Symbolic Protocol Analysis for Diffie-Hellman

Daniel J. Dougherty Joshua D. Guttman*

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

Abstract. We extend symbolic protocol analysis to apply to 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 resisting previous symbolic approaches.

We work in an algebra defined by the normal forms of a rewriting theory (modulo associativity and commutativity). These 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 several different protocols that use Diffie-Hellman operators in subtle ways.

We also give a model-theoretic justification of our rewriting theory: the theory proves all equations that are uniformly true as the order of the cyclic group varies.

1 Introduction

Despite vigorous research in symbolic methods for cryptographic protocol analysis, many gaps and limitations remain. While systems such as NPA-Maude [17], ProVerif [6], CPSA [33], and Scyther [12] are extremely useful, great ingenuity is still needed—as for instance in [29]—to analyze protocols that use fundamental cryptographic ideas such as Diffie-Hellman key agreement (henceforth, DH). Moreover, important types of protocols, such as implicitly authenticated key-agreement, appear to be out of reach of known symbolic techniques. Indeed, for these protocols, computational techniques have also led to considerable controversy, with arduous proofs that provide little confidence [25,27,28,31].

In this paper we present foundational results and a new analysis technique that together expand the range of applicability of symbolic analysis. In preparation for stating our contributions we remind the reader of the basics of the Diffie-Hellman key exchange [13]. In the protocol's original form, the principals A, B agree on a suitable prime p, and a generator $1 < g < p$ such that the powers of g form a cyclic group of some large prime order q. For a particular session, A

* We gratefully acknowledge support by the National Science Foundation under grant CNS-0952287. Version of February 9, 2012.

and B choose random values x, y respectively, raising a base g to these powers mod p:

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

They can then both compute the value $(g^y)^x = g^{xy} = (g^x)^y$ (modulo p, as we will no longer explicitly repeat). We can regard g^{xy} as a new shared secret for A, B. This is reasonable because of the Decisional Diffie-Hellman assumption (DDH), which is the assumption that g^{xy} is indistinguishable from the g^z we would get from a randomly chosen z, for any observer who was given neither x nor y. The protocol is thus secure against a passive adversary, who observes what the compliant principals do, but can neither create messages nor alter (or misdirect) messages of compliant principals.

However, an active adversary can choose its own x', y', sending $g^{y'}$ to A instead of g^y, and sending $g^{x'}$ to B instead of g^x. Now, each of A, B actually shares one key with the adversary, who can act as a man in the middle, re-encrypting messages in any conversation between A and B. Various protocols have been proposed to achieve a range of security goals in the presence of an active attacker, such as *implicit authentication*, *forward secrecy*, and *preventing impersonation attacks*. In Section 2 we describe some of these protocols.

The algebra of the structures on which DH protocols operate has been an obstacle to analyzing them These structures are cyclic groups of some prime order q, together with an exponentiation operator. The exponents are integers modulo the prime q, which form a field of characteristic q. We will call such structures *DH-structures*. The algebraic richness of DH-structures has resisted full symbolic formalization, despite substantial steps for subalgebras [17, 26, 29].

In this paper, we make five contributions.

1. We represent security goals as logical formulas about transmission and reception events, together with freshness and non-compromise assumptions. These clean, structural definitions are easy to work with, in contrast with the procedural notations prevalent among cryptographers. They are based on strand spaces [21, 35] as a model of protocol execution.
2. We give a new treatment of the values used for DH exchanges. These values are characterized by a set of equations, namely the equations $s = t$ that are valid in infinitely many DH-structures. In fact, we prove that if an equation holds in infinitely many DH-structures then it holds in all of them.
   a. Using an ultraproduct construction, we build a *single* model $\mathcal{M}_D$ that realizes precisely those equations true in all (equivalently, infinitely many) DH-structures. (Theorem 20)
   b. We define an equational theory $\mathsf{AG}\hat{}$ that can be presented by a rewrite system that is terminating and confluent modulo associativity and commutativity (Theorem 1). Furthermore, for all equations $s = t$, $\mathsf{AG}\hat{}$ rewrites s and t to the same normal form if and only if $s = t$ is true in $\mathcal{M}_D$ for all values of its free variables (Theorem 20). The normal forms of this rewrite system represent the messages.

3. The theory of DH-structures suggests an adversary model (Section 5). The *uniformly algebraic adversary* is the Dolev-Yao adversary augmented with the functions in the signature of DH-structures. These functions are governed by the equations $s = t$ derivable in $\mathsf{AG}\hat{}$. Thus—given the correspondence between $\mathsf{AG}\hat{}$ and truth in DH-structures—the adversary can rely on any equation that, as the size of the underlying cyclic group grows, is valid infinitely often.
4. Using the $\mathsf{AG}\hat{}$ normal forms, we define the *indicators* of a message. Indicators count occurrences of secret values in exponents. We prove that the adversary cannot create a message with a new indicator. If the adversary transmits a message with a particular indicator, then it must have received some message with that indicator previously (Theorem 5). This invariant extends the Honest Ideal theorem [35] to the algebra $\mathsf{AG}\hat{}$. It is our primary proof method.
5. To illustrate the power of our method, we prove about a dozen different security goals for three protocols (Sections 7 and 8). These implicitly authenticated DH protocols have previously resisted attempts to give concise, convincing proofs of the goals they achieve. We also use our method to show why certain protocols do not meet some goals, matching the relevant attacks from the cryptographic literature.

The set of indicators of a message is a set of vectors that count how many times uncompromised values appear in exponents. They are a refinement of the standard notion of an atom occurring in term, needed since our terms are considered modulo equations. For instance, suppose that in some execution, the exponents a, b, x, y are assumed uncompromised, where x, y are ephemeral secret values and a, b are long-term secret values. The sequence $\langle a, b, x, y\rangle$ determines a *basis* for writing these indicator vectors.

Relative to this basis, the factor g^{xy} has indicator $\langle 0, 0, 1, 1\rangle$ because a, b appear 0 times each, and x, y appear once each. $g^{x/y}$ would have indicator $\langle 0, 0, 1, -1\rangle$, since y appears -1 times, i.e. inverted. The factor g^{ax} has indicator $\langle 1, 0, 1, 0\rangle$ since a, x appear once. When we multiply factors, we take unions of indicators. Thus, $g^{xy}g^{bx}g^{ay}g^{ab}$ has indicators

$$\{\langle 0, 0, 1, 1\rangle, \langle 0, 1, 1, 0\rangle, \langle 1, 0, 0, 1\rangle, \langle 1, 1, 0, 0\rangle\}.$$

There is good motivation for protocols in which each non-zero integer in an indicator is ± 1 [8].

In our model, when the indicator basis consists of uncompromised exponents, adversary actions never produce any message containing any new indicator (Theorem 5). If the adversary *transmits* a message with some indicator vector $\boldsymbol{v}$, then it previously *received* some message with that indicator vector $\boldsymbol{v}$. Only the regular, non-adversary, participants can emit messages with new indicators.

This idea, which is natural and appealing for DH, is challenging to justify, which is probably why it is not familiar from the cryptographic literature. Its soundness as a proof technique rests on our foundational results concerning DH-structures (contribution 2).

Structure of this paper. We next, in Sec. 2, introduce a few protocols we will use as running examples. Sec. 3 introduces the strand space theory, and the Sec. 4 presents our equational theory $\mathsf{AG}\hat{}$. We use strand spaces in Sec. 5 to define IADH protocols and the adversary actions. Section 6 proves the key limitative theorem on indicators and the adversary. Sec. 7 defines a variety of security goals for IADH protocols, and applies the key limitative result to establish these goals; the focus shifts specifically to implicit authentication in Sec. 8. Sec. 9 takes a model-theoretic point of view on DH-structures and proves completeness of the theory $\mathsf{AG}\hat{}$. In Sec. 10, we comment on some related work and conclude.

2 Some Protocols of Interest

We start by describing some illustrative protocols at the level of detail typically seen in the literature. In order not to prejudice ourselves in evaluating possible attacks, we will write R_B for the public value that A receives, purportedly from B, rather than writing g^y, since no one yet knows whether it is the same value that B sent. We likewise write R_A for the public value that B receives, purportedly from A. The participants hope that $R_A = g^x$ and $R_B = g^y$.

The Station-to-Station protocol [14] authenticates the Diffie-Hellman exchange by digital signatures on the exchange. In a simplified STS, the exchange in Eqn. 1 is followed by the signed messages:

$$A \qquad \bullet \xrightarrow{[\![\, g^x \| R_B \,]\!]_{\mathsf{sk}(A)}} \qquad \xleftarrow{[\![\, g^y \| R_A \,]\!]_{\mathsf{sk}(B)}} \bullet \qquad B \tag{2}$$

The signatures[1] exclude a man in the middle, assuming some public key infrastructure to certify $\mathsf{sk}(A), \mathsf{sk}(B)$. The costs of STS includes an additional message transmission and reception for each participant, in each session. Moreover, each participant must also prepare one digital signature and also verify one digital signature specifically for that session. There is also a privacy concern, since the signatures publicly associate A and B in a shared session.

An alternative to using per-session digital signatures is *implicit authentication* [5]. Here the goal is to ensure that any principal that can compute the same value as A *can only* be B, and conversely. To implement this idea, each principal maintains a long-term secret, which we will write as a for principal A, and as b for B; they publish the long-term public values g^a, g^b, which we will refer to as Y_A, Y_B, etc. The trick is to build the use of the private values a, b into the computation of the shared secret, so that only A, B can do it. In the "Unified Model" UM of Ankney, Johnson, and Matyas [2], the principals combine long term values with short term values by concatenating and hashing. They send only the messages shown in Eqn. 1, and then—letting $H(x)$ be a hash of x—compute their keys:

$$A: \; k = H({Y_B}^a \parallel {R_B}^x) \quad B: \; k = H({Y_A}^b \parallel {R_A}^y), \tag{3}$$

[1] We write $t \parallel t'$ for the concatenation of t with t'. A digitally signed message $[\![\, t \,]\!]_{\mathsf{sk}(A)}$ means $t \parallel \mathsf{sig}(H(t), \mathsf{sk}(A))$, where sig is a signature algorithm, $H(t)$ is a hash of t, and $\mathsf{sk}(A)$ is a signing key owned by A.

obtaining the shared value $H(g^{ab} \parallel g^{xy})$ if $R_A = g^x$ and $R_B = g^y$. Again, public key infrastructure must associate the public value Y_P with the intended peer P. However, no digital signature is generated or checked specific to this run. If A has frequent sessions with B, A can amortize the cost of the certificate verification by keeping Y_B in secure storage.

Menezes-Qu-Vanstone (MQV) [30] relies only on algebraic operations. MQV computes the key via the rules:

$$A:\ k = (R_B \cdot {Y_B}^{[R_B]})^{s_A} \quad B:\ k = (R_A \cdot {Y_A}^{[R_A]})^{s_B} \tag{4}$$

where $s_A = x + a[R_A]$ and $s_B = y + b[R_B]$. The "box" operator coerces numbers mod p to a convenient form in which they can be used as exponents. In the literature this is written in the typographically more cumbersome form of a bar, as $\overline{R_B}$. In a successful run, A obtains the value

$$(g^y \cdot (g^b)^{[g^y]})^{s_A} = (g^{(y+b[g^y])})^{(x+a[g^x])} = g^{(s_B \cdot s_A)} \tag{5}$$

and B obtains $g^{s_A \cdot s_B}$, which is the same value. MQV differs from UM only in the function that the principals use to compute the key. MQV's key computation makes it algebraically challenging to model and to analyze. Controversy about its security remains [25, 27, 28, 31].

3 Background: Strand Spaces

In this paper, we adopt the strand space formalism, although allowing messages to form more complex algebraic structures than in earlier papers, e.g. [21, 35].

Strands. A *strand* is a sequence of local actions called *nodes*. A node may be either:

- a message *transmission*;
- a message *reception*; or else
- a *neutral* node. Neutral nodes are local events in which a principal consults or updates its local state [22].

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 write bullets $\bullet$ for transmission and reception events and circles $\circ$ for neutral events, involving only the local state. Double arrows indicate successive events on the same strand, e.g. $\circ \Rightarrow \bullet \Rightarrow \bullet$.

Each strand is either a *regular strand*, which represents the sequence of local actions made be a single principal in a single local session of a protocol, or else an *adversary strand*, which represents a single action of the adversary.

A *protocol* is a set of regular strands, called the *roles* of the protocol. We assume that every protocol contains a specific role, called the *listener* role, which consists of a single reception node $n = \!\!\rightarrow \bullet$. We use listener strands to provide "witnesses" when $\mathsf{msg}(n)$ has been disclosed, especially to specify confidentiality properties.

Adversary strands consist of zero or more reception nodes followed by one transmission node. They represent the adversary obtaining the transmitted value as a function of the values received; or creating 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. In fact, allowing the adversary to use neutral nodes—or strands of other forms—provides no additional power. (See Section 6.)

We regard the messages transmitted and received on $\bullet$ nodes, and obtained from long-term state on neutral nodes $\circ$, as forming an abstract algebra. *Concatenation* and *encryption* are operators that construct values in the algebra from a pair of given values, and we regard $v_0 \parallel v_1$ as equal to $u_0 \parallel u_1$ just in case $v_0 = u_0$ and $v_1 = u_1$. Similarly, $\{\!|v_0|\!\}_{v_1}$ equals $\{\!|u_0|\!\}_{u_1}$ just in case $v_0 = u_0$ and $v_1 = u_1$. That is, they are *free* operators. For our present purposes, it suffices to represent other operators such as hash functions and digital signatures in terms of these.

The *basic* values that are neither concatenations nor encryptions include principal names; keys of various kinds; group elements x, $x \cdot y$, and g^x; and text values. We regard variables ("indeterminates") such as x as values distinct from values of other forms, e.g. products $z \cdot y$, or from other variables. A variable represents a "degree of freedom" in a description of some 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. DH algebras are defined later in this section as the normal forms of an AC rewriting system.

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 (non-encrypted, non-concatenated) values a, b, we have $a \sqsubseteq b$ iff $a = b$.

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 an execution if there is exactly one node at which it originates. Freshly chosen nonces or DH values g^x are typically assumed to be uniquely originating.

A value is *non-originating* if there is no node at which it originates. An uncompromised long term secret such as a signature key or a private decryption key is assumed to be non-originating. Because adversary strands receive their arguments as incoming messages, an adversary strand that encrypts a message receives its key as a message, thus originating somewhere. Decryption and signature creation are similar.

The set of non-originating values is denoted non; the set of uniquely originating values is denoted unique.

Very often in DH-style 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.

Executions are bundles. The strand space theory formalizes 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 strand 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 $(n_1, n_2) \in E_{\rightarrow}$;
- The precedence ordering $\preceq_{\mathcal{B}}$ for $\mathcal{B}$, defined to be $(E_{\Rightarrow} \cup E_{\rightarrow})^*$, is a well-founded relation.

The first clause says that a node has a causal explanation from the occurrence of the earlier nodes on its strand. The second says that any reception has the causal explanation that the message was obtained from some particular transmission node. The last clause says that causality is globally well-founded. It holds automatically in *finite* dags $\mathcal{B}$, which are the only ones we consider here.

When we assume that a value is non-originating, or uniquely originating, we constrain which bundles $\mathcal{B}$ are of interest to us, namely those in which the value originates on no node of $\mathcal{B}$, or on one node of $\mathcal{B}$, respectively.

4 An Equational Theory of Messages

As described in the Introduction, our challenge is to define an equational theory that captures the relevant algebra of DH structures and admits 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. This limitation is reflected in our formalism in a straightforward way. Namely, we do not provide a logarithm function in the signature of DH-structures.

In addition we must confront the fact that the exponents in a DH structure form a field, and fields cannot be axiomatized by equations.

Our strategy 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 whose intended interpretation is the non-0 elements of E.

The device of expressing "non-zero" as a sort fits well with the philosophy of capturing uniform capabilities algebraically. For instance no term which is a sum e_1+e_2 will inhabit the sort NZE because each finite field has finite characteristic and so there may be 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.

Sorts: G, E, and NZE, with NZE a subsort of E;	
$\cdot : G \times G \to G$	$+,\ * : E \times E \to E$
$1 :\to G$	$1 :\to NZE$
$inv : G \to G$	$i : NZE \to NZE$
$exp : : G \times E \to E$	
$box : G \to NZE$	$** : NZE \to NZE$

Table 1. The signature for $\mathsf{AG}\hat{}$

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

Definition 1. *The theory $\mathsf{AG}\hat{}$ is the equational theory comprising the sorts and operation given in Table 1 and the equations given in Table 2. We write $box(t)$ as $[t]$, and we write $exp(t, e)$ and t^e.*

We next construct an associative-commutative rewrite system from $\mathsf{AG}\hat{}$. We orient each equation in Table 2 in the left-to-write direction, except for the associativity and commutativity of $\cdot, +$, and $*$. Confluence requires the new rules shown in Table 3, corresponding to equations derivable from $\mathsf{AG}\hat{}$ that are needed to join critical pairs.

Definition 2. *Let R be the set of rewrite rules given in Table 2—read from left to right, but without associativity and commutativity—and in Table 3. The rewrite relation $\to_{\mathsf{AG}\hat{}}$ is rewriting with R modulo associativity and commutativity of $\cdot, +$, and $*$.*

Theorem 1. *The reduction $\to_{\mathsf{AG}\hat{}}$ is terminating and confluent modulo AC.*

Proof. Termination can be established using the AC-recursive path order defined by Rubio [34] 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 [19].

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 [15].

Terms that are irreducible with respect to $\to_{\mathsf{AG}\hat{}}$ 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 4. The proof is a routine simultaneous induction over the size of e and t.

	$(E, +, 0, -, \ast, \mathbf{1}, i)$ is a commutative unitary ring
$(G, \cdot, inv, 1)$ is an abelian group	$(x+y)+z = x+(y+z)$
$(a \cdot b) \cdot c = a \cdot (b \cdot c)$	$x+y = y+x$
$a \cdot b = b \cdot a$	$x+0 = x$
$b \cdot 1 = b$	$x+(-x) = 0$
$b \cdot inv(b) = 1$	$(x \ast y) \ast z = x \ast (y \ast z)$
	$x \ast y = y \ast x$
	$x \ast (y+z) = (x \ast y)+(z+z)$
	$x \ast 1 = x$
Multiplicative inverse, closure at sort NZE	Exponentiation makes G a unitary right E-module
$u \ast\ast v = u \ast v$	$(a^x)^y = a^{x \ast y}$
$i(u \ast v) = i(u) \ast i(v)$	$a^1 = a$
$i(1) = 1$	$(a \cdot b)^x = a^x \cdot b^x$
$i(i(w)) = w$	$a^{(x+y)} = a^x \cdot a^y$
	$1^x = 1$

Table 2. The theory AGˆ

Lemma 1. *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 $e_1 \ast \ldots \ast e_k \quad k \geq 0$, (ii) no e_i is of the form $i(e_j)$, and (iii) each e_i is one of:*

$$x, \quad i(x), \quad [t], \quad i([t])$$

with x a G-variable and $t : G$ a G-normal form.
The case $n = 0$ is taken to mean $e = 0$; the case $k = 0$ is taken to mean $m_i = 1$ We call terms of the form m_i irreducible monomials

2. If $t : G$ is a normal form then t is a product $t_1 \cdot \ldots \cdot t_n, \quad n \geq 0$ where (i) no t_i is of the form $inv(t_j)$, and (ii) each t_i is one of:

$$v \qquad inv(v) \qquad v^e \qquad inv(v^e)$$

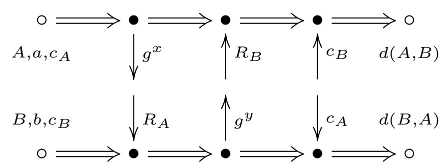
with v a G-variable $e : E$ an irreducible monomial.
The case $n = 0$ is taken to mean $t = 1$.

5 Formalizing the Protocols and the Adversary

We consider a collection of protocols that all involve the same strands, i.e. sequences of transmissions, receptions, and neutral events. They differ almost exclusively in the key computations used to generate the shared secret.

At sort G		At sort E	
$inv(1)$	$\rightarrow 1$	$-(0)$	$\rightarrow 0$
$inv(a \cdot b)$	$\rightarrow inv(a) \cdot inv(b)$	$-(x+y)$	$\rightarrow -(x) + (-(y))$
$inv(inv(b))$	$\rightarrow b$	$-(-(x))$	$\rightarrow x$
$(inv(a))^x$	$\rightarrow inv(a^x)$	$0 * x$	$\rightarrow 0$
a^0	$\rightarrow 1$	$-(x) * y$	$\rightarrow -(x * y)$
$a^{-(x)}$	$\rightarrow inv(a^x)$		

Table 3. Additional rewrite rules for $\rightarrow_{\mathsf{AG\hat{}}}$



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

Fig. 1. IADH Initiator and Responder Strands

MQV and UM [5] both fit our pattern. Various other protocols fit this pattern with some cajoling. KEA [5] fits the pattern too, although its key computation uses addition mod p to combine g^{ay} and g^{bx}. Cremers-Feltz's protocol CF [11], in which the shared secret is $g^{(x+a)(y+b)}$, almost fits: They use the digitally signed messages $[\![\, R_A \,]\!]_{\mathsf{sk}(A)}$ and $[\![\, R_B \,]\!]_{\mathsf{sk}(B)}$. Our analysis is equally applicable in this case.

In these protocol descriptions, we make explicit aspects that are normally left implicit. One is the interaction with the certifying authority. Kaliski [25] argues that the certification protocol should be considered in analysis, because the correctness of forms of a protocol may depend on exactly what checks a CA has actually made. We will also show how the local session interacts with the local principal state.

The IADH initiator and responder roles. We summarize the activities of regular initiators and responders in Figure 1. We specify, for the initiator A:

1. A retrieves its principal name A, its long term secret a, and its public certificate c_A from its secure storage.
2. A chooses a fresh ephemeral x, transmitting $R_A = g^x$.
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. We do not specify here how the participant determines what name B to require in this certificate, or how it determines which CAs to accept. This is implementation-dependent.
5. Finally, A performs the protocol-specific key computation to determine K. A checks the exponentiations yield non-1 values, and fails if any do. On success, A deposits a *key record* into its local state database, so that K may be used for a secure conversation between A and B.

In clause 2, A chooses x freshly. Because A never sends x as an ingredient in any message—but only g^x—it follows that x has negligible probability of occurring in a message. After all, A does not send it; any other regular participant is overwhelmingly unlikely to choose the same value again; and the adversary is overwhelmingly unlikely to choose it, e.g. as a guess. For this reason we model x as being "non-originating"; for the same reason, g^x is declared to be uniquely originating.

We always add the assumptions that x is non-originating and g^x is uniquely originating whenever a *regular* strand selects $R_A = g^x$. In particular, since x is a fresh, unconstrained choice that the principal makes, we always instantiate it with a simple value, essentially a *parameter*, and never with a compound expression like $y \cdot 1/z$. Essentially, x is a generator of the algebra of normal forms of $\mathsf{AG}\hat{}$.

A responder B behaves in a corresponding fashion, with predictable changes to the names of its parameters. The only real change is that it receives an ephemeral public value R_A in step 2 before generating its ephemeral secret y and transmitting its ephemeral public value g^y in step 3. We will assume that y is non-originating and g^y is uniquely originating whenever a regular responder strand selects $R_B = g^y$.

The parameters to an initiator strand are A, B, a, x, Y_B, R_B. We write them in this order, and refer (e.g.) to the fourth parameter as x, despite the fact that in different instances of the role have different choices for the parameter x. The parameters to a responder strand are A, B, b, y, Y_A, R_A; thus, we will write the (purported) initiator's name first, and the (actual, known) responder's name second.

We make an assumption on the principal states, namely that the node $\circ\, A, a, c_A$ starting an initiator or responder strand is possible only if the same principal A has on some earlier occasion received a certificate c_A, and deposited it into its state. Certificates do not emerge *ex nihilo*. Gathering our assumptions on regular initiator and responder strands:

Assumption 2 *Suppose that $\mathcal{B}$ is a bundle.*

1. *If $\mathcal{B}$ contains an initiator strand s with parameters A, B, a, x, Y_B, R_B, then:*
   a. *x is non-originating, and g^x is uniquely originating.*
   b. *x is a parameter, not a compound expression.*
   c. *For some transmission node $n \in \mathcal{B}$, $c_A \sqsubseteq \mathsf{msg}(n)$ and $n \preceq_{\mathcal{B}} s_1$, where s_1 is the first node on strand s.*

2. *Symmetrically for responder strands s in $\mathcal{B}$, with parameters A, B, b, y, Y_A, R_A:*
   a. *y is non-originating, and g^y is uniquely originating.*
   b. *y is a parameter, not a compound expression.*
   c. *For some transmission node $n \in \mathcal{B}$, $c_B \sqsubseteq \mathsf{msg}(n)$ and $n \preceq_{\mathcal{B}} s_1$, where s_1 is the first node on strand s.*

Our results do not depend on the specific ordering of events in initiator and responder strands. As long as the neutral node retrieving the long term secret and certificate occurs before any of the other events, and as long as the neutral node depositing the K into the state occurs only after the other event, then our results remain correct. They also do not distinguish between initiator and responder strands: We would allow two initiator strands to succeed in implicit authentication, for instance.

Key computation functions. The shared secret K is generated using different functions in different IADH protocols. In the Unified Model UM, the key is generated by

$$A:\ k = H({Y_B}^a \parallel {R_B}^x) \quad B:\ k = H({Y_A}^b \parallel {R_A}^y) \tag{6}$$

In the optimistic case that $R_A = g^x$ and $R_B = g^y$

$$K = H(g^{ab} \parallel g^{xy}) \tag{7}$$

In MQV the key is generated by

$$A:\ K = (R_B \cdot {Y_B}^{[R_B]})^{s_A} \quad B:\ K = (R_A \cdot {Y_A}^{[R_A]})^{s_B} \tag{8}$$

(where $s_A = (x + a[g^x])$ and $s_B = (y + b[g^y])$), so when $R_A = g^x$ and $R_B = g^y$ the principals compute

$$(g^{(y+b[g^y])})^{(x+a[g^x])} = g^{xy} \cdot g^{xb[g^y]} \cdot g^{ya[g^x]} \cdot g^{ab[g^x][g^y]} \tag{9}$$

The key computation for Cremers-Feltz CF—somewhat simplified to make it more parallel to the UM and MQV computations—is:

$$A:\ K = (R_B \cdot Y_B)^{(a+x)} \quad B:\ K = (R_A \cdot Y_A)^{(b+y)} \tag{10}$$

so that, in the same optimistic case,

$$(g^y \cdot g^b)^{(x+a)} = g^{(y+b)(x+a)} = g^{xy} \cdot g^{xb} \cdot g^{ya} \cdot g^{ab} \tag{11}$$

The occurrences of a, b, x, y in these terms show us a contrast between UM and the other two. In the latter, all four pairs consisting of one parameter from a, x and one from b, y, appearing together, may be found in the exponent of some factor of the final shared secret. However, in UM, only two of these pairs appears. This suggests that UM is more fragile than the latter two, and this explains why it is vulnerable to key compromise impersonation while the others are not.

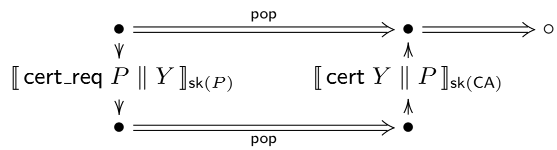


Fig. 2. Strands for Certificate Requests

In Section 9, we will develop an algebraic theory to justify this kind of analysis.

Requesting and issuing certificates. We also identify protocol roles for requesting certificates from certificate authorities, and for the CAs to issue them (Fig. 2). The client makes a request with its name P and public value Y_P, and, if successful, receives a certificate which it deposits into its local state. In its request, a compliant principal named P always chooses a fresh long-term secret a, and computes $Y = g^a$. The CA, on receiving a request, issues a certificate after a "proof of possession" protocol pop intended to show P possesses an a such that $g^a = Y$. We will not make pop explicit.

We assume, whenever a bundle contains a regular certificate request, that its $g^a = Y$ is uniquely originating. Any subsequent use of Y must obtain it through some sequence of message transmissions and receptions tracing back, ultimately, to this originating node. We will not, however, always assume a is non-originating, since carelessness or malice may eventually lead to the disclosure of a. Instead, if a particular a is non-originating, we will explicitly state that as a hypothesis in the security goals that depend on it.

We assume the CA is uncompromised, i.e. $\mathsf{sk}(\mathsf{CA}) \in \mathsf{non}$. CA, when receiving Y, should ensure that $Y \neq g^0, g^1$, and that it is a member of the group (e.g. via the little Fermat test). Hence, there is an e such that $Y = g^e$.

Moreover, a successful pop means that the requester possesses an exponent e such that $Y = g^e$. The requester is either a regular participant or the adversary. Thus, either:

– e is some parameter a, and g^a originates uniquely on a certificate request strand; or else
– the request comes from an adversary strand, and e is available to the adversary.

We will model the latter by assuming that the bundle containing this certification generation strand also contains a listener strand $n \Rightarrow \bullet$ with $\mathsf{msg}(n) = e$.

Assumption 3 *Let $\mathcal{B}$ be a bundle containing $[\![\mathsf{cert}\ Y \parallel P]\!]_{\mathsf{sk}(\mathsf{CA})}$. Assume $\mathsf{sk}(\mathsf{CA}) \in \mathsf{non}_{\mathcal{B}}$, and moreover:*

1. *For a certificate request strand, with parameters P, a, CA:*

    a. g^a *originates uniquely;*
    b. a *is a parameter, not a compound expression.*
2. *For a* CA *strand, with parameters* P, Y, CA*:*
    a. *There exists an* $e \neq 0, 1$ *such that* $Y = g^e$*;*
    b. *Either* $Y = g^a$ *where* $g^a \in \mathsf{unique}_{\mathcal{B}}$*, and* g^a *originates on a regular certificate request strand, or else there exists* $n \in \mathcal{B}$ *with* $\mathsf{msg}(n) = e$*.*

By Clause 2b, if $a \in \mathsf{non}_{\mathcal{B}}$, then for at most one P can a certificate $[\![\, \mathsf{cert}\ g^a \parallel P \,]\!]_{\mathsf{sk(CA)}}$ be issued. The IADH protocols are defined by the four roles shown in Figs. 1–2, using a key computation such as those in MQV, UM, and CF.

The Adversary. An adversary strand has zero or more reception nodes followed by a transmission node:

Definition 3. Adversary strands *take the forms:*

– *Emission of a basic value* a*:* $\langle +a \rangle$
– *Constructor strands:* $\langle -a_1 \Rightarrow \ldots \Rightarrow -a_n \Rightarrow +t \rangle$ *where is* t *is in* $\mathrm{Gen}(a_1, \ldots, a_n)$
– *Destructor strands:* $\langle -t \Rightarrow +s_1 \ldots \Rightarrow +s_n \rangle$ *where* t *is a concatenation of the values* s_i*.*
– *Encryption strands:* $\langle -K \Rightarrow -t \Rightarrow +\{\!|t|\!\}_K \rangle$
– *Decryption strands:* $\langle -K^{-1} \Rightarrow -\{\!|t|\!\}_K \Rightarrow +t \rangle$

Suppose that $S_1, \ldots, S_k$ *are node-disjoint adversary strands. An* adversary web *[21] using* $S_1, \ldots, S_k$ *is an acyclic graph whose vertices are the nodes of the* S_i*, where for each edge* (n, n')*, either (i)* $n \Rightarrow n'$ *on some strand or (ii)* n *is a transmission node,* n' *is a reception node, and* $\mathsf{msg}(n) = \mathsf{msg}(n')$*.*

This adversary model motivates a game between the adversary and the system:

1. The system chooses a security goal Φ, involving secrecy, authentication, key compromise, etc., as in Figs. 3–6.
2. The adversary chooses a potential counterexample $\mathbb{A}$ consisting of regular strands with equations between values on the nodes, e.g. an equation between a session key computed by one participant and a session key computed by another participant.
3. To show that $\mathbb{A}$ can occur, the adversary chooses how to generate the messages in $\mathbb{A}$.
   For each message reception node in $\mathbb{A}$, the adversary must provide an acceptable message in time for that event. The adversary benefits from transmission events on regular strands, which he can use to build messages for subsequent reception events. For each reception node, the adversary chooses a recipe, consisting of an adversary web, using the strands of Def. 3.
   This map—which, to every message reception event, associates an adversary web—is the *adversary strategy.*
   The adversary strategy determines a set of equalities between a value computed by the adversary and a value t "expected" by the recipient, or acceptable to the recipient. They are the *adversary's proposed equations.*

4. The adversary wins if his proposed equations are valid in (G_q, F_q), for infinitely many 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. However, the adversary's proposed equations determine polynomials, and each of these polynomials has a syntactically determined degree d. If it is not valid, it can have at most d solutions, independent of the choice of (G_q, F_q). Hence, the set of values for which the adversary's strategy works remains small, regardless of how the cardinality of the structure (G_q, F_q) grows.

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. This objection motivates future research into the *computational soundness* of our approach. The hardness of DDH seems to suggest that the adversary acquires no useful advantage from seeing the values g^x etc. Any definite claim would require a reduction argument.

6 Indicators

We turn now to a formal definition of indicators and the proof of a key invariant that all adversary actions preserve.

Let $\mathbb{Z}^k$ denote the set of all k-tuples of integers. For intuition about the following definition, think of N as being a set of *non-originating values* for a bundle. If m is a monomial occurring as a subterm of a term t, say that m is "maximal-monomial" if t has a subterm of the form b^m.

Definition 4 (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.*

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

When t_0 is any base term in normal form, then $\mathrm{Ind}_N(t_0)$ 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_0.

If $t = t_1 \parallel t_2$, then $\mathrm{Ind}_N(t) = \mathrm{Ind}_N(t_1) \cup \mathrm{Ind}_N(t_2)$.

If $t = \{|t_1|\}_{t_2}$, then $\mathrm{Ind}_N(t) = \mathrm{Ind}_N(t_1)$.

Thus, $\mathrm{Ind}_N(t)$ for a compound term t is the union $\bigcup \mathrm{Ind}_N(t_0)$, taking the union over all the base terms t_0 that are ingredients of t, i.e. $t_0 \sqsubseteq t$.

Example: For $N = \langle x, y \rangle$, if t is

$$g^{x\ i(y)} \cdot g^{zx[g^x]} \cdot g^{xx[g^y]},$$

then $\mathrm{Ind}_N(t) = \{\langle 1, -1 \rangle, \langle 1, 0 \rangle, \langle 2, 0 \rangle\}$. The boxed values do not contribute to the indicators.

Since we often encounter indicators with no non-zero entries, we will write $\mathbf{0}$ for this indicator $\langle 0, \ldots, 0 \rangle$. We will also write $\mathbf{1}_x$, $\mathbf{1}_a$, etc., for the indicator that has a single 1 in the position for that parameter, e.g. for $\langle 0,0,1,0\rangle$ and $\langle 1,0,0,0\rangle$ if the parameters are a, b, x, y in that order. Every message sent in IADH protocols is of this form: All the indicator weight is concentrated in at most a single 1. A message g^c with $c \notin \mathsf{non}$ has indicator $\mathbf{0}$.

Since the union $\bigcup \mathrm{Ind}_N(t_0)$ is over all the *ingredients* $t_0 \sqsubseteq t$, it does not include values used only as keys in encryptions. Thus, a protocol may compute a secret such as g^{xy} with an indicator $\langle 0,0,1,1\rangle = \mathbf{1}_x + \mathbf{1}_y$, and then applies a key derivation function, obtaining $k = \mathsf{kdf}(g^{xy})$. If participants then send encrypted messages $\{\!|t_1|\!\}_k$, then it has not transmitted a message with indicator $\mathbf{1}_x + \mathbf{1}_y$.

Definition 5. *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 the term-forming operations.*

The term-forming operations cannot cancel to reveal a $v_i \in N$:

Theorem 4. *Suppose T is a collection of terms such that every $e \in T$ of sort E 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 $\boldsymbol{z} \in \mathrm{Ind}(u)$ then for some $t \in T$, $\quad \boldsymbol{z} \in \mathrm{Ind}(t)$.*

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

The interesting cases are when u is of the form $u_1 u_2$ or t^e where t, u_1, u_2 and e are each normal form terms in $\mathrm{Gen}(T)$.

In the first case, then, u is a product

$$t_1 \cdot \ldots \cdot t_n$$

where each factor comes from u_1 or u_2. Since each t_i is of the form v, $inv(v)$, v^e, or $inv(v^e)$, the normal form of this term results by canceling any factors (from different u_i) that are inverses of each other. No new E-subterms are created, so no new indicator vectors are created, and our assertion follows.

The other case is when u is t^e. Note that since e is in $\mathrm{Gen}(T)$ we know that e is N-free. It suffices to show that $\mathrm{Ind}(t^e) = \mathrm{Ind}(t)$. Letting t be in normal form, t^e is

$$(t_1)^e \cdot \ldots \cdot (t_n)^e$$

Each $(t_i)^e$ is of the form

$$v^e \quad (i(v))^e \quad (v^{e'})^e \quad (inv(v^{e'}))^e$$

The first two terms are N-free. The second kind of term reduces to $v^{e * e'}$, and the indicator set for this term is precisely $\mathrm{Ind}(e)$ since e' is N-free. The last term reduces to $inv(v^{e' * e})$ and we can argue just as in the previous case.

The cases for concatenation and encryption are immediate from the induction hypothesis, since they simply propagate indicator vectors.

An Adversary Limitation. We justify now our central technique, that the adversary cannot generate messages with new indicators, using variables of sort E that are *non-originating before* node n.

Definition 6. *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:*

$$\{a \text{ of sort } E\colon a \text{ is non-originating before } n\}.$$

We assume $\mathrm{IB}_{\mathcal{B}}(n)$ is ordered in some conventional way.

Theorem 5. *Let W be an adversary web of $\mathcal{B}$, and let n be a transmission node of W, and let N be a sequence of elements drawn from $\mathrm{IB}_{\mathcal{B}}(n)$. If $v \in \mathrm{Ind}_N(\mathsf{msg}(n))$, 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 received on W, and let T_M be the set of basic values emitted by W; set $T = T_R \cup T_M$. The message $u = \mathsf{msg}(n)$ is in $\mathrm{Gen}(T)$. The set T_R is N-free, as a consequence of the fact that every message received on W must have originated, and T_M is N-free since it is a set of basic values not in N (indeed, each term in T_M has an empty indicator set). So Theorem 4 applies. Since each $t \in T_M$ has empty indicator set we conclude that every indicator in u comes from a message in T_R, as desired.

In IADH protocols, every message from regular participants has indicators in $\{\mathbf{0}\}, \{\mathbf{1}_a\}, \{\mathbf{1}_b\}, \{\mathbf{1}_x\}, \{\mathbf{1}_y\}$, etc. Since the adversary can never transmit a message with any indicators he has not received, no messages with other indicators will ever be sent or received. Messages encrypted using keys derived from Diffie-Hellman values preserve this property. Using Thm. 5 and Assumption 3, 2b:

Corollary 1. *Let $\mathcal{B}$ be a bundle for an IADH protocol using certificates $[\![\,\mathsf{cert}\ g^a \parallel P\,]\!]_{\mathsf{sk(CA)}}$ and $[\![\,\mathsf{cert}\ g^\alpha \parallel P'\,]\!]_{\mathsf{sk(CA)}}$. If $a \in \mathsf{non}_{\mathcal{B}}$ and $\mathrm{Ind}_{\langle a\rangle}(\alpha) \neq \mathbf{0}$, then $\alpha = e$ and $P = P'$.*

7 Analyzing IADH Protocols

We now embark on analyzing IADH protocols, focusing on UM, MQV, and CF. We aim to illustrate the way that our algebraic tools—normal forms and indicators—work together with the more familiar tools of symbolic protocol analysis. These are notions such as causal well-foundedness that are basic to strand spaces. We start with properties for which the indicators bear the main burden. In Section 8 we turn to implicit authentication. It requires subtler proofs, which are more sensitive to the details of the key computation.

We believe that our presentation of security goals is a contribution in itself. They appear to us to be clear distillations of the structural elements in the goals,

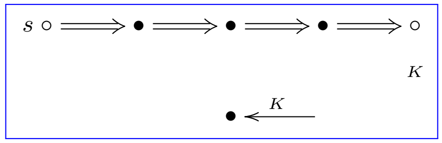


Fig. 3. Key secrecy: This diagram cannot occur

which have often appeared in more cluttered forms—particularly obscured by more operational ideas—in some of the literature.

We start though with a useful lemma about the session keys produced by regular strands, saying that they always reflect the parameters of that strand.

Lemma 2. *Let protocol Π be an* IADH *protocol, but possibly without Assumption 2, Clauses 1a and 2a.*

Suppose $\mathcal{B}$ is a Π-bundle, and s is a Π initiator or responder strand with long term secret a and ephemeral value x, succeeding with key K:

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

Π is MQV: *If $x \in \mathsf{non}_{\mathcal{B}}$, then for $K = (R_B \cdot {Y_B}^{[R_B]})^{s_A}$, we have $\mathbf{1}_x \in \mathrm{Ind}_{\langle x \rangle}(K)$. If $a \in \mathsf{non}_{\mathcal{B}}$, then $\mathbf{1}_a \in \mathrm{Ind}_{\langle a \rangle}(K)$.*

Π is CF: *If $x \in \mathsf{non}_{\mathcal{B}}$, then for $K = (R_B \cdot Y_B)^{x+a}$, we have $\mathbf{1}_x \in \mathrm{Ind}_{\langle x \rangle}(K)$. If $a \in \mathsf{non}_{\mathcal{B}}$, then $\mathbf{1}_a \in \mathrm{Ind}_{\langle a \rangle}(K)$.*

Proof. For UM, 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. 5, m_0 cannot be an adversary node either, when a or $x \in \mathsf{non}_{\mathcal{B}}$ resp.

For MQV, let $R_B = g^\eta$, where η is a possibly compound value the adversary may have engineered, and let $Y_B = g^\beta$. Now $K = g^{x\eta} \cdot g^{a\eta[g^x]} \cdot g^{x\beta[g^\eta]} \cdot g^{a\beta[g^x][g^\eta]}$. K may be a-free because $g^{a\eta[g^x]}$ and $g^{a\beta[g^x][g^\eta]}$ cancel. This occurs when $a\eta[g^x] = -a\beta[g^x][g^\eta]$, i.e. $\eta = -\beta[g^\eta]$. However, in this case x also cancels out, as $x\eta = -x\beta[g^\eta]$. So the exponent is 0 and $K = 1$, contradicting the assumption that strand s delivers a successful key.

MQV could also cancel if R_B or Y_b has indicator $\langle -1 \rangle$, but this is excluded by the same argument as with UM.

The argument for CF is the same as for MQV.

Key Secrecy and Impersonation. In Fig. 3 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.

Security Goal 6 (Key Secrecy) *Suppose that $\mathcal{B}$ is a Π-bundle with $a, b \in \mathsf{non}_{\mathcal{B}}$, and strand s is a Π 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 7. *Let protocol Π be an* IADH *protocol using any of the key computation methods in Eqns. 6, 8, 10, but possibly without Assumption 2, Clauses 1a and 2a. Then Π achieves the security goal of key secrecy.*

Proof. For sake of contradiction suppose that $\bullet \leftarrow K$ is in $\mathcal{B}$. Then K is transmitted on some node. Computing indicators relative to the basis $\langle a, b\rangle$, K has indicator $\langle 1, 1\rangle$ (by Eqns. 7–9, 11 and Lemma 2). By Thm. 5, some regular node transmits a message with indicator $\langle 1, 1\rangle$. But this is a contradiction, since regular strands transmit only values with indicators $\langle 0, 0\rangle$ and, during certification, $\langle 1, 0\rangle$ and $\langle 0, 1\rangle$.

Curiously, resistance to impersonation attacks concerns the same diagram, Fig. 3, although with different assumptions. An impersonation attack is a case in which the adversary, having compromised A's long term secret a, uses it to obtain a session key K, while causing A to have a session yielding K as session key. If A's session uses $Y_B = g^b$, where b is the uncompromised long term secret of B, then the adversary has succeeded in *impersonating* B to A.[2] The protocols MQV and CF resist impersonation attacks, but UM does not. In this result, we rely here on the freshness assumptions on regular initiator and responder strands, Assumption 2, Clauses 1a and 2a. We are in effect trading off an assumption on a long term secret for assumptions on the ephemeral values.

Security Goal 8 (Resisting Impersonation) *Suppose that $\mathcal{B}$ is a Π-bundle with $b \in \mathsf{non}_{\mathcal{B}}$, and strand s is a Π initiator or responder strand using ephemeral secret x and long term peer public value $Y = g^b$. Then $\mathcal{B}$ does not contain a listener $\bullet \leftarrow K$.*

Theorem 9. *Let protocol Π be an* IADH *protocol using either of the two key computation methods in Eqns. 8 and 10. Then Π achieves the security goal of resisting impersonation.*

Proof. For sake of contradiction suppose that $\bullet \leftarrow K$ is in $\mathcal{B}$. Then K is transmitted on some node. When we compute indicators relative to the basis $\langle b, x\rangle$, K has indicator $\langle 1, 1\rangle$ (by Eqns. 7–9, 11 and Lemma 2). By Thm. 5 we conclude that some regular node transmits a message with indicator $\langle 1, 1\rangle$. But this is a contradiction, since regular strands transmit only values with indicators $\langle 0, 0\rangle$ and, during certification, $\langle 1, 0\rangle$ and $\langle 0, 1\rangle$.

This argument does not apply to UM, because its key $K = H(g^{ab} \parallel g^{xy})$ has indicators $\{\langle 1, 0\rangle, \langle 0, 1\rangle\}$ in this basis. Thus, Theorem 5 buys us nothing. In fact, UM fails to prevent impersonation attacks.

[2] By contrast, it is hopeless—when a is compromised—to try to prevent the adversary from impersonating A to others.

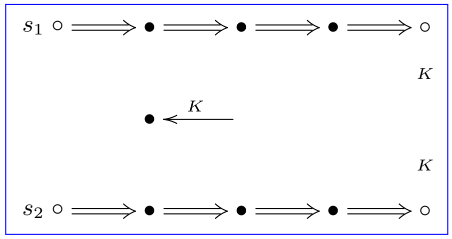


Fig. 4. Weak forward secrecy: This diagram cannot occur

Forward Secrecy. Forward secrecy is generally described as preventing disclosure of the session key of a session, if the long-term secrets of the regular participants in that session are compromised subsequently. We consider two different versions of the forward secrecy property. The first may be called *weak* forward secrecy, and all of our IADH protocols achieve it. We present weak forward secrecy in Fig. 4. Essentially, weak forward secrecy holds because the non-originating ephemeral values x, y prevent the adversary from computing the session key. Thus, Assumption 2, Clauses 1a and 2a are essential.

Security Goal 10 (Weak Forward Secrecy) *Suppose that $\mathcal{B}$ is a Π-bundle, and strands s_1, s_2 are distinct Π initiator or responder strands, issuing the same session key K. Then $\mathcal{B}$ does not contain a listener $\bullet \leftarrow K$.*

Theorem 11. *Let protocol Π be an IADH protocol using any of the key computation methods in Eqns. 6, 8, 10. Then Π achieves the weak forward secrecy security goal.*

Proof. Just as for Theorems 7 and 9: in this case compute indicators relative to the basis $\langle x, y\rangle$, and note that K has indicator $\langle 1, 1\rangle$ yet regular strands transmit only values with indicators $\langle 0, 0\rangle$ and, during certification, $\langle 1, 0\rangle$ and $\langle 0, 1\rangle$.

A stronger notion of forward secrecy stresses the word *subsequently.* A local session occurs, and the compromise of the long term keys happens after that session is finished: Can the adversary then retrieve the session key? We formalize this idea in a diagram in which the long term secrets a, b are transmitted after a session issuing in session key K completes. Moreover, we assume that the long term secrets are *uniquely originating.* This implies that they cannot have been used before the session completed, which is exactly the intended force of considering a *subsequent compromise.*

Figure 5 illustrates this situation. The slanted dotted line separates past from future, meaning that any event northwest of the dotted line occurs before any event southwest of it. This ordering relation between the end of the strand and the point of disclosure is essential to the idea. Also essential is $a, b, \mathsf{sk}(B) \in \mathsf{unique}$, where $\mathsf{sk}(B)$ is B's signing key. MQV and UM do not achieve perfect forward secrecy. CF, like the Station-to-Station protocol (Eqn 2), does, for a similar reason.

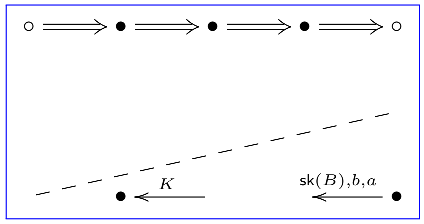


Fig. 5. Strong Forward Secrecy: This diagram cannot occur

Security Goal 12 (Forward Secrecy) *Suppose that $\mathcal{B}$ is a Π-bundle with $a, b, \mathsf{sk}(B) \in \mathsf{unique}_{\mathcal{B}}$, and strand s is a Π initiator or responder strand using long term secret a and long term peer public value $Y = g^b$. Suppose that $\bullet \rightarrow a, b$ occurs subsequent to the last reception on s. Then $\mathcal{B}$ does not contain a listener $\bullet \leftarrow K$.*

Theorem 13. *Let protocol Π be the* CF *protocol, with the ephemeral values R_A, R_B signed as $[\![\, R_A \,]\!]_{\mathsf{sk}(A)}$ and $[\![\, R_B \,]\!]_{\mathsf{sk}(B)}$. Then Π achieves the forward secrecy goal.*

Proof. Since $[\![\, R_B \,]\!]_{\mathsf{sk}(B)}$ is received on a node of s, and there is no compromise of B's signing key until it has been received, there has been a regular node transmitting $[\![\, R_B \,]\!]_{\mathsf{sk}(B)}$. This follows from the Honest Ideal Theorem [35] or the Authentication Test Principle [23].

Since a signed value $[\![\, R_B \,]\!]_{\mathsf{sk}(B)}$ is transmitted only on a regular initiator or responder strand, we know that $R_B = g^y$ for some $y \in \mathsf{non}_{\mathcal{B}}$. We may now take indicators relative to $\langle x, y \rangle$, and the rest of the proof proceeds as before.

Given the absence of signed units in MQV and UM, they have no analog to the first step of this proof.

8 The Implicit Authentication Goal

Implicit authentication has been controversial, with a distinction between "implicit key authentication" and "resisting unknown key-share attacks" [5, 25, 30].

The essential common idea is expressed in Figure 6. 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*.

Implicit key authentication and resisting unknown key share attacks differ in what non-compromise assumptions they make.

Resistance to unknown key-share attacks is the property that $A = A'$ and $B = B'$ whenever $a, b \in \mathsf{non}$. The weaker assertion, implicit key authentication, is

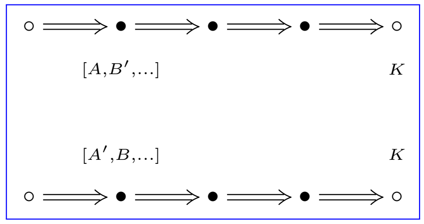


Fig. 6. Implicit authentication: In this diagram, $A = A'$ and $B = B'$

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:

> by definition the provision of implicit key authentication is considered only where B engages in the protocol with an honest entity (which E isn't). [5]

Law et al. [30] use similar language. Resisting unknown key share attacks is simpler and more robust, and we will refer to it as *implicit authentication* (without "key").

Security Goal 14 (Implicit Authentication) *Suppose that $\mathcal{B}$ is a Π-bundle with $a, b, \mathsf{sk}(B) \in \mathsf{non}_{\mathcal{B}}$, 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., where s_1, s_2 both yield session key K. Then $A = A'$ and $B = B'$.*

Weak implicit authentication states that $A = A'$, under the extra assumption that there exists an $a' \in \mathsf{non}_{\mathcal{B}}$ such that $Y_{A'} = g^{a'}$. Symmetrically, $B = B'$, under the extra assumption that there exists a $b' \in \mathsf{non}_{\mathcal{B}}$ such that $Y_{B'} = g^{b'}$.

We will prove four results. We will show that UM and CF achieve implicit authentication. Moreover, MQV achieves weak implicit authentication. Finally, (strong) implicit authentication holds for MQV, under an additional assumption.

Of these protocols, UM allows the simplest proof.

Theorem 15. 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 3, $\mathsf{sk(CA)} \in \mathsf{non}_{\mathcal{B}}$. Hence, there was a certifying strand that transmitted this certificate, and by 3, Cl. 2a, $Y_{B'} = g^{b'}$ for some b'. By symmetry, $Y_{A'} = g^{a'}$.

The key computation, with the injectiveness of $\parallel$ and H,[3] ensures $g^{a'b} = g^{ab'}$, hence $a'b = ab'$. Thus, there is some c such that $a' = ca$ and $b' = cb$. Applying Cor. 1, we conclude $B' = B$. Symmetrically, $A' = A$.

Using indicators in a richer way than previously, we obtain:

[3] In the symbolic model, hash functions are modeled as injective.

xy'	xb'	$y'a$	ab'
$\langle 0,0,1,?,?,1\rangle$	$\langle ?,?,1,0,?,0\rangle$	$\langle 1,0,0,?,?,1\rangle$	$\langle 1,?,0,0,0,0\rangle$
$\langle 0,0,?,1,1,?\rangle$	$\langle 0,1,?,0,1,?\rangle$	$\langle ?,?,0,1,0,?\rangle$	$\langle ?,1,0,0,0,0\rangle$
$x'y$	$x'b$	ya'	$a'b$

Table 4. Indicator vectors for CF authentication

Theorem 16. CF *achieves implicit authentication, when the strands* s_1, s_2 *receive* $[\![\, R_{B'} \,]\!]_{\mathsf{sk}(B')}$ *and* $[\![\, R_{A'} \,]\!]_{\mathsf{sk}(A')}$*, and* $\mathsf{sk}(A'), \mathsf{sk}(B') \in \mathsf{non}_{\mathcal{B}}$.

Proof. We start with $a, b \in \mathsf{non}_{\mathcal{B}}$, and Assumption 2 tells us $x, y \in \mathsf{non}_{\mathcal{B}}$. Using the signatures, there exist regular initiator or responder strands transmitting $[\![\, R_{B'} \,]\!]_{\mathsf{sk}(B')}$ and $[\![\, R_{A'} \,]\!]_{\mathsf{sk}(A')}$. Hence, by Assumption 2, $R_{A'} = g^{x'}$ and $R_{B'} = g^{y'}$, where $x', y' \in \mathsf{non}_{\mathcal{B}}$ and $g^{x'}, g^{y'} \in \mathsf{unique}_{\mathcal{B}}$. We may also use the certificates (as in the previous proof) to infer that $Y_{B'} = g^{b'}$ and $Y_{A'} = g^{a'}$. Also $g^{a'}, g^{b'} \in \mathsf{unique}_{\mathcal{B}}$. Since the strands compute the same session key,

$$g^{xy'} \cdot g^{xb'} \cdot g^{y'a} \cdot g^{ab'} = g^{x'y} \cdot g^{x'b} \cdot g^{ya'} \cdot g^{a'b} \tag{12}$$

We also know that none of these parameters can be replaced by a compound expression, since they are independently chosen on regular strands. Moreover, none of x, y, x', y' can equal any of a, b, a', b', as $g^x, g^y, g^{x'}, g^{y'}$ are uniquely originating, on initiator or responder strands. The exponentials of the latter all originate on certificate request strands.

Moreover, if $x = y$, then $s_1 = s_2$, so $A = A'$ and $B = B'$, and authentication is assured. So assume $x \neq y$.

We compute indicators for the four monomials on each side of Eqn. 12, as shown in Table 4. We use as basis the non-originating parameters a, b, x, y, x', y', in this order. Since we do not know whether the primed variables equal their unprimed counterparts, there are undetermined entries (?) in the indicator vectors; an integer 0 or 1 shows the definite presence or absence of a parameter.

In the table, every vertically aligned pair is compatible, i.e. we can fill in the undetermined entries so as to make the vectors agree. Moreover, if two vectors are not vertically aligned, they are incompatible. For instance, the rightmost entries have 0s for all the slots for ephemeral parameters, which put them in conflict with all of the other vectors.

Hence, $xy' = x'y, \ldots,\ ab' = a'b$. Since each of these is a parameter, and not compound, we have $x = x', y = y', a = a', b = b'$. Applying Cor. 1, $A = A'$ and $B = B'$.

Turning now to MQV:

Theorem 17. MQV *achieves weak implicit authentication.*

Proof. Let s_1, s_2 be strands in $\mathcal{B}$ as in the weak implicit authentication goal, where also $a, b, a' \in \mathsf{non}_{\mathcal{B}}$ and $Y_{A'} = g^{a'}$. Here the starting point is weaker than

$x\eta$	$a\eta[g^x]$	$x\beta[g^\eta]$	$a\beta[g^x][g^\eta]$
$\langle ?, ?, ?, 1, ?\rangle$	$\langle 1, ?, ?, ?, ?\rangle$	$\langle ?, ?, ?, 1, ?\rangle$	$\langle 1, ?, ?, ?, ?\rangle$
$\langle ?, ?, ?, ?, 1\rangle$	$\langle ?, ?, 1, ?, ?\rangle$	$\langle ?, 1, ?, 0, 1\rangle$	$\langle ?, 1, 1, 0, 0\rangle$
ψy	$\psi b[g^y]$	$a'y[g^\psi]$	$a'b[g^y][g^\psi]$

Table 5. Indicator vectors for MQV weak authentication

in CF, since we do not know that $R_{B'}, R_{A'}$ originate on regular strands; we know only that they are group elements, so of the form g^η, g^ψ, resp., for $\psi, \eta : NZE$. Likewise, $Y_{B'}$, having been certified, is some group element g^β. Since s_1, s_2 yield the same key, we have:

$$g^{x\eta} \cdot g^{a\eta[g^x]} \cdot g^{x\beta[g^\eta]} \cdot g^{a\beta[g^x][g^\eta]}$$
$$= g^{\psi y} \cdot g^{\psi b[g^y]} \cdot g^{a'y[g^\psi]} \cdot g^{a'b[g^y][g^\psi]}$$

An adversary strategy for solving this consists of an assignment of possibly compound expressions to the Greek letters ψ, η, β. The adversary wins if both sides of this equation reduce to the same normal form, but without forcing $A = A'$.

We write the indicator vectors for this in Table 5, relative to the basis $\langle a, a', b, x, y\rangle$, all in non. There are many entries ?, because we do not know whether $a = a'$, or what the adversary incorporated into the Greek letters β, η, ψ. Nevertheless, the lower right entry has 0 for the x slot, so it cannot equal the first or third entry in the first row, in which the x slot is 1. This leaves two possibilities, the second and fourth terms.

In these terms, the a slot is 1. Thus, either $a' = a$ or $b = a$. If $a' = a$, we may apply Cor 1.

So assume $a' \neq a$ and $b = a$. If we choose term 2, i.e. $a'b[g^y][g^\psi] = a\eta[g^x]$, then $\eta = a'r$, where r is the ratio of boxed terms. Turning to the term $x\eta$, we have $x\eta = xa'r$, i.e. its y and b slots are 0. Thus, it cannot equal any of the monomials on the RHS.

Choosing term 4, $a'b[g^y][g^\psi] = a\beta[g^x][g^\eta]$, then $\beta = a'r$, where r is a ratio of boxed values. But since β was certified, we can apply Cor 1 to infer that $\beta = a'$. Plugging in, we now have $a'y[g^\psi]$ with indicator $\langle 0, 1, 0, 0, 1\rangle$. Since $x\beta[g^\eta]$ has indicator $\langle 0, 1, 0, 1, 0\rangle$, there is no term in the top row that $a'y[g^\psi]$ can match.

Kaliski [25] showed implicit authentication does not hold for MQV. An adversary, observing A's ephemeral public value $R_A = g^x$, may generate a new R_E depending on R_A and $Y_A = g^a$, and then a new long-term Y_E:

$$R_E = g^x \cdot (g^a)^{[g^x]} \cdot g^{-1} \qquad Y_E = g^{[R_E]^{-1}}. \tag{13}$$

Thus, $R_E = g^{x+a[g^x]-1}$. The adversary asks CA to certify Y_E, successfully proving possession of $[R_E]^{-1}$. This is compatible with our assumptions as $\mathrm{Ind}_{\langle a\rangle}(Y_E) = \mathbf{0}$.

E's operations cancel out, so the certificate misleads B into thinking K is shared with E, when it is shared with A. A mischievous priest E can cause a criminal B to believe K shared with E, when it fact it is shared with the district attorney A. E can thus induce B to misdeliver a confession to A, leading to an unexpected plot twist in Hitchcock's movie with Montgomery Clift [24].

Definition 7. *Strands s, d with parameters $[\dots, a, x, \dots]$ and $[\dots, Y_A, R_A]$ are a* doping pair *if x appears in Y_A.*

Bundle $\mathcal{B}$ respects ephemerals *if no doping pair in $\mathcal{B}$ yields a shared key K.*

Doping, which [25] uses, is not visible to the principal executing d. We mention below a way to prevent it.

Theorem 18. *Suppose $\mathcal{B}$ is an* MQV *bundle that respects ephemerals. Then $\mathcal{B}$ satisfies (full) implicit authentication.*

Proof. Let s_1, s_2 be strands in $\mathcal{B}$ as in the implicit authentication goal, where $a, b \in \mathsf{non}_{\mathcal{B}}$ and $Y_{A'} = g^{\alpha}, Y_{B'} = g^{\beta}$, for $\alpha, \beta : NZE$. $R_{B'}, R_{A'}$ are group elements of the form g^{η}, g^{ψ}, resp., for $\psi, \eta : NZE$. Since s_1, s_2 yield the same key,

$$\begin{aligned} & g^{x\eta} \cdot g^{a\eta[g^x]} \cdot g^{x\beta[g^\eta]} \cdot g^{a\beta[g^x][g^\eta]} \\ = \; & g^{\psi y} \cdot g^{\psi b[g^y]} \cdot g^{\alpha y[g^\psi]} \cdot g^{\alpha b[g^y][g^\psi]} \end{aligned} \tag{14}$$

By Cor. 1, either $\alpha = a$ or $\alpha = b$ or $\mathrm{Ind}_{\langle a,b\rangle}(\alpha) = \mathbf{0}$. Likewise, either $\beta = a$ or $\beta = b$ or $\mathrm{Ind}_{\langle a,b\rangle}(\beta) = \mathbf{0}$.

If both $\alpha, \beta \in \{a, b\}$, then we have a case of weak authentication from both sides, so Thm. 17 gives the desired result. Assume then that at least one, e.g. α, has $\mathrm{Ind}_{\langle a,b\rangle}(\alpha) = \mathbf{0}$. Since $\mathcal{B}$ respects ephemerals, (s_1, s_2) is not a doping pair, and (s_2, s_2) is not a doping pair, so x, y are syntactically absent from α. So in fact $\mathrm{Ind}_{\langle a,b,x,y\rangle}(\alpha) = \mathbf{0}$. By Equation 14

$$\begin{aligned} & x\eta + a\eta[g^x] + x\beta[g^\eta] + a\beta[g^x][g^\eta] \\ = \; & \psi y + \psi b[g^y] + \alpha y[g^\psi] + \alpha b[g^y][g^\psi] \end{aligned}$$

The Greek letters may be compound expressions. Thus, $\alpha y[g^\psi]$ and $\alpha b[g^y][g^\psi]$ may each yield a number of monomials when reduced to normal forms. However, because α has indicator $\mathbf{0}$, and the boxed terms have indicator $\mathbf{0}$, monomials resulting from $\alpha y[g^\psi]$ all have indicator $\langle 0, 0, 0, 1\rangle$. Monomials resulting from $\alpha b[g^y][g^\psi]$ all have indicator $\langle 0, 1, 0, 0\rangle$.

When the LHS normalizes, no monomial on the LHS can have indicator $\langle 0, 1, 0, 0\rangle$ or $\langle 0, 0, 0, 1\rangle$ because each one has a factor of x or a. So, the last two summands on the RHS cannot contribute any monomials to the normal form.

By Lemma 2, the LHS has non-zero contributions of a and x. Hence, ψ must have non-zero contributions of them.

We write ψ as the sum $\psi = \psi_{nz} + \psi_0$, where ψ_{nz} collects all the monomials in ψ with non-zero indicators, and ψ_0 collects all those with indicator $\mathbf{0}$.

In particular, $\psi_0 y$ must cancel $\alpha y[g^\psi]$, and $\psi_0 b[g^y]$ must cancel $\alpha b[g^y][g^\psi]$. Each of these leads to the conclusion:

$$-(\psi_0/[g^\psi]) = \alpha. \tag{15}$$

Hence the normal form of ψ_0 must be some $\varphi[g^\psi]$, so that $[g^\psi]$, which has occurrences of x, can syntactically cancel. Hence, the normal form of ψ is

$$\psi_{nz} + \varphi[g^\psi],$$

contradicting the well-foundedness of the syntactic terms.

The preceding analysis sheds some light on Kaliski's attack (13) on MQV. There, equation 15 holds with $\psi_0 = -1$ and $\alpha = [g^\psi]^{-1}$. However, we here have the additional assumption above that $\mathcal{B}$ respects ephemerals: since s_1, s_2 is not a doping pair, α can have no occurrence of x, but as we have observed, ψ must.

The interesting approaches to preventing the Kaliski attack—that is, to ensure that executions respect ephemerals—involve time and causality. Suppose that the CA always takes at least a minimum time t_C between receiving a certification request and issuing the certificate. Moreover, the initiator always times out and discards a session if it does not complete within a period t_I, where $t_I < t_C$. For instance, if T_C is an hour and t_I is a half hour, this approach would be practically workable. No synchronization between different principals is required for this, since each participant makes purely local decisions about timing. Non-malicious sessions would be entirely unaffected. Then, in any completed session, no certified value can involve an ephemeral in that session, since it cannot yet have been generated at the time the value was certified.

9 Uniform Equality and the Completeness of $\mathsf{AG}\hat{}$

In this section we justify the use of $\mathsf{AG}\hat{}$, specifically the use of $\mathsf{AG}\hat{}$-normal forms to model messages. Any theorem of $\mathsf{AG}\hat{}$ surely holds in all DH-structures. Theorem 19 gives us the converse, namely that every equation that holds in all DH-structures is a theorem of $\mathsf{AG}\hat{}$. Indeed, given a non-principal ultraflter D over the set of primes, there is a single structure $\mathcal{M}_D$ that is "generic" for all of the DH-structures: An equation $s = t$ is valid in $\mathcal{M}_D$ if and only if it is valid in infinitely many DH-structures.

We work first with models of the language of $\mathsf{AG}\hat{}$ but with the $[\cdot]$ removed from the signature. They have all the structure required to analyze UM and CF. We then extend our results to DH-structures equipped with a $[\cdot]$ function.

Algebraically isomorphic structures can have very different *computational* properties. Indeed, the prime field $\mathbb{F}_q$ presented as the group of integers mod q induces 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. Although the algebra is blind to the computational distinctions, we focus here on the algebraic equations between terms in DH-structures.

First, we show that the field of scalars, i.e. the exponents, carries all the algebraic information in a model of $\mathsf{AG}\hat{}$.

Definition 8. *Let F be a field. We construct a $[\cdot]$-free model $\mathcal{M}_F$ of theory $\mathsf{AG}\hat{}$ 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 operation $\cdot$ is taken to be $+$ from E, thus 1 and inv are taken to be 0 and $-$. Exponentiation is multiplication: a^e is interpreted as $a * e$.*

For each field F, any $\mathcal{M}_F$ satisfies all of the equations in $\mathsf{AG}\hat{}$. When F is the prime field of order q then $\mathcal{M}_F = \mathcal{M}_{\mathbb{F}_q}$ is, up to isomorphism, precisely the standard DH algebra of order q. When F is the additive group of rational numbers then $\mathcal{M}_F = \mathcal{M}_{\mathbb{Q}}$ will be of interest to us below.

The key device for reasoning about uniform equality across DH-structures is the notion of *ultraproduct*, cf. e.g. [9]. We let the variable D range over non-principal ultrafilters over the set of prime numbers.

Definition 9. *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 8. For simplicity we write $\mathcal{M}_D$ for $\mathcal{M}_{\mathbb{F}_D}$.*

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) when D is non-principal, every cofinite set is in D. We show below that the set of *equations* valid in $\mathcal{M}_D$ does not depend on which non-principal D we use.

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

Lemma 3. *The structure $\mathcal{M}_{\mathbb{Q}}$ can be embedded as a submodel in any $\mathcal{M}_D$.*

Proof. 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 $\mathbb{Q}$ 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 $\mathcal{M}_D$.

Lemma 4. *Let $t : G$ be in normal form, in the $[\cdot]$-free sublanguage of $\mathsf{AG}\hat{}$. There exists an environment $\eta : Vars \to \mathbb{Q}$ such that if u and u' are distinct subterms of t, $\eta(u) \neq \eta(u')$ in $\mathcal{M}_{\mathbb{Q}}[D]$.*

Proof. In the structure $\mathcal{M}_{\mathbb{Q}}$, exponentiation is interpreted as multiplication, so it suffices to consider the expression obtained by replacing $\cdot$ and *inv* by $+$ and $-$, and the exponentiation operator by $*$, and viewing t as an ordinary rational expression in several variables $x_1, \ldots, x_k$ (the variables occurring in t). We may view t as determining a real function $f_t : \mathbb{R}^k \to \mathbb{R}$. In fact each subterm t' of t similarly determines a function from $\mathbb{R}^k$ to $\mathbb{R}$ (not all variables of t will occur in all subterms, but we may still treat each as inducing a k-ary function). So the family of subterms of t determines a (finite) set of rational functions, and we can find a rational point $\boldsymbol{r} = (r_1, ..., r_k)$ such that no two of these functions agree on $\boldsymbol{r}$. We define η to map each x_i to r_i.

Corollary 2. *If s and t are distinct normal forms then it is not the case that $\mathcal{M}_{\mathbb{Q}} \models s = t$.*

Proof. Form the term $u \equiv s \cdot inv(t)$. Since s and t are distinct normal forms this term is in normal form. By Lemma 4 there is an environment η with $\eta(s) \neq \eta(t)$, and the result follows.

$\mathsf{AG}\hat{}$ is complete for uniform equality in the absence of the $[\cdot]$-function:

Theorem 19. *For each pair of G-terms s and t in the $[\cdot]$-free fragment of $\mathsf{AG}\hat{}$, 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 some non-principal D, $\mathcal{M}_D \models s = t$*
5. $\mathcal{M}_{\mathbb{Q}} \models s = t$
6. *if s reduces to s' with s' irreducible, and t reduces to t' with 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 6 implies 1. The first three of these steps are immediate, as is the fact that 6 implies 1. The fact that 4 implies 5 follows from Lemma 3. To conclude 6 from 5, use Corollary 2.

As a corollary of Theorem 19, these equivalences hold for E-term equations as well. 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.

Corollary 3. *If $\mathcal{M}_{\mathbb{F}_q} \models s = t$ holds for infinitely many q, then for all q, $\mathcal{M}_{\mathbb{F}_q} \models s = t$.*

Proof. Suppose that $\{q \colon \mathcal{M}_{\mathbb{F}_q} \models s = t\}$ is infinite. Then there is a non-principal ultrafilter D containing this set. So (4) in Thm. 19 holds, and we apply (4)$\Rightarrow$(2).

The equivalence of $\mathsf{AG}\hat{}$-provability with equality in the models is the technical core of our claim that $\mathsf{AG}\hat{}$ captures "uniform equality."

The model $\mathcal{M}_{\mathbb{Q}}$ is convenient: this single model, based on a familiar structure, serves to witness uniform equality simplifies analyses. Our first analysis of MQV used this.

The model $\mathcal{M}_D$ satisfies an even more striking property. It follows from results of Ax [3] that the first-order theory of $\mathcal{M}_D$ is decidable. So the structure $\mathcal{M}_D$ is an attractive one for closer study of the "uniform" properties of DH-structures.

Incorporating $[\cdot]$. An analogue of Theorem 19 holds for the full language of $\mathsf{AG}\hat{}$, the language appropriate for reasoning about MQV. The starting point is like Lemma 4.

Lemma 5. *Let $t : G$ be in normal form, in the language of* $\mathsf{AG}\hat{}$. *There exists an interpretation of the* $[\cdot]$ *function and an environment* η *such such that if* u *and* u' *are distinct subterms of* t, $\eta(u) \neq \eta(u')$ *in* $\mathcal{M}_{\mathbb{Q}}$.

Proof. The proof is by induction on the number of $[\cdot]$-subterms of t. If this number is 0 then we may apply Lemma 4 and simply use the following simple $[\cdot]$ function: $[a] = a$ if $a \neq 0$ and $[0] = 1$.

Otherwise let $[s]$ be a subterm of t such that s is $[\cdot]$-free. Let t' be the term obtained from t by replacing each occurrence of $[s]$ by a variable v occurring nowhere in t. Then t' is in normal form, so by induction there is a function $[\cdot]_0$ and an environment η that acts as an injection over the subterms of t'. We may assume that η is defined on all the variables of t (even though some may not occur in t'). We claim that we can define $[\cdot]$ so that the resulting function, taken with the same environment η satisfies the Lemma. We define $[\cdot]$ to agree with $[\cdot]_0$ on all values except $\eta(s)$, where we put $[\eta(s)] = \eta(v)$. Since η is guaranteed to yield different values on distinct subterms of t', the use of $[\cdot]_0$ will yield the same values as the use of $[\cdot]$ on subterms of t other than $[s]$.

By an argument similar to that establishing Corollary 2 we obtain

Corollary 4. $\mathsf{AG}\hat{} \vdash s = t$ *iff for all* $[\cdot]$ *functions* $\mathcal{M}_{\mathbb{Q}} \models s = t$.

From this follows, finally:

Theorem 20. *For each pair of* G*-terms* s *and* t *in the full language of* $\mathsf{AG}\hat{}$ *the following are equivalent*

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

Proof. As for Theorem 19 we can establish a cycle of entailments. The non-trivial changes to the arguments presented for Theorem 19 are

- to conclude 5 from 4 now, we observe that given a $[\cdot]$-function on $\mathcal{M}_{\mathbb{Q}}$ that entails $\mathcal{M}_{\mathbb{Q}} \models s \neq t$ we can, via the embedding of $\mathcal{M}_{\mathbb{Q}}$ into $\mathcal{M}_D$, construct a $[\cdot]$-function on $\mathcal{M}_D$ such that $\mathcal{M}_D \models s \neq t$, and
- to conclude 6 from 5 now, we use Corollary 4.

10 Conclusion and Related Work

Related Work. Within the symbolic model, there has been substantial work on some aspects of DH, starting with Boreale and Buscemi [7], which provides a symbolic semantics [1, 18, 32] 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 [20] use a method based on Horn clauses and resolution modulo AC, providing automated proofs of passive security. Maude-NPA [16, 17] is also usable to analyze many protocols involving DH, again depending heavily on unification.

All of these approaches appear to face a fundamental problem with a theory like the $\mathsf{AG}\hat{}$ theory of Section 9, in which it would be unwise to rely on the decidability of the unifiability problem. Unifiability is undecidable in the theory of rings, essentially 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 [4].

Küsters and Truderung [29] 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 [6]. Their approach covers a surprising range of protocols, although, like [10], not IADH protocols such as MQV or CF.

Another contrast between this paper and previous work is the uniform treatment of numerous security goals. Our methods are applicable to confidentiality, authentication, and further properties such as forward secrecy.

Our adversary model is active. For passive attacks, there has been some work on computational soundness for Diffie-Hellman, with Bresson et al. [8] giving an excellent treatment.

Conclusion and Future Work. In this paper, we have applied the strand space framework to IADH protocols, such as UM, CF, and MQV, establishing about a dozen security properties of them. While all of them have been previously claimed, few have been proved in as informative a way as we do here. Moreover, our proofs rely on a few fundamental principles that can be easily applied. They combine rewriting techniques and the *indicator* idea.

We also provided a deeper model-theoretic treatment that justifies our rewriting theory with respect to an adversary model. Our adversary can use any algebraic facts that are true in all but finitely many DH-structures. Since other cryptographic primitives such as bilinear pairings are built by enriching DH-structures, it is highly desirable to have proof techniques that work in this rich algebraic framework.

Connecting this with the standard computational model remains for future work. In our model the adversary must choose its whole strategy before seeing the concrete messages for a particular run, or even knowing the prime q. This raises the question of the *computational soundness* of our approach, a focus of future research: Does the Decisional Diffie-Hellman assumption ensure that the adversary gets no asymptotic advantage from knowing q and the concrete messages?

Our proofs here are handcrafted. However, we are currently pursuing an approach using model-finding in *geometric logic*, a generalization of Horn logic, which offers great promise for mechanizing many of these conclusions.

Acknowledgments. We gratefully acknowledge support by the National Science Foundation under grant CNS-0952287. We are grateful to Shriram Krishnamurthi, Moses Liskov, Cathy Meadows, John Ramsdell, Paul Rowe, Paul Timmel, and Ed Zieglar for extremely helpful and often vigorous discussions.

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. James Ax. The elementary theory of finite fields. *The Annals of Mathematics*, 88(2):pp. 239–271, 1968.
4. Jan A. Bergstra and J. V. Tucker. The rational numbers as an abstract data type. *Journal of The ACM*, 54, 2007.
5. Simon Blake-Wilson and Alfred Menezes. Authenticated Diffe-Hellman key agreement protocols. In *Selected Areas in Cryptography*, pages 630–630. Springer, 1999.
6. Bruno Blanchet. An efficient protocol verifier based on Prolog rules. In *14th Computer Security Foundations Workshop*, pages 82–96. IEEE CS Press, June 2001.
7. 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.
8. 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.
9. C.C. Chang and H.J. Keisler. Model Theory, volume 73 of Studies in Logic and the Foundations of Mathematics, 1990.
10. 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.
11. 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.
12. C.J.F. Cremers. *Scyther - Semantics and Verification of Security Protocols*. Ph.D. dissertation, Eindhoven University of Technology, 2006.
13. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.
14. Whitfield Diffie, Paul C. van Oorschot, and Michael J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2(2):107–125, 1992.

15. 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.
16. Santiago Escobar, Catherine Meadows, and José Meseguer. State space reduction in the Maude-NRL protocol analyzer. *Computer Security-ESORICS 2008*, pages 548–562, 2008.
17. 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.
18. Marcelo Fiore and Martín Abadi. Computing symbolic models for verifying cryptographic protocols. In *Computer Security Foundations Workshop*, June 2001.
19. 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.
20. 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.
21. 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.
22. Joshua D. Guttman. State and progress in strand spaces: Proving fair exchange. *Journal of Automated Reasoning*, 2011. Accepted, March 2010. DOI: 10.1007/s10817-010-9202-1.
23. Joshua D. Guttman and F. Javier Thayer. Authentication tests and the structure of bundles. *Theoretical Computer Science*, 283(2):333–380, June 2002. Conference version appeared in *IEEE Symposium on Security and Privacy*, May 2000.
24. Alfred Hitchcock. I confess. Warner Brothers, March 1953. http://www.imdb.com/title/tt0045897/.
25. 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.
26. 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.
27. H. Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol. In *Advances in Cryptology–CRYPTO 2005*, pages 546–566. Springer, 2005.
28. Sebastian Kunz-Jacques and David Pointcheval. About the Security of MTI/C0 and MQV. *Security and Cryptography for Networks*, pages 156–172, 2006.
29. 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.
30. 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.
31. Alfred Menezes. Another look at HMQV. *Journal of Mathematical Cryptology*, 1:47–64, 2007.
32. 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.

33. 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.
34. 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.
35. 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.