An Algebra for Symbolic Diffie-Hellman Protocol Analysis

Daniel J. Dougherty Joshua D. Guttman*

Worcester Polytechnic Institute
{dd,guttman}@wpi.edu

Abstract. We study the algebra underlying symbolic protocol analysis for protocols using Diffie-Hellman operations. Diffie-Hellman operations act on a cyclic group of prime order, together with an exponentiation operator. The exponents form a finite field: this rich algebraic structure has resisted previous symbolic approaches.

We define an algebra that validates precisely the equations that hold almost always as the order of the cyclic group varies. We realize this algebra as the set of normal forms of a particular rewriting theory.

The normal forms allow us to define our crucial notion of *indicator*, a vector of integers that summarizes how many times each secret exponent appears in a message. We prove that the adversary can never construct a message with a new indicator in our adversary model. Using this invariant, we prove the main security goals achieved by UM, a protocol using Diffie-Hellman for implicit authentication.

Despite vigorous research in symbolic analysis of security protocols, many limitations remain. While systems such as NPA-Maude [21], ProVerif [8], AVISPA [3,5], CPSA [36], and Scyther [16] are extremely useful, great ingenuity is still needed—as for instance in [31]—for the analysis of protocols that use fundamental cryptographic ideas such as Diffie-Hellman key agreement [17], henceforth, DH. Moreover, important protocols, such as the implicitly authenticated key-agreement protocol MQV [7], appear to be out of reach of known symbolic techniques. Indeed, for these protocols, computational techniques have led to arduous proofs after which controversy remains [27,29,30,33]. In this paper, we develop algebraic ideas that allow us to give rigorous proofs of security goals such as authentication and confidentiality in a symbolic model. Moreover, our techniques also help identify the security goals that the protocol does not achieve.

DH protocols work in a cyclic group of prime order q, which we will write multiplicatively, using an agreed-upon generator g. For a particular session, A and B choose random values x, y respectively, raising a base g to these scalar powers:

$$A, x \qquad \bullet \xrightarrow{g^x} \qquad \xleftarrow{g^y} \bullet \qquad B, y \tag{1}$$

* We gratefully acknowledge support by the National Science Foundation under grant CNS-0952287.

They can then each compute the value $(g^y)^x = g^{xy} = (g^x)^y$ as a new shared secret for A, B. The Decisional Diffie-Hellman assumption (DDH) says that, in suitable groups, any observer who has observed neither x nor y, cannot distinguish g^{xy} from the g^z we would get from a randomly chosen z.

This basic protocol—while secure against a passive adversary, who observes messages, but can neither create them nor alter (or misdirect) messages of compliant principals—is, however, vulnerable to an active attacker. The adversary chooses his own values w, g^w, substituting g^w for the values each participant should receive. Then the two participants will end up with different keys, g^{xw} and g^{yw}, unfortunately each shared with the attacker.

One idea to avoid this man-in-the-middle attack is for each of the principals A and B to maintain a long-term secret value. We will write A's long term secret as a, and B's as b. They publish the long term public values $Y_A = g^a, Y_B = g^b$, having a certificate authority certify the bindings to A and B. Now any pair of participants may each use the long term public value of the other—and their own long term secrets—to compute the same fresh secret, in such a way that no principal other than A or B can. The "Unified Model" UM of Ankney, Johnson, and Matyas [2] is an example. A and B send only the messages shown in Eqn. 1. For clarity, the value B receives, purportedly from A, will be called R_A. A receives the value R_B, purportedly from B. Without adversary interference, $R_A = g^x$ and $R_B = g^y$. Letting $\mathsf{h}(x)$ be a hash function, A and B compute their keys:

$$A: \; k = \mathsf{h}({Y_B}^a \parallel {R_B}^x) \quad B: \; k = \mathsf{h}({Y_A}^b \parallel {R_A}^y), \tag{2}$$

obtaining the shared value $\mathsf{h}(g^{ab} \parallel g^{xy})$ if $R_A = g^x$ and $R_B = g^y$. We will present a technique for proving authentication and confidentiality results about protocols such as this.

The heart of this paper develops a well-behaved rewriting theory for DH values, which yields a powerful tool for symbolic analysis. The challenge for such a theory derives from the fact that, since we are operating in a cyclic group of prime order, the exponents form a *field*. Although UM uses only the field multiplication, some protocols (including MQV) also use the field addition. This is challenging for rewriting-based approaches to protocol analysis since the theory of fields does not admit an axiomatization using equations, or even conditional equations. The standard axiomatization uses negation to say that 0 has no multiplicative inverse; to see that there can be no conditional-equational axiomatization, note that the category of fields is not closed under products. This paper makes the following contributions:

1. We define an order-sorted equational theory $\mathsf{AG\hat{}}$ whose models include all fields. We equip $\mathsf{AG\hat{}}$ with a rewrite system modulo associativity and commutativity (AC), and show that this system is terminating and confluent modulo AC: an equation $s = t$ is derivable in $\mathsf{AG\hat{}}$ if and only if s and t rewrite to the same normal form modulo AC. The free algebra over this rewrite system offers a natural DH message algebra. (Section 1.)
2. We show, via a model-theoretic argument using ultraproducts, that $\mathsf{AG\hat{}}$ captures uniform equality in the theory of finite fields. Namely, if $s = t$ is an

equation that is valid in the field $\mathbb{F}_q$ of characteristic q for infinitely many q, then $\mathsf{AG}\hat{}$ proves $s = t$. In particular, $\mathsf{AG}\hat{}$ proves every equation that is valid in $\mathbb{F}_q$ aymptotically as q increases. (Section 2.)
3. We use $\mathsf{AG}\hat{}$ to prove Thm. 12, the *indicator theorem*, a symbolic analogue to the computational Diffie-Hellman assumption (CDH). It states that the adversary cannot obtain a new exponentiated value t^{xy} without access either to x, or to y, or to some value that already included t^{xy}. Thm. 12 gives a proof method in $\mathsf{AG}\hat{}$ that avoids unification. (Section 3.)
4. We apply the indicator theorem within the strand space framework (introduced in Section 4) to prove that UM meets its authentication and confidentiality goals (construed as trace properties). We also explain why it does not meet another goal, resisting impersonation attacks. (Section 5.)

Elsewhere, we apply our method to more challenging protocols, e.g. MQV [18].

Related Work. Within the symbolic model, there has been substantial work on some aspects of DH, starting with Boreale and Buscemi [9], which provides a symbolic semantics [1, 22, 34] for a process calculus with algebraic operations for DH. Their symbolic semantics is based on unification.

Indeed, symbolic approaches to protocol analysis have relied on unification as a central part of their reasoning. Goubault-Larrecq, Roger, and Verma [24] use a method based on Horn clauses and resolution modulo AC, providing automated proofs of passive security. Maude-NPA [20, 21] is also usable to analyze many protocols involving DH, again depending heavily on unification. Tamarin [15] offers a new approach to analysis, also relying on unification.

All of these approaches model the multiplication in the exponents, but do not explicitly model the addition. This suffices for many protocols, but not for protocols such as Menezes-Qu-Vanstone MQV [7] and Cremers-Feltz CF [14], in which the ring structure in the exponents is used in the protocol definition. Indeed, even in protocols which use only the multiplicative structure, the adversary may choose to use the ring or field properties. The richer theory is needed to prove no new attacks can arise.

This field structure combines poorly with the heavy reliance of previous approaches on unification. Unifiability is undecidable in the theory of rings, by the unsolvability of Hilbert's tenth problem. There are, however, many related theories for which undecidability is not known, for instance the diophantine theory of the rationals [6]; see the beautiful paper by Kapur, Narendran, and Wang [28].

Küsters and Truderung [31] finesse this issue by rewriting protocol analysis problems. The original problems use an AC theory involving exponentiation. They transform it into a corresponding problem that does not require the AC property, and so can work using standard ProVerif resolution [8]. Their approach covers a surprising range of protocols, although, like [13], not Implicitly Authenticated Diffie-Hellman protocols such as MQV.

Another contrast between this paper and previous work is our uniform treatment of security goals (see Figs. 2–3). Our methods are applicable to confidentiality, authentication, and further properties such as forward secrecy.

Meadows and Pavlovic [35], cf. [11], do not explicitly represent the algebra. Instead, they offer a family of authentication axioms. Each axiom in the family expresses a limitation on the adversary by saying that some receptions can be only explained only by actions of regular principals. Such an axiom may be justified by a computational principle such as CDH. While this method leads to illuminating results, it appears to sidestep a foundational question about the algebraic structures in which these axioms are satisfied. our paper is a complementary attempt to fill in information about these models.

Our adversary model is active. For passive attacks, there has been some work on computational soundness for Diffie-Hellman, with Bresson et al. [10] giving an excellent treatment.

1 An Equational Theory of Messages

By *DH-structure* we mean a cyclic group G of prime order q, together with an exponentiation operator. The exponents E are integers modulo the prime q, which form a field of characteristic q. In cryptographic applications G is often taken to be a subgroup of the multiplicative group of integers modulo a prime p, where q divides $p-1$; sometimes G is a prime-order subgroup of the group of points over an elliptic curve.

Our challenge is to define an equational theory that captures the relevant algebra of DH structures, with a notion of reduction that supports modeling messages as normal forms. By the Decisional Diffie-Hellman assumption, an adversary *cannot* retrieve the exponent x from a value g^x that a regular participant has constructed. Our formalism reflects this limitation by not including a logarithm function in the signature of DH-structures.

Our strategy for handling the fact that the field of exponents in a DH structure cannot be axiomatized by equations is as follows. We work with a sort G for base-group elements and a sort E for exponents. The novelty is that we enrich E by adding a subsort NZE. Its intended interpretation is the non-0 elements of E, and it does not include 0 in any interpretation.

The device of approximating "non-zero" reflects a philosophy of capturing uniform capabilities algebraically. For instance no term which is a sum $e_1 + e_2$ is syntactically of sort NZE because each finite field has finite characteristic and so there are instantiations of the variables in $e_1 + e_2$ driving the term to 0. On the other hand, we will want to ensure that NZE is closed under multiplication; this is the role of the operator $**$ below.

We show in this section that $\mathsf{AG}\hat{}$ admits a confluent and terminating notion of reduction. In section 2 we prove Thm. 9 that describes the sense in which $\mathsf{AG}\hat{}$ captures the equalities that hold in almost all finite prime fields.

Definition 1. *The order-sorted signature $\Sigma(\mathsf{AG}\hat{})$ has the sorts G, E, and NZE, with NZE a subsort of E with operators:*

$$\begin{array}{rlll} \cdot : & G \times G \to G & id : \to G & inv : G \to G \\ +,\ -,\ * : & E \times E \to E & 0 : \to E & exp : G \times E \to E \\ i : & NZE \to NZE & 1 : \to NZE & ** : NZE \times NZE \to NZE \end{array}$$

and axioms (writing $exp(t, e)$ *as* t^e*):*

1. $(G, \cdot, inv, id)$ *is an abelian group;*
2. $(E, +, 0, -, *, \mathbf{1})$ *is a commutative ring with identity;*
3. *Exponentiation makes* G *a right* E*-module with identity, i.e.*
$$(a^x)^y = a^{x\,*\,y} \qquad a^1 = a \qquad id^x = id$$
$$(a \cdot b)^x = a^x \cdot b^x \qquad a^{(x+y)} = a^x \cdot a^y$$
4. *Multiplicative inverse, closure at sort* NZE*:*
$$u ** v = u\ *\ v \qquad u\ *\ i(u) = \mathbf{1} \qquad i(-u) = -i(u)$$
$$i(u * v) = i(u) * i(v) \qquad i(1) = 1 \qquad i(i(w)) = w$$

We extract an AC rewrite system from $\mathsf{AG}\hat{}$ by orienting the non-AC equations, using additional equations derivable from $\mathsf{AG}\hat{}$ to join critical pairs:

Definition 2. *Let* R *be the set of rewrite rules given by the natural orientation of the equations in Definition 1, other than associativity and commutativity, together with the additional rules presented in Table 1. The rewrite relation* $\rightarrow_{\mathsf{AG}\hat{}}$ *is rewriting with* R *modulo the associativity and commutativity equations.*

Theorem 3. *The reduction* $\rightarrow_{\mathsf{AG}\hat{}}$ *is terminating and confluent modulo AC.*

Proof. Termination can be established using the AC-recursive path order defined by Rubio [37] with a precedence in which exponentiation is greater than inverse, which is in turn greater than multiplication (and 1). This has been verified with the Aprove termination tool [23].

Then confluence follows from local confluence, which is established via a verification that all critical pairs are joinable. This result has been confirmed with the Maude Church-Rosser Checker [19]. □

Terms that are irreducible with respect to $\rightarrow_{\mathsf{AG}\hat{}}$ are called *normal forms*. The following taxonomy of the normal forms will be crucial in what follows, most of all in the definition of indicators, Definition 10. The proof is a routine simultaneous induction over the size of e and t. By G-variables and E-variables, we mean variables of those types.

At sort G	At sort E
$inv(id) \rightarrow id$	$-(0) \rightarrow 0$
$inv(a \cdot b) \rightarrow inv(a) \cdot inv(b)$	$-(x + y) \rightarrow -(x) + (-(y))$
$inv(inv(b)) \rightarrow b$	$-(-(x)) \rightarrow x$
$(inv(a))^x \rightarrow inv(a^x)$	$0 * x \rightarrow 0$
$a^0 \rightarrow id$	$-(x) * y \rightarrow -(x * y)$
$a^{-(x)} \rightarrow inv(a^x)$	

Table 1. Additional rewrite rules for $\rightarrow_{\mathsf{AG}\hat{}}$

Lemma 4. *1. If $e : E$ is a normal form then e is a sum $m_1 + \ldots + m_n$ where (i) each m_i is of the form $\pm(e_1 \ * \ \ldots \ * \ e_k)$ where $k \geq 0$, (ii) no e_i is of the form $i(e_j)$, and (iii) each e_i is one of x and $i(x)$, with x an E-variable. When $n = 0$, e is the ring element 0; when $k = 0$, m_i is the ring element 1. We call terms of the form $\pm m_i$* irreducible monomials*.*

2. If $t : G$ is a normal form then t is a product $t_1 \cdot \ldots \cdot t_n$, for $n \geq 0$ where (i) no t_i is of the form $inv(t_j)$, and (ii) each t_i is one of: v, $inv(v)$, v^e, $inv(v^e)$, with v a G-variable, and $e : E$ an irreducible monomial. When $n = 0$, $t = id$.

2 Uniform Equality and the Completeness of AGˆ

In this section we justify the use of AGˆ, specifically the use of AGˆ-normal forms to model messages. Since the axioms of AGˆ are clearly true in all DH-structures, any theorem of AGˆ holds in all DH-structures. Theorem 9 gives us a strong converse, namely that every equation that holds in infinitely many DH-structures is a theorem of AGˆ. If fact we show how to construct a single structure $\mathcal{M}_D$ that is "generic" for all DH-structures: An equation $s = t$ is holds in $\mathcal{M}_D$ if and only if it holds in infinitely many DH-structures.

Algebraically isomorphic DH-structures can have very different *computational* properties. Indeed, the prime field $\mathbb{F}_q$ presented as the group of integers mod q can be viewed as a DH-structure where the base group is the *additive* group of $\mathbb{F}_q$ and exponentiation is multiplication. The discrete log problem in this structure is computationally tractable. However, $\mathbb{F}_q$ is isomorphic to a subgroup of order q of the *multiplicative* group of integers modulo some prime p. There, the discrete log problem may be intractable. We focus on algebraic equations between terms in DH-structures; the absence of the log operator in our signature models the fact that our intended models are those in which discrete log is intractable.

First, we observe that the field of scalars, i.e. the exponents, carries all the algebraic information in a model of AGˆ.

Definition 5. *Let F be a field. We define the model $\mathcal{M}_F$ of theory AGˆ to be as follows. The sorts E and G are each interpreted as the domain of F; the sort NZE is interpreted as the set of non-0 elements of E. The operations of E are interpreted just as in F itself. The group operation $\cdot$ in G is taken to be $+$ from E, thus id and inv are taken to be 0 and $-$. Exponentiation is multiplication: a^e is interpreted as $a \ * \ e$.*

For each field F, $\mathcal{M}_F$ satisfies all of the equations in AGˆ. It is easy to check the following.

Lemma 6. *Every DH-structure is isomorphic to some $\mathcal{M}_{\mathbb{F}_q}$, where F is the prime field of order q.*

The key device for reasoning about uniform equality across DH-structures is the notion of *ultraproduct*, cf. e.g. [12]. We let the variable D range over non-principal ultrafilters over the set of prime numbers. The crucial facts about ultraproducts for our purposes are: (i) a first-order sentence is true in an ultraproduct if and only if the set of indices at which it is true is a set in D; (ii) every infinite set belongs to some non-principal ultrafilter; (iii) when D is non-principal, every set whose complement is finite is in D.

Definition 7. *Let D be a non-principal ultrafilter over the set of prime numbers and let $\mathbb{F}_D$ be the ultraproduct structure $\prod_D \{\mathbb{F}_q \mid q \text{ prime}\}$. $\mathcal{M}_{\mathbb{F}_D}$ is the DH structure obtained from $\mathbb{F}_D$ via Definition 5. For brevity we write $\mathcal{M}_D$ for $\mathcal{M}_{\mathbb{F}_D}$.*

$\mathbb{F}_D$ is a field, since each $\mathbb{F}_q$ satisfies the first-order axioms for fields, and has characteristic 0, since each equation $1 + \ldots + 1 = 0$ is false in all but finitely many $\mathbb{F}_q$.

When F is the additive group of rational numbers then $\mathcal{M}_F = \mathcal{M}_{\mathbb{Q}}$ is of special interest to us. The proof of the following lemma is in Appendix A.

Lemma 8. *1. The structure $\mathcal{M}_{\mathbb{Q}}$ can be embedded as a submodel in any $\mathcal{M}_D$.*
2. If s and t are distinct normal forms then it is not the case that $\mathcal{M}_{\mathbb{Q}} \models s = t$.

Our main result is that $\mathsf{AG}\hat{}$ is complete for uniform equality, in the following sense:

Theorem 9. *For each pair of G-terms s and t, the following are equivalent*

1. $\mathsf{AG}\hat{} \vdash s = t$
2. *For all q, $\mathcal{M}_{\mathbb{F}_q} \models s = t$*
3. *For all non-principal D, $\mathcal{M}_D \models s = t$*
4. *For infinitely many q, $\mathcal{M}_{\mathbb{F}_q} \models s = t$*
5. *For some non-principal D, $\mathcal{M}_D \models s = t$*
6. $\mathcal{M}_{\mathbb{Q}} \models s = t$
7. *If s reduces to s' and t reduces to t', with s', t' irreducible, then s' and t' are identical modulo associativity and commutativity of $\cdot$, $+$, and $*$.*

Proof. It suffices to establish the cycle of entailments 1 implies 2 ... implies 7 implies 1. The first four of these steps are immediate, as is the fact that 7 implies 1. The fact that 5 implies 6 follows from Lemma 8, item 1. To conclude 7 from 6, use Lemma 8, item 2. □

The results of Theorem 9 hold as well for equations between E-terms. Given terms e and e', form the equation $g^e = g^{e'}$. It is provable iff $e = e'$ is provable, and is true in a given model $\mathcal{M}$ iff $e = e'$ is.

The model $\mathcal{M}_{\mathbb{Q}}$ is convenient: this single model, based on a familiar structure, witnesses uniform equality faithfully. The models $\mathcal{M}_D$ satisfy another striking property. It follows from results of Ax [4] that a first-order sentence in the language of rings/fields is true in a given $\mathcal{M}_D$ if and only if it is true in all but a finite set of finite fields. Moreover this theory is decidable. So the structures $\mathcal{M}_D$ are attractive for closer study of the "uniform" properties of DH-structures.

3 Indicators

We turn now to a formal definition of indicators and the proof of a key invariant that all adversary actions preserve. For intuition about the following definition, think of N as being a set of secret values in a protocol run (such as A's x) not transmitted by any participant (although a related value such as g^x may be transmitted). Say that a monomial m is a *maximal-monomial* of t if t has a subterm of the form b^m.

Definition 10 (Indicators). *Let $N = \langle v_1, \ldots, v_d \rangle$ be a vector of NZE-variables. If m is an irreducible monomial, the* N-vector *for m is $\langle z_1, \ldots, z_k \rangle$ where z_i is the multiplicity of v_i in m, counting occurrences of $i(v_i)$ negatively.*

An E-term $e = m_1 + \ldots + m_k$ is N-free *if each m_i has N-vector $\langle 0, \ldots, 0 \rangle$.*

If t is irreducible, then $\mathrm{Ind}_N(t)$ is the set of all vectors $\boldsymbol{z}$ such that $\boldsymbol{z}$ is the N-vector of m, where m is a maximal-monomial subterm of t.

Example: For $N = \langle x, y \rangle$, $\mathrm{Ind}_N(\, g^{x\ i(y)} \cdot g^{zxy} \cdot g^{xx}\,) = \{\langle 1, -1 \rangle, \langle 1, 1 \rangle, \langle 2, 0 \rangle\}$.

If e is N-free, then $\mathrm{Ind}_N(t^e) = \mathrm{Ind}_N(t)$, because no new occurrences of N-variables are created in passing from t to t^e.

Definition 11. *Let $T = \{t_1, \ldots, t_k\}$ be a set of terms. The set* $\mathrm{Gen}(T)$ generated *by T is the least set of terms including T and closed under applications of function symbols.*

Functions cannot cancel to reveal a $v_i \in N$, which leads to our main theorem.

Theorem 12 (Indicator Theorem). *Let N be a vector of NZE-variables and let T be a set of terms where each $e : E \in T$ is N-free. Then*

1. *every $e \in \mathrm{Gen}(T)$ of sort E is N-free, and*
2. *if $u \in \mathrm{Gen}(T)$ is of sort G and $\boldsymbol{z} \in \mathrm{Ind}_N(u)$, then for some $t \in T$, $\boldsymbol{z} \in \mathrm{Ind}_N(t)$.*

Proof. By induction on operations used to construct terms from elements of T.

The main cases are for 2., when (i) $u = u_1 \cdot u_2$ or (ii) $u = t^e$, where t, u_1, u_2 and e are irreducible terms in $\mathrm{Gen}(T)$. First, if $u = u_1 \cdot u_2$, then u is a product $t_1 \cdot \ldots \cdot t_n$, and each factor t_i is of the form $v, \mathit{inv}(v), v^e$, or $\mathit{inv}(v^e)$ and comes from u_1 or u_2. Thus, the normal form of this term results by canceling any pair of factors, one from u_1 and one from u_2 that are inverses of each other. No new E-subterms are created, so no new indicator vectors are created, and our assertion holds.

Otherwise $u = t^e$. Since e is in $\mathrm{Gen}(T)$, we know inductively that e is N-free. It suffices to show that $\mathrm{Ind}_N(t^e) = \mathrm{Ind}_N(t)$. Letting t be in normal form, t^e is $(t_1)^e \cdot \ldots \cdot (t_n)^e$. However, as we just observed, $\mathrm{Ind}_N(t_i^e) = \mathrm{Ind}_N(t_i)$. ◻

This "conservation of indicators" principle essentially restricts adversary behavior; Theorem 15 below makes this precise in the strand-space setting.

4 Strands and Indicators

We will now adapt the strand space theory [25,38] to the case where the messages include a free algebra over $\mathsf{AG}\hat{}$. A *strand* is a sequence of local actions called *nodes*, each of which is:

- a message *transmission*, written $\bullet \rightarrow$;
- a message *reception*, written $\bullet \leftarrow$; or
- a *neutral* node $\circ$. Neutral nodes are local events in which a principal consults or updates its local state [26].

If n is a node, and the message t is transmitted, received, or coordinated with the state on n, then we write $t = \mathsf{msg}(n)$. We sometimes write $+t = \mathsf{msg}(n)$ and $-t = \mathsf{msg}(n)$ when n is respectively a transmission or reception node. Double arrows indicate successive events on the same strand, e.g. $\circ \Rightarrow \bullet \Rightarrow \bullet$.

A *protocol Π* is a set of strands, called the *roles* of the protocol. We assume every protocol contains a specific role, called the *listener* role, consisting of a single reception node $n = \rightarrow \bullet$. Listener strands provide "witnesses" when $\mathsf{msg}(n)$ has been disclosed, aiding in specifying confidentiality properties. A *regular* strand for Π means an instance of one of the roles of Π.

Adversary strands consist of zero or more reception nodes followed by one transmission node. The adversary obtains the transmitted value as a function of the values received; or creates it, if there are no reception nodes. All values that the adversary handles are received or transmitted; none are silently obtained from long-term state. Allowing the adversary to use neutral nodes—or strands of other forms—provides no additional power. (See Defn. 13.)

Messages. The messages transmitted and received on $\bullet$ nodes, and obtained from long-term state on neutral nodes $\circ$, form an abstract algebra. The message algebra MA includes as basic values:

- Elements of the free algebra over $\mathsf{AG}\hat{}$ built from the infinite sets of E-variables $\mathcal{V}^E$ and G-variables $\mathcal{V}^G$; we denote this algebra by $\mathsf{Free}(\mathsf{AG}\hat{})$,
- Disjoint infinite sets of *names*, *symmetric* and *asymmetric keys*, and *texts*.

The elements of the algebra $\mathsf{Free}(\mathsf{AG}\hat{})$ are equivalence classes of terms. However, the results in Section 1 say that each class has a canonical representative, namely an AC normal form modulo $\rightarrow_{\mathsf{AG}\hat{}}$. This justifies a syntactic approach, particularly in our treatment of indicators in Thm. 15.

We assume that some of the asymmetric keys are of the form $\mathsf{pk}(A)$ and $\mathsf{vk}(A)$, where A ranges over names, denoting the public encryption and signature verification key of A. We also assume that asymmetric keys are equipped with an inverse operation; for instance, $\mathsf{pk}(A)^{-1}$ is A's private decryption key.

The *parameters* of an $\mathsf{AG}\hat{}$ normal form are the $\mathcal{V}^E$ and $\mathcal{V}^G$ variables occurring in it. The parameter of a value $\mathsf{pk}(A)$ or $\mathsf{vk}(A)$ is A. For all other basic values a, the parameter of a is a. MA is closed under the constructors:

- Pairing, where the pair of t_1 and t_2 is written $t_0 \parallel t_1$;

– Encryption, where the encryption of t_0 using t_1 as key is written $\{\!|t_0|\!\}_{t_1}$.

As constructors, the operations are free, yielding equal results only when the arguments are equal: $\{\!|t_0|\!\}_{t_1} = \{\!|t_2|\!\}_{t_3}$ implies $t_0 = t_2$ and $t_1 = t_3$, etc. We regard hashes and digital signatures as coded using (deterministic) encryption: the hash $\mathsf{h}(t) = \{\!|t|\!\}_{K_0}$, where K_0 is an asymmetric encryption key to which no one knows the inverse. We will always assume that K_0^{-1} is uncompromised. The digital signature $[\![\, t_0 \,]\!]_{t_1}$ can be encoded as $t_0 \parallel \{\!|t_0|\!\}_{t_1}$.

The parameters of a pair, encryption, digital signature, or hash are the union of the parameters of its immediate subterms.

A parameter represents a "degree of freedom" in describing executions, which can be instantiated or restricted. It may also represent an independent choice, as A's choice of a group element x to build g^x is independent of B's choice of y.

Ingredients and origination. A value t_1 is an *ingredient* of another value t_2, written $t_1 \sqsubseteq t_2$, if t_1 contributes to t_2 via concatenation or as the plaintext of encryptions: $\sqsubseteq$ is the least reflexive, transitive relation such that:

$$t_1 \sqsubseteq t_1 \parallel t_2, \qquad t_2 \sqsubseteq t_1 \parallel t_2, \qquad t_1 \sqsubseteq \{\!|t_1|\!\}_{t_2}.$$

By this definition, $t_2 \sqsubseteq \{\!|t_1|\!\}_{t_2}$ implies that (anomalously) $t_2 \sqsubseteq t_1$. For basic values a, b, we have $a \sqsubseteq b$ iff $a = b$. Thus, the ingredient relation is much coarser than the "occurs in" relation.

A value t *originates* on a transmission node n if $t \sqsubseteq \mathsf{msg}(n)$, so that it is an ingredient of the message sent on n, but it was not an ingredient of any message earlier on the same strand. That is, $m \Rightarrow^+ n$ implies $t \not\sqsubseteq \mathsf{msg}(m)$.

A basic value is *uniquely originating* in a bundle $\mathcal{B}$ if there is exactly one $n \in \mathsf{node}(\mathcal{B})$ at which it originates. Freshly chosen nonces or DH values g^x are typically assumed to be uniquely originating. A basic value is *non-originating* if there is no $n \in \mathsf{node}(\mathcal{B})$ at which it originates. An uncompromised long term secret (e.g. a private decryption key) is assumed to be non-originating. Because adversary strands receive their arguments as incoming messages, an adversary strand that decrypts a message receives its key as a message, which must originate somewhere. The set of non-originating values is denoted non; the set of uniquely originating values is denoted unique.

In DH protocols unique origination and non-origination are used in tandem. When a compliant principal generates a random x and transmits g^x, the former will be non-originating and the latter uniquely originating. A probabilistic implementation of the (non-probabilistic) unique- and non-origination randomly chooses values from large sets, with overwhelming probability of faithfulness.

Adversary model The adversary strands are defined:

Definition 13. *1. A strand $+a$, having one transmission node, is an adversary strand if a is a parameter or a constant $id, 1, 0$.*

2. A strand $-t \Rightarrow +f(t)$, having a reception node and a transmission node, is an adversary strand if f is any of the unary functions $inv, i, -, \mathsf{pk}, \mathsf{sk}, \mathsf{h}$.

3. A strand $-t_1 \Rightarrow -t_2 \Rightarrow +g(t_1, t_2)$, having two reception nodes and a transmission node, is an adversary strand if g is any of the binary functions $\cdot, *, +, \cdot \parallel \cdot, \{\!|\cdot|\!\}_\cdot, [\![\cdot]\!]_\cdot$.
4. A strand $-\{\!|t_1|\!\}_K \Rightarrow -K^{-1} \Rightarrow +t_1$ is an adversary strand.

Importantly, there is no adversary strand executing the asymmetric key inverse function K^{-1}, nor any logarithm operation.

This adversary model suggests a game between adversary and system:

1. The system chooses a security goal Φ, involving secrecy, authentication, key compromise, etc., as in Figs. 2–3.
2. The adversary proposes a potential counterexample $\mathbb{A}$ consisting of regular strands with equations between values on the nodes, e.g. an equation between session keys as computed by two participants.
3. For each message reception node in $\mathbb{A}$, the adversary chooses a recipe, intended to produce an acceptable message, using the strands of Def. 13. The adversary may use earlier transmission events on regular strands to build messages for subsequent reception events.
   These recipes determine a set of equalities between the values computed by the adversary and the values t "expected" by the recipient (i.e. acceptable to the recipient). They are the *adversary's proposed equations*.
4. The adversary wins if his proposed equations are valid in $\mathcal{M}_{\mathbb{F}_q}$, for infinitely many primes q; or equivalently, by Theorem 9, valid for all primes q.

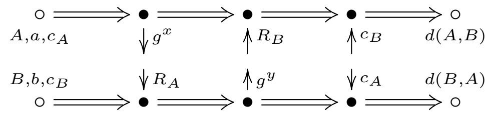
This game may seem too challenging for the adversary. First, it wins only if the equations are valid, i.e. true for all instances of the variables. Second, the adversary must choose how to generate all the messages, its adversary strategy, before seeing any concrete bitstrings, or indeed learning the prime q.

These objections motivate work on *computational soundness*. The hardness of DDH suggests that, when an equation is not valid, it is hard to obtain a satisfying instance. Moreover, the adversary should acquire no advantage from seeing the values g^x etc. However, precise results will require reduction arguments.

Executions are bundles. We formalize protocol executions by *bundles*. A bundle is a directed, acyclic graph. Its vertices are nodes on some strands (which may include both regular and adversary strands). Its edges include the succession edges $n_1 \Rightarrow n_2$, as well as *communication edges* written $n_1 \rightarrow n_2$. Such a dag $\mathcal{B} = (V, E_\Rightarrow \cup E_\rightarrow)$ is a *bundle* if it is causally self-contained, meaning:

- If $n_2 \in V$ and $n_1 \Rightarrow n_2$, then $n_1 \in V$ and $(n_1, n_2) \in E_\Rightarrow$;
- If $n_2 \in V$ is a reception node, then there is a unique transmission node $n_1 \in V$ such that $\mathsf{msg}(n_2) = \mathsf{msg}(n_1)$ and $(n_1, n_2) \in E_\rightarrow$;
- Precedence $\preceq_\mathcal{B}$ for $\mathcal{B}$, defined to be $(E_\Rightarrow \cup E_\rightarrow)^*$, is a well-founded relation.

Indicators and the adversary. We justify now our central technique, that the adversary cannot generate messages with new indicators. We will write $\mathbf{0}$ for the all zero vector, i.e. the origin. We will also write $\mathbf{1}_v$ for the v^{th} basis vector $\langle \ldots, 0, \ldots, 1, \ldots, 0, \ldots \rangle$.



$c_P = [\![\, \mathsf{cert}\ Y_P \parallel P \,]\!]_{\mathsf{sk(CA)}}$ $\qquad$ $d(P,P') = \mathsf{keyrec}\ P \parallel P' \parallel K$

Fig. 1. UM Initiator and Responder Strands

Definition 14. *Let N be a vector of NZE-variables. If a is a name, symmetric key, asymmetric key, or text, then its indicator set* $\mathrm{Ind}_N(a) = \{\mathbf{0}\}$*, the singleton of the origin.* $\mathrm{Ind}_N(t_0 \parallel t_1) = \mathrm{Ind}_N(t_0) \cup \mathrm{Ind}_N(t_1)$.

$$\mathrm{Ind}_N(\{\!|t_0|\!\}_{t_1}) = \mathrm{Ind}_N([\![\, t_0 \,]\!]_{t_1}) = \mathrm{Ind}_N(\mathsf{h}(t_0)) = \mathrm{Ind}_N(t_0).$$

A basic value a is non-originating before *n in bundle $\mathcal{B}$ if, for all $n' \preceq_{\mathcal{B}} n$, a does not originate at n'. The* indicator basis *$\mathrm{IB}_{\mathcal{B}}(n)$ of node n, where n is a node of $\mathcal{B}$, is the set (ordered in some conventional way):*

$$\{a \in \mathsf{Params}(\mathcal{B}) \colon a \textit{ of sort } E \textit{ is non-originating before } n\}.$$

Theorem 15 (Indicator Theorem for Strands). *Let n be an adversary transmission node of $\mathcal{B}$, and let N be a sequence of elements drawn from $\mathrm{IB}_{\mathcal{B}}(n)$. If $v \in \mathrm{Ind}_N(\mathsf{msg}(n))$ and $v \neq \mathbf{0}$, then there is a regular transmission node $n' \prec_{\mathcal{B}} n$ in $\mathcal{B}$ such that $v \in \mathrm{Ind}_N(\mathsf{msg}(n'))$.*

Proof. Let T_R be the set of messages transmitted on a regular node $m \prec n$, and let T_M be the set of parameters and constants transmitted on one-node adversary strands $\prec n$. By induction on adversary actions, $\mathsf{msg}(n) \in \mathrm{Gen}(T_R \cup T_M)$. T_R and T_M are N-free, by the definition of IB. So Theorem 12 applies.

Since $t : G \in T_M$ implies $\mathrm{Ind}_N(t) = \{\mathbf{0}\}$, we conclude that every non-zero indicator in u comes from a message in T_R, as desired. □

5 Analyzing the Unified Model

Regular participants in the UM protocol [2] act as *initiators* and *responders* as shown in Figure 1. We specify, for the initiator A:

1. A retrieves from its secure storage its principal name A, its long term secret a, and its public certificate c_A.
2. A chooses an ephemeral parameter $z \in \mathcal{V}^E$ to instantiate x, sending $R_A = g^z$.
3. A receives some R_B, which it checks to be a non-trivial group element, i.e. a value of the form g^y for some $y \neq 0, 1 \bmod q$.
4. It receives a certificate c_B associating Y_B with B's identity. How the participant determines what name B to require in this certificate, or how it determines which CAs to accept, is implementation-dependent.

5. A computes $K = \mathsf{h}({Y_B}^a \parallel {R_B}^z)$, depositing a *key record* into its local database, so that K may be used as a session key between A and B.

In clause 2, A chooses z freshly. A never sends z as an ingredient in any message, only g^z, and the adversary cannot find a strategy to guess the same value z, we model z as non-originating, and g^z as uniquely originating. In other Implicitly Authenticated Diffie-Hellman protocols, other key computations may be used instead of Eqn. 2. A responder B behaves correspondingly. The syntax of Fig. 1 entails that no regular node n ever transmits a product $t_1 \cdot t_2$ as a (normal form) ingredient of any message, $t_1 \cdot t_2 \not\sqsubseteq \mathsf{msg}(n)$.

Regular initiator and responder strands that choose that parameters x, y transmit only messages g^x, g^y, where

$$\mathrm{Ind}_{\langle a,b,x,y\rangle} g^x = \{\mathbf{1}_x\} \text{ and } \mathrm{Ind}_{\langle a,b,x,y\rangle} g^y = \{\mathbf{1}_y\}.$$

Strands with other choices transmit the zero vector $\mathbf{0}$ relative to this x, y basis. In case 2, $\mathrm{Ind}_{\langle a,b,x,y\rangle}(Y) = \{\mathbf{1}_a\}$. However, the key K has indicators

$$\mathrm{Ind}_{\langle a,b,x,y\rangle} = \{\langle 1,1,0,0\rangle, \langle 0,0,1,1\rangle\}.$$

Here, the regular principals transmit only messages with basis vectors $\mathbf{1}_v$ or $\mathbf{0}$ as indicators, but the key has two non-zero entries in its two indicators.

Cryptographically, DH ensures that the choices of the principals always contribute in a non-cancellable way to the result. An analogue is:

Lemma 16 (Contributive Parameters). *Let $\mathcal{B}$ be a* UM*-bundle, and s be an initiator or responder strand with long term secret a and ephemeral value x:*

1. *If $x \in \mathsf{non}_{\mathcal{B}}$, then for $K = \mathsf{h}({Y_B}^a \parallel {R_B}^x)$, we have $\mathbf{1}_x \in \mathrm{Ind}_{\langle x\rangle}(K)$.*
2. *If $a \in \mathsf{non}_{\mathcal{B}}$, then $\mathbf{1}_a \in \mathrm{Ind}_{\langle a\rangle}(K)$.*

Proof. Since $\mathsf{h}(\cdot)$ and $\parallel$ are constructors, a or x can cancel only if s receives a value R_B or Y_b with indicator $\langle -1\rangle$ for a or x, resp. Hence there is some earlier node m on which some message with indicator $\langle -1\rangle$ was transmitted, and let m_0 be a minimal such node.

However, by the definitions, m_0 is not a regular node, which transmit only values with non-negative indicators. By Thm. 15, m_0 cannot be an adversary node either, when $x \in \mathsf{non}_{\mathcal{B}}$ or $a \in \mathsf{non}_{\mathcal{B}}$ resp. $\square$

Key Secrecy and Impersonation. In Fig. 2 we present the core idea of key secrecy. Suppose that the upper strand s is an initiator or responder run that ends by computing session key K. Moreover, suppose that a listener strand is present, which receives K. Then, if the long term secrets $a, b \in \mathsf{non}$, this diagram cannot be completed to a bundle $\mathcal{B}$. This holds even without the freshness assumptions on regular initiator and responder strands. It includes bundles in which we add any number of regular strands, so long as these particular long-term secrets $a, b \in \mathsf{non}$. Other principals' long term keys may be freely compromised or not.

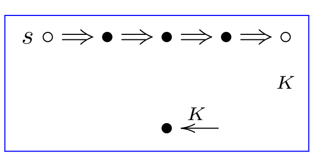


Fig. 2. Key secrecy: This diagram cannot occur

Security Goal 17 (Key Secrecy) *Suppose $\mathcal{B}$ is a bundle with $a, b \in \mathsf{non}_{\mathcal{B}}$, and s is an initiator or responder strand with long term secret parameter a and long term peer public value $Y = g^b$. Then $\mathcal{B}$ does not contain a listener $\bullet \leftarrow K$.*

Theorem 18. UM *achieves the security goal of key secrecy.*

Proof. Suppose instead that $\bullet \leftarrow K$ is in $\mathcal{B}$, so some node transmits K.

Computing indicators using the basis $\langle a, b\rangle$ by applying Lemma 16 to both a and b, K has indicator $\langle 1, 1\rangle$. By Thm. 15, some regular node transmits a message with indicator $\langle 1, 1\rangle$. But regular strands transmit only values with indicators $\mathbf{0}$ and, in certificates, $\mathbf{1}_a$, $\mathbf{1}_b$, relative to basis $\langle a, b\rangle$. $\square$

Curiously, resistance to impersonation attacks concerns the same diagram, Fig. 2, although with different assumptions. An impersonation attack is a case in which the adversary, having compromised B's long term secret b, uses it to obtain a session key K, while causing B to have a session yielding K as session key. If B's session uses $Y_A = g^a$, where a is the uncompromised long term secret of A, then the adversary has succeeded in *impersonating* A to B. By contrast, it is hopeless—when b is compromised—to try to prevent the adversary from impersonating B to others.

Security Goal 19 (Impersonation Resistance) *Suppose $\mathcal{B}$ is a bundle with $a, x \in \mathsf{non}_{\mathcal{B}}$, and s is an initiator or responder strand with long term secret parameter a ephemeral value x. Then $\mathcal{B}$ does not contain a listener $\bullet \leftarrow K$.*

This goal trades off a long term secret for an ephemeral value. UM does not achieve it. Its key $K = \mathsf{h}(g^{ab} \parallel g^{xy})$ has indicators $\{\langle 1, 0\rangle, \langle 0, 1\rangle\}$ in the basis $\langle a, x\rangle$, suggested by our assumptions. Thus, Theorem 15 buys us nothing.

Example 20 *The adversary can impersonate A to B by supplying its own g^z, as B supplies g^y; it computes $K = \mathsf{h}(g^{ab} \parallel g^{zy})$ by raising A's public g^a to the compromised value b, and raising g^y to its own ephemeral value z.*

Implicit Authentication. Implicit authentication takes two forms [7,27,32].

The essential common idea is expressed in Figure 3. It shows two strands that compute the same session key K. One has parameters $[A, B', \ldots]$ and the other has parameters $[A', B, \ldots]$, where we assume that the parameter for the initiator's name appears first (A, A') and parameter for the responder's name appears second (B', B). The authentication property is that the participants agree on each other's identities, so that the responder has the correct opinion about the initiator's identity and *vice versa*. That is, we want $A = A'$ and $B = B'$ whenever the computed keys agree. Stronger and weaker implicit key authentication properties differ in what non-compromise assumptions they make. The stronger property is that

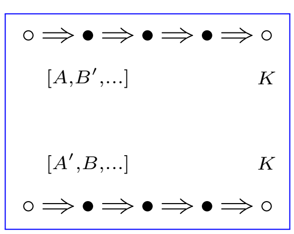


Fig. 3. Implicit authentication: In this diagram, $A = A'$ and $B = B'$

$A = A'$ and $B = B'$ whenever $a, b \in \mathsf{non}$. A weaker assertion is that $A = A'$ and $B = B'$ whenever $a, b, a' \in \mathsf{non}$. The additional non-compromise assumption is about a', the long term secret of the principal E that B *thinks* he is communicating with [7,18,32]. MQV satisfies only this weaker form [27]. We focus on the stronger property here.

Authentication depends on the certification protocol, which ensures proof of possession. Rather than representing it, we characterize it by an assumption:

Assumption 21 *If $c_P \sqsubseteq \mathsf{msg}(n)$ for $n \in \mathsf{node}(\mathcal{B})$, then $c_P = [\![\, \mathsf{cert}\ g^e \parallel P \,]\!]_{\mathsf{sk(CA)}}$ for some E-value $e \neq 0, 1$, and either:*

1. *there exists $n \in \mathcal{B}$ with $e \sqsubseteq \mathsf{msg}(n)$, or else*
2. *(i) $e \in \mathcal{V}^E$ is a parameter, and*
   (ii) if $[\![\, \mathsf{cert}\ g^e \parallel P' \,]\!]_{\mathsf{sk(CA)}} \sqsubseteq \mathsf{msg}(n')$ for any $n' \in \mathsf{node}(\mathcal{B})$, then $P = P'$.

Clause (1) holds when e is generated by the adversary; clause (2) applies when e is chosen by a compliant principal.

Security Goal 22 (Implicit Authentication) *Suppose that $\mathcal{B}$ is a Π-bundle with $a, b \in \mathsf{non}_{\mathcal{B}}$, and strands s_1, s_2 are Π initiator and responder strands with parameters $[A, B', a, x, Y_{B'}, R_{B'}]$ and $[A', B, b, y, Y_{A'}, R_{A'}]$ resp. If s_1, s_2 both yield session key K, then $A = A'$ and $B = B'$.*

Theorem 23. UM *achieves implicit authentication.*

Proof. Let s_1, s_2 be strands in $\mathcal{B}$ as in the implicit authentication goal, where also $a, b \in \mathsf{non}_{\mathcal{B}}$. Since s_1 receives a certificate $[\![\, \mathsf{cert}\ Y_{B'} \parallel B' \,]\!]_{\mathsf{sk(CA)}}$, by Assumption 21, $Y_{B'} = g^e$ for some $e \neq 0, 1$. By symmetry, $Y_{A'} = g^d$.

The key computation ensures $g^{db} = g^{ae}$; by injectiveness, $db = ae$. Thus, there is some c such that $d = ca$ and $e = cb$. Thus, by Assumption 21 either:

1. there exists $n_d \in \mathsf{node}(\mathcal{B})$ such that $cb \sqsubseteq \mathsf{msg}(n_d)$, or else
2. cb's normal form is a parameter, i.e. $c = 1$ and $e = b$.

In the latter case, we also have that $B' = B$. In the former case, n_d lies on an adversary strand. It must result from multiplying the values b and c, since no regular strand transmits a message with any product as an ingredient. But this contradicts $b \in \mathsf{non}(\mathcal{B})$. Symmetrically, $A' = A$. □

Future work. We will apply these methods to more challenging protocols [18]. We will also study their computational soundness. A tool implementation approach is to represent AGˆ and protocols using it in geometric logic; model-finding can generate counterexamples or establish their absence. An alternative approach is integration with Tamarin [15]. AGˆ appears to extend to represent bilinear pairings.

Acknowledgments. We have benefited from discussions with Shriram Krishnamurthi, Moses Liskov, Cathy Meadows, Paliath Narendran, John Ramsdell, Paul Rowe, Paul Timmel, and Ed Zieglar.

References

1. R.M. Amadio, D. Lugiez, and V. Vanackère. On the symbolic reduction of processes with cryptographic functions. *Theoretical Computer Science*, 290(1):695–740, 2003.
2. R. Ankney, D. Johnson, and M. Matyas. The Unified Model. contribution to ANSI X9F1. *Standards Projects (Financial Crypto Tools), ANSI X*, 42, 1995.
3. A. Armando, D. A. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. Hankes Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron. The AVISPA tool for the automated validation of internet security protocols and applications. In Kousha Etessami and Sriram K. Rajamani, editors, *CAV*, volume 3576 of *Lecture Notes in Computer Science*, pages 281–285. Springer, 2005.
4. James Ax. The elementary theory of finite fields. *The Annals of Mathematics*, 88(2):pp. 239–271, 1968.
5. David A. Basin, Sebastian Mödersheim, and Luca Viganò. OFMC: A symbolic model checker for security protocols. *Int. J. Inf. Sec.*, 4(3):181–208, 2005.
6. Jan A. Bergstra and J. V. Tucker. The rational numbers as an abstract data type. *Journal of The ACM*, 54, 2007.
7. Simon Blake-Wilson and Alfred Menezes. Authenticated Diffe-Hellman key agreement protocols. In *Selected Areas in Cryptography*, pages 630–630. Springer, 1999.
8. Bruno Blanchet. An efficient protocol verifier based on Prolog rules. In *14th Computer Security Foundations Workshop*, pages 82–96. IEEE CS Press, June 2001.
9. M. Boreale and M.G. Buscemi. Symbolic analysis of crypto-protocols based on modular exponentiation. *Mathematical Foundations of Computer Science 2003*, pages 269–278, 2003.
10. Emmanuel Bresson, Yassine Lakhnech, Laurent Mazaré, and Bogdan Warinschi. Computational soundness: The case of Diffie-Hellman keys. In Veronique Cortier and Steve Kremer, editors, *Formal Models and Techniques for Analyzing Security Protocols*, Cryptology and Information Security Series. IOS Press, 2011.
11. I. Cervesato, C. Meadows, and D. Pavlovic. An encapsulated authentication logic for reasoning about key distribution protocols. In *Computer Security Foundations, 2005. CSFW-18 2005. 18th IEEE Workshop*, pages 48–61. IEEE, 2005.
12. C.C. Chang and H.J. Keisler. Model Theory, volume 73 of Studies in Logic and the Foundations of Mathematics, 1990.
13. Yannick Chevalier, Ralf Küsters, Michaël Rusinowitch, and Mathieu Turuani. Deciding the security of protocols with Diffie-Hellman exponentiation and products in exponents. *FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science*, pages 124–135, 2003.
14. Cas Cremers and Michele Feltz. One-round strongly secure key exchange with perfect forward secrecy and deniability. Cryptology ePrint Archive, Report 2011/300, 2011. http://eprint.iacr.org/2011/300.
15. Cas Cremers, Benedikt Schmidt, Simon Meier, and David Basin. Automated analysis of Diffie-Hellman protocols and advanced security properties. In *Computer Security Foundations (CSF)*, 2012.
16. C.J.F. Cremers. *Scyther - Semantics and Verification of Security Protocols*. Ph.D. dissertation, Eindhoven University of Technology, 2006.
17. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

18. Daniel J. Dougherty and Joshua D. Guttman. Symbolic protocol analysis for Diffie-Hellman. *Arxiv preprint arXiv:1202.2168*, 2012. At http://arxiv.org/abs/1202.2168v1.
19. Francisco Durán and José Meseguer. A Church-Rosser checker tool for conditional order-sorted equational Maude specifications. In Peter Csaba Ölveczky, editor, *WRLA*, volume 6381 of *Lecture Notes in Computer Science*, pages 69–85. Springer, 2010. Version 3j, available at http://maude.lcc.uma.es/CRChC.
20. Santiago Escobar, Catherine Meadows, and José Meseguer. State space reduction in the Maude-NRL protocol analyzer. *Computer Security-ESORICS 2008*, pages 548–562, 2008.
21. Santiago Escobar, Catherine Meadows, and José Meseguer. Maude-NPA: Cryptographic protocol analysis modulo equational properties. *Foundations of Security Analysis and Design V*, pages 1–50, 2009.
22. Marcelo Fiore and Martín Abadi. Computing symbolic models for verifying cryptographic protocols. In *Computer Security Foundations Workshop*, June 2001.
23. J. Giesl, P. Schneider-Kamp, and R. Thiemann. Aprove 1.2: Automatic termination proofs in the dependency pair framework. In *Proceedings IJCAR '06*, LNAI 4130, pages 281–286. Springer, 2006.
24. Jean Goubault-Larrecq, Muriel Roger, and Kumar Verma. Abstraction and resolution modulo AC: How to verify Diffie-Hellman-like protocols automatically. *Journal of Logic and Algebraic Programming*, 64(2):219–251, 2005.
25. Joshua D. Guttman. Shapes: Surveying crypto protocol runs. In Veronique Cortier and Steve Kremer, editors, *Formal Models and Techniques for Analyzing Security Protocols*, Cryptology and Information Security Series. IOS Press, 2011.
26. Joshua D. Guttman. State and progress in strand spaces: Proving fair exchange. *Journal of Automated Reasoning*, 2012. Accepted, March 2010. DOI: 10.1007/s10817-010-9202-1.
27. Burton S. Kaliski. An unknown key-share attack on the MQV key agreement protocol. *ACM Transactions on Information and System Security*, 4(3):275–288, 2001.
28. Deepak Kapur, Paliath Narendran, and Lida Wang. An E-unification algorithm for analyzing protocols that use modular exponentiation. *RewritingTechniques and Applications*, pages 150–150, 2003.
29. H. Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol. In *Advances in Cryptology–CRYPTO 2005*, pages 546–566. Springer, 2005.
30. Sebastian Kunz-Jacques and David Pointcheval. About the Security of MTI/C0 and MQV. *Security and Cryptography for Networks*, pages 156–172, 2006.
31. Ralf Küsters and Tomasz Truderung. Using ProVerif to analyze protocols with Diffie-Hellman exponentiation. In *IEEE Computer Security Foundations Symposium*, pages 157–171. IEEE, 2009.
32. L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone. An efficient protocol for authenticated key agreement. *Designs, Codes and Cryptography*, 28(2):119–134, 2003.
33. Alfred Menezes. Another look at HMQV. *Journal of Mathematical Cryptology*, 1:47–64, 2007.
34. Jonathan K. Millen and Vitaly Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *8th ACM Conference on Computer and Communications Security (CCS '01)*, pages 166–175. ACM, 2001.
35. D. Pavlovic and C. Meadows. Deriving secrecy in key establishment protocols. *Computer Security–ESORICS 2006*, pages 384–403, 2006.

36. John D. Ramsdell and Joshua D. Guttman. CPSA: A cryptographic protocol shapes analyzer. In *Hackage*. The MITRE Corporation, 2009. http://hackage.haskell.org/package/cpsa; see esp. `doc` subdirectory.
37. Albert Rubio. A fully syntactic AC-RPO. In Paliath Narendran and Michaël Rusinowitch, editors, *RTA*, volume 1631 of *Lecture Notes in Computer Science*, pages 133–147. Springer, 1999.
38. F. Javier Thayer, Jonathan C. Herzog, and Joshua D. Guttman. Strand spaces: Proving security protocols correct. *Journal of Computer Security*, 7(2/3):191–230, 1999.

A Appendix

Lemma 8

1. *The structure $\mathcal{M}_{\mathbb{Q}}$ can be embedded as a submodel in any $\mathcal{M}_D$.*
2. *If s and t are distinct normal forms then it is not the case that $\mathcal{M}_{\mathbb{Q}} \models s = t$.*

1. Since $\mathbb{F}_D$ has characteristic 0, and $\mathbb{Q}$ is the prime field of characteristic 0, $\mathbb{Q}$ is embeddable in $\mathbb{F}_D$. The models $\mathcal{M}_D$ and $\mathcal{M}_{\mathbb{Q}}$ are definitional expansions of $\mathbb{F}_D$ and $\mathbb{Q}$, so the embedding of $\mathbb{Q}$ into $\mathbb{F}_D$ extends to embed $\mathcal{M}_{\mathbb{Q}}$ into $\mathcal{M}_D$.
2. If s and t are distinct normal forms, the term $u \equiv s \cdot inv(t)$ is in normal form and not identically id. With this observation we see that our result follows if we establish the following fact: if u is a normal form not identically id then it is not the case that $\mathcal{M}_{\mathbb{Q}} \models u = id$.
   To see this, note that in the structure $\mathcal{M}_{\mathbb{Q}}$, the group operation is interpreted as addition, inverse by additive inverse, and exponentiation as multiplication, so it suffices to consider the expression obtained from u by replacing $\cdot$ and inv by $+$ and $-$, and the exponentiation operator by $*$. In this way we may view u as an ordinary rational expression in the variables $x_1, \ldots, x_k$ occurring in u. So u determines a real function $f_u : \mathbb{R}^k \to \mathbb{R}$ not identically 0. We can find a rational point $\boldsymbol{r} = (r_1, ..., r_k)$ such that $f_u(\boldsymbol{r}) \neq 0$. Then the environment $\eta : Vars \to \mathbb{Q}$ with $\eta(x_i) = r_i$ witnesses the fact that $\mathcal{M}_{\mathbb{Q}} \not\models u = id$. $\square$