$.

A right-admissible ordering $\succ$ on monomials can be extended in the natural way to an ordering $\succ$ on polynomials by comparing head monomials first and if these are equal, recursively comparing the rest of the polynomials. Sets of polynomials can be compared using its multiset extension $\succ\!\!\succ$. Since $\succ$ is well-founded, $\succ\!\!\succ$ are well-founded too.

Definition 10. *Given a set F of polynomials, for any $f, g \in \mathsf{Z}{<}X_1, \ldots, X_n{>}$, f can be reduced to g wrt F, denoted by $f \rightarrow_F g$, iff*

1. *f contains a term t with coefficient a, (i.e. f contains a monomial at).*
2. *F contains a polynomial h such that $t = HT(h) \cdot s$ for some s and a is not a* remainder *of $HC(h)$.*
3. *$g = f - b\ h\ s$ where $a = b\ HC(h) + c$ and c is a* remainder *of $HC(h)$.*

We use $nf_F(f)$ to denote a normal form of f with respect to $\rightarrow_F$.

Lemma 2. *Let $F = \{f_1, \ldots, f_k\}$ be a set of polynomials. For any f_i $(1 \leq i \leq k)$, if $f_i \rightarrow^*_{F \backslash \{f_i\}} f'_i$, then $F \backslash \{f_i\} \cup \{f'_i\}$ generates the same right ideal as F.*

Proof. Based on the definition of reduction rules $\rightarrow_{F \backslash \{f_i\}}$, $f_i = f'_i + \sum a_j\ f_{h_j}\ t_j)$ where $f_{h_j} \in F \backslash \{f_i\}$. So $F \backslash \{f_i\} \cup \{f'_i\}$ generates the same right ideal as F. $\square$

Algorithm 5.1: Gröbner basis Algorithm for Right Polynomial Ideals.

In the beginning, $F_0 = \{p_1, \ldots, p_m\}$ and all pairs of indices are unmarked. Without loss of generality, we assume all $HC(p_i) > 0$ $(1 \le i \le m)$. Assume that F_k $(k \le 0)$ is already defined. If there is the zero polynomial 0 in F_k, we erase it. As long as there are $f = p_i$ and $g = p_j$ in F_k such that:

(a) (i, j) is not marked and
(b) $f = at + R(f)$ and $g = btr + R(g)$ for $a, b \in Z^+$, and $t, r \in W_n$, we do the following:
  1. $a \le b$. Let $b = c\, a + d$ for some $c,\ d$, where d is a remainder of a. Define $g_1 = g - c\ f\ r = d\ t\ r + R(g) - c\ R(f)\ r$. Reduce g_1 to its normal form wrt F_k. If $HC(nf_{F_k}(g_1)) \ge 0$, define $g_1' = nf_{F_k}(g_1)$. Otherwise define $g_1' = -nf_{F_k}(g_1)$. $F_{k+1} = (F_k \cup \{g_1'\} \setminus \{g, 0\})$. Since $g = g_1 + c\ f\ r$, F_{k+1} generates the same right ideal as F_k by **Lemma 1**. Note that $HM(g_1') \prec HM(g)$.
  2. $r \ne 1, a > b$ and b does not divide a. Let $a = c\, b + d$ for some $c,\ d$, where d is a remainder of b. We define $g_1 = f\ r - c\ g = d\ t\ r + R(f)\ r - c\ R(g)$. Since $d < b$, g is reducible by g_1. Let $b = c_1\ d +\ d_1$ where d_1 is a remainder of d, and $g_2 = g - c_1\ g_1$. Reduce g_1, g_2 to their normal forms $g_1' = nf_{F_k}(g_1)$ and $g_2' = nf_{F_k}(g_2)$ wrt F_k. If $HC(g_1') \ge 0$, define $g_1'' = g_1'$, otherwise define $g_1'' = -g_1'$. Similarly if $HC(g_2') \ge 0$, define $g_2'' = g_2'$ and otherwise define $g_2'' = -g_2'$. Now $F_{k+1} = (F_k \cup \{g_1'', g_2''\} \backslash \{g, 0\})$. In other words, g is deleted and non-zero normal forms are added to F_k. Since $g = g_2 + c_1\ g_1$, F_{k+1} generates the same ideal as F_k by **Lemma 1**. Again, $HM(g_1'), HM(g_2') \prec HM(g)$
  3. $r \ne 1, a > b$ and b divides a. That is, there exists c such that $a = c\ b$. Define $g_1 = f\ r - c\ g = R(f)\ r - c\ R(g)$. Reduce g_1 to its normal form wrt F_k. If $nf_{F_k}(g_1) \ge 0$, define $g_1' = g_1$, otherwise define $g_1' = -g_1$. Define $F_{k+1} = F_k \cup \{g_1'\}$. Since $g_1 = f\ r - c\ g$, F_{k+1} generates the same ideal as F_k by **Lemma 1**. Mark (i, j). Here $HT(g_1) \prec HT(g)$.

Note that there is no need to mark the pair in cases 1 and 2, since g is deleted in both cases. For cases $1, 2$, clearly $F_k \ggcurly F_{k+1}$. For case 3, however, it is possible that $F_{k+1} \ggcurly F_k$. It is easy to see that if $p, q \in F_k$ such that $HT(p) = HT(q)$, then due to case 1 above, eventually one of these polynomials will be deleted from the basis leaving at most one polynomial with any head term.

Lemma 3. *Algorithm 5.1 terminates on any finite input basis of polynomials.*

Proof-sketch: The proof follows from two crucial insights: (i) a new polynomial added to a basis during the algorithm (due to any of cases 1, 2 and 3) is smaller than the largest polynomial in the basis because its head term is lower than or equal to the head terms of the polynomials used in its generation, and (ii) the head term of any polynomial has only finitely many prefixes, and consequently, only finitely many new polynomials can be generated using any given polynomial.

The proof is by contradiction. Consider a run of the algorithm starting with a basis G_0: $G_0, G_1, \cdots, G_i, G_{i+1}, \cdots$, where G_{i+1} is obtained from G_i by one of the cases.

Claim: There exists an i such that every basis $G_{i'}, i' \geq i$, has the same largest polynomial, say f_i (since the ordering $\succ$ is well-founded)); furthermore, no overlaps involving f_i are considered subsequently, i.e., in all $G_{i'}, i' \geq i$.

It can then be shown that any Gröbner basis of $G_i \setminus \{f_i\}$ when extended with f_i is a Gröbner basis of G_i.

Repeat the above argument on $G_i \setminus \{f_i\}$. The largest polynomial in $G_i \setminus \{f_i\}$ is lower in the ordering $\prec$ than f_i. This process must terminate because the term ordering is well-founded. This contradicts the assumption that the algorithm goes on forever. □

The correctness of the above Gröbner basis algorithm is established by a series of lemmas, patterned after the proof in [1] using ideas from [6] to establish that the proposed algorithm indeed computes a strong Gröbner basis for a right ideal over $\mathsf{Z}{<}X_1, \ldots, X_n{>}$. Proofs had to be omitted because of space limitations; an interested reader can consult [8].

Lemma 4. *Starting with a finite input basis G, let Algorithm 5.1 terminate in G_n by generating a sequence of bases $G, G_1, \ldots, G_n$. For any term t, if $f = \sum_{g_i \in G} g_i a_i$ with $HT(g_i a_i) \prec t$, then $f = \sum_{g'_i \in G_n} g'_i a'_i$ with $HT(g'_i a'_i) \prec t$.*

Lemma 5. *Starting with a finite input basis G, let Algorithm 5.1 terminate in G_n. Assume $G_n = \{g_1, \cdots, g_m\}$. For any $f \in <G>$, there exist polynomials $a_1, \ldots, a_n$ such that $f = \sum_{g_i \in G_n} g_i a_i$ and $|\{i \mid HT(g_i a_i) = HT(f)\}| = 1$.*

Lemma 6. *Starting with a finite input basis G, let Algorithm 5.1 terminate in G_n. Assume $G_n = \{g_1, \cdots, g_m\}$. Any polynomial $f \in <G>$ can be reduced to 0 with respect to $\rightarrow_{G_n}$.*

Theorem 4. *Starting with a finite input basis G, let Algorithm 5.1 terminate in G_n by generating a sequence of bases $G, G_1, \ldots, G_n$. Then, G_n is a* strong *Gröbner basis.*

The key idea in the proof is that for any two polynomials p_1 and p_2, if $p_1 - p_2$ reduces to 0, then one of p_1 and p_2 are reducible by $\rightarrow_{G_n}$. Further, the ordering $\succ_{\mathsf{Z}}$ on Z satisfies the unique remainder property discussed in [6].

6 Conclusion

Unification problems for theories admitting modular multiplication and exponentiation are discussed. As shown above, for theories that admit the distributivity property of exponentiation over multiplication, unifiability checks are undecidable. However, if this property is excluded, then unification algorithms for various restricted theories, depending upon properties of exponentiation used, can be obtained. In most cases, these algorithms are of high complexity.

In order to integrate these unification algorithms into the NRL Protocol Analyzer (NPA) and effectively use them for cryptographic protocol analysis, it will be useful to further specialize these algorithms so as to keep their complexity manageable in practice. That is likely to be the main challenge in investigating the effectiveness of the approach proposed in [13]. This issue needs further investigation.

References

1. F. Baader. Unification in Commutative Theories, Hilbert's Basis Theorem, and Gröbner Bases. *J. ACM,* 40 (3), 1993, 477–503.
2. F. Baader and W. Nutt. Adding Homomorphisms to Commutative/Monoidal Theories, or: How Algebra Can Help in Equational Unification. Proc. *4th Intl. Conf. on Rewriting Techniques and Applications (RTA 91),* LNCS 488, 1991, 124–135.
3. 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.
4. 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`.
5. M. Davis. *Computability and Unsolvability.* Dover Publications, 1982.
6. 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. See also A. Kandri-Rody and D. Kapur. An Algorithm for Computing the Gröbner Basis of a Polynomial Ideal over an Euclidean Ring. Journal of *Symbolic Computation,* 6 (1), August 1988, 37-57.
7. D. Kapur, P. Narendran, and L. Wang. *A Unification Algorithm for Analysis of Protocols with Blinded Signatures.* TR 02-5, Department of Computer Science, SUNY, Albany, NY. To appear in the Festschrift for Jörg Siekmann (Dieter Hutter, Werner Stephan, eds.), Lecture Notes in Artificial Intelligence 2605, Springer.
8. D. Kapur, P. Narendran, and L. Wang. *Analyzing Protocols that use Modular Exponentiation: Semantic Unification Techniques* Technical Report, Department of Computer Science, SUNY, Albany, NY. An expanded version of this paper.
9. K. Madlener and B. Reinert. On Gröbner bases in Monoid and Group Rings. SEKI Report SR-93-08, Universität Kaiserslautern, Germany.
10. C. Meadows. The NRL Protocol Analyzer: An Overview. *J. Logic Programming,* 26(2), 1996, 113–131.
11. C. Meadows. Analysis of the Internet Key Exchange protocol using the NRL Protocol Analyzer. In: Proc. the 1999 Symp. on Security and Privacy, IEEE.
12. C. Meadows, P. Syverson and I. Cervesato. Formal Specification and Analysis of the Group Domain of Interpretation Protocol using NPATROL and the NRL Protocol Analyzer. To appear in the *Journal of Computer Security.*
13. C. Meadows and P. Narendran. A Unification Algorithm for the Group Diffie-Hellman Protocol. *Workshop on Issues in the Theory of Security (WITS 2002),* Portland, OR, Jan 2002.
14. P. Narendran. On solving linear equations over polynomial semirings. In: Proc. 11th Annual Symp. on Logic in Computer Science (LICS), NJ, July 96, 466–472.
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. O. Pereira and J.-J. Quisquater. A Security Analysis of the Cliques Protocols Suites. Proc. *14th IEEE Computer Security Foundations Workshop,* June 2001.
17. R.L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. CACM 21 (2), 1978, 120–126.
18. G. Simmons and C. Meadows. The Role of Trust in Information Integrity Protocols. *Journal of Computer Security* 3 (2), 1994.
19. S.G. Stubblebine and C. Meadows. On Searching for Known and Chosen Cipher Pairs using the NRL Protocol Analyzer. Presented at the DIMACS Workshop on Design and Formal Verification of Security Protocols, September 1997.