Decidable Analysis of Cryptographic Protocols with Products and Modular Exponentiation

Vitaly Shmatikov *

SRI International
shmat@csl.sri.com

Abstract. We demonstrate that the symbolic trace reachability problem for cryptographic protocols is decidable in the presence of an Abelian group operator and modular exponentiation from arbitrary bases. We represent the problem as a sequence of symbolic inference constraints and reduce it to a system of linear Diophantine equations. For a finite number of protocol sessions, this result enables fully automated, sound and complete analysis of protocols that employ primitives such as Diffie-Hellman exponentiation and modular multiplication without imposing any bounds on the size of terms created by the attacker, but taking into account the relevant algebraic properties.

1 Introduction

Symbolic constraint solving for cryptographic protocols is a subject of very active research in the protocol analysis community [1, 9, 17, 3, 13, 6, 14, 5]. While the analysis problem is undecidable in its most general form [8], it has been proved NP-complete [17] for a finite number of protocol sessions even without *a priori* bounds on the size of terms that may be created by the attacker. Therefore, symbolic constraint solving provides a *fully automated* technique for discovering attacks on trace-based security properties such as secrecy and authentication.

Attacker's capabilities are represented by a set of inference rules modeling how the attacker can learn new terms from the terms he already knows. Since protocol messages may include variables (representing data fields whose values are not known to the honest recipient in advance), an attack is modeled as a symbolic protocol trace or skeleton (*e.g.*, an interleaving of several protocol sessions, at the end of which the attacker learns the secret). The goal of symbolic constraint solving is to determine whether there exists a consistent instantiation of all variables such that every message sent by the attacker is derivable, using the chosen inference rules, from the set of terms available to him.

Initial research on symbolic protocol analysis [1, 17, 3, 13] followed the so-called Dolev-Yao model in assuming that the attacker does not have access to the algebraic properties of the underlying cryptographic primitives. This assumption fails for primitives such as `xor` (exclusive or) and modular exponentiation, which are widely used in protocol constructions. The attacker may exploit associativity,

* Partially supported by ONR Grants N00014-01-1-0837 and N00014-03-1-0961.

commutativity, and cancellation of inverses. Bull's recursive authentication protocol [15, 18] and group Diffie-Hellman protocol [19, 16] had been proved correct in the free algebra model, but then attacks were discovered once algebraic properties of, respectively, `xor` and modular exponentiation were taken into account. In this paper, we demonstrate that the symbolic analysis problem is decidable even if the attacker term algebra is extended with an Abelian group operator and modular exponentiation from an arbitrary base. In particular, this result enables fully automated analysis of Diffie-Hellman-based protocols.

Overview. Section 2 introduces the term algebra that we use to model the attacker's capabilities, and the equational theory modeling algebraic properties of the relevant cryptographic primitives. In Section 3, we briefly describe how the protocol analysis problem is reduced to a sequence of symbolic inference constraints. In Section 4, we extend previous results on ground derivability [6] to modular exponentiation, and, following [14], demonstrate the existence of *conservative solutions*. Section 5 contains the main technical result: symbolic constraint solving problem in the presence of an Abelian group operator and modular exponentiation is reduced to the solvability in integers of a special *decidable* system of quadratic equations. Conclusions follow in Section 6.

Related work. The techniques of this paper follow closely those of [6, 14]. In [6], the problem was only considered in the ground (as opposed to symbolic) case, and the term algebra did not include exponential terms. The reduction to a quadratic Diophantine system was first developed in [14], but only exponentiation from a constant base was supported and, most importantly, decidability remained an open question. Proving decidability is the main contribution of this paper.

Partial results for protocol analysis in the presence of Diffie-Hellman exponentiation were recently obtained by Boreale and Buscemi [4], and Chevalier *et al.* [5]. Neither addresses decidability for an Abelian group operator outside exponents. The decision procedure of [4] requires an *a priori* upper bound on the number of factors in each product. In general, computing upper bounds on the size of variable instantiations needed for a feasible attack is the most challenging task in establishing decidability. Therefore, [4] does not fully solve the problem.

Chevalier *et al.* [5] prove that the protocol analysis problem is NP-complete in the presence of Diffie-Hellman exponentiation, but only for a restricted class of protocols. No more than one new variable may be introduced in each protocol message, and the attacker is not permitted to instantiate variables with products. These restrictions rule out not only non-deterministic protocols, but also some well-defined, deterministic protocols.

Narendran *et al.* investigated decidability of unification modulo the equational theory of multiplication and exponentiation [11, 10]. While equational unification is an important subproblem in symbolic protocol analysis, unification alone is insufficient to decide whether a particular symbolic term is derivable given a set of attacker's inference rules.

Pereira and Quisquater [16] analyzed the group Diffie-Hellman protocol [19] taking into account algebraic properties of Diffie-Hellman exponents. They did not attempt to address the general problem of deciding whether a particular symbolic attack trace has a feasible instantiation.

2 Model

A protocol specification is a set of roles. Each role is a sequence of sent and received messages. Messages received by the role may contain variables, representing data fields whose value is unknown to the recipient (*e.g.*, the counterparty's nonce). Since the source of the received messages cannot be established on an insecure network, we assume that the attacker receives all messages sent by the honest roles, and sends all messages received by the honest roles.

An attack on any trace-based property of cryptographic protocols can be represented as a *symbolic attack trace* (see [13, 3, 9] for details). A symbolic trace is a particular interleaving of a finite number of protocol roles. For example, an attack on secrecy is modeled by an interleaving at the end of which the attacker outputs the value that was supposed to remain secret. An attack on authentication is modeled by an interleaving at the end of which the attacker has successfully authenticated himself to an honest party.

A trace is *feasible* if every message received by the honest roles can be derived by the attacker from his initial knowledge and intercepted messages. Therefore, for each message sent by the attacker, a symbolic inference problem must be decided: is there an instantiation of variables such that the sent term is derivable in the attacker's term algebra? To stage the attack, the attacker may need to send several messages in a particular order. Deciding whether the attack is feasible thus requires solving several symbolic inference problems simultaneously.

Term algebra. The attacker's capabilities are modeled by a term algebra with pairing, symmetric encryption, multiplication, and exponentiation. The notation is shown in fig. 1. For multiplication, there is a unit **1** and a multiplicative inverse. Like [5], do not allow products in the base of exponentials, nor permit exponential terms to be multiplied with other terms. This restriction is necessary, because introducing distributive laws for exponentials results in an undecidable equational unification problem [10]. In contrast to [5], we impose no restrictions on multiplication of terms other than exponentials, permit variables to be instantiated to products, and allow more than one new variable per message.

Our algebra is untyped, *e.g.*, we do not distinguish between keys and other messages. This enables us to discover a wider class of attacks than strongly typed techniques. Extensions of the algebra with primitives for public-key encryption, digital signatures, and one-way functions do not present any conceptual problems as far as decidability is concerned (*e.g.*, see [17, 14]) and are left out for brevity.

Since our primary motivation is analysis of protocols based on Diffie-Hellman, we use the relations of fig. 2 to model the corresponding algebraic structure (see, *e.g.*, [12]). In Diffie-Hellman, exponentiation is $\mod$ prime p, and the base α is chosen so as to generate a cyclic subgroup $\alpha, \alpha^2, \ldots, \alpha^q \equiv 1 \mod p$ of some prime order q that divides $p-1$. We implicitly assume that exponential terms

$\langle t_1, t_2 \rangle$	Pairing of terms t_1 and t_2
$\{t_1\}_{t_2}$	Term t_1 encrypted with term t_2 using a symmetric algorithm
$t_1 \cdot \ldots \cdot t_n$	Product of terms where $\forall i\ t_i \neq \mathsf{exp}(u, v)$
t^{-1}	Multiplicative inverse of term t where $t \neq \mathsf{exp}(u, v)$
$\mathsf{exp}(t_1, t_2)$	$t_1^{t_2}$ where t_1 is not headed with $\cdot$, $t_2 \neq \mathsf{exp}(u, v)$

Fig. 1. Message term constructors

Rules for products:	Associative, commutative, and
	$t \cdot \mathbf{1} \to t$
	$t \cdot t^{-1} \to \mathbf{1}$
Rules for inverses:	$(t^{-1})^{-1} \to t$
	$(t_1 \cdot t_2)^{-1} \to t_2{}^{-1} \cdot t_1{}^{-1}$
Rules for exponentials:	$\mathsf{exp}(t, \mathbf{1}) \to t$
	$\mathsf{exp}(\mathsf{exp}(t_1, t_2), t_3) \to \mathsf{exp}(t_1, t_2 \cdot t_3)$

Fig. 2. Normalization rules for products, inverses, and exponentials

are $\mod p$, and that multiplication is $\mod q$ (recall that exponential terms may not be multiplied in our term algebra). Because $\cdot$ forms a cyclic Abelian group, every member has a multiplicative inverse.

The rules of fig. 2 are convergent modulo associativity and commutativity of $\cdot$, thus every term t has a unique normal form $t\downarrow$ up to associativity and commutativity. We assume that terms are kept in normal form.

Attacker model. The attacker's ability to derive terms is characterized as a term set closure under the inference rules of fig. 3. These rules reflect common cryptographic assumptions about the difficulty of some number-theoretic problems. For example, the attacker cannot compute v when given $\mathsf{exp}(u, v)$ (the discrete logarithm problem). Given $\mathsf{exp}(u, v)$ and $\mathsf{exp}(u, v')$, there is no rule that enables the attacker to compute $\mathsf{exp}(u, v \cdot v')$ (the computational Diffie-Hellman problem).

3 Symbolic Inference Constraints

Any symbolic trace can be converted into a ordered sequence of symbolic inference constraints. Suppose u_i is the message received by some honest role, and let T_i be the set of all messages sent by the honest roles (and thus learned by the attacker) prior to sending u_i. The constraint sequence is simply $\mathbf{C} = \{u_i : T_i\}$.

Each constraint $u_i : T_i$ can be interpreted as "at step i, the attacker knows messages in T_i and must generate message u_i." We will refer to u_i as the *target term* of the constraint. Both u_i and messages in T_i may contain variables. We assume that T_1 contains terms that are initially known to the attacker, such as $\mathbf{1}$ and constants specific to the protocol. Observe that constraint sequences are

$$
\begin{array}{cccc}
\text{Unpairing (UL, UR)} & \text{Decryption (D)} & \text{Pairing (P)} & \text{Encryption (E)} \\
\dfrac{T \vdash \langle u,v\rangle}{T \vdash u} \quad \dfrac{T \vdash \langle u,v\rangle}{T \vdash v} & \dfrac{T \vdash \{u\}_v \;\; T \vdash v}{T \vdash u} & \dfrac{T \vdash u \;\; T \vdash v}{T \vdash \langle u,v\rangle} & \dfrac{T \vdash u \;\; T \vdash v}{T \vdash \{u\}_v}
\end{array}
$$

$$
\begin{array}{ccc}
\text{Multiplication (M)} & \text{Inversion (I)} & \text{Exponentiation (X)} \\
\dfrac{T \vdash u_1 \;\; \ldots \;\; T \vdash u_n}{T \vdash u_1 \cdot \ldots \cdot u_n} & \dfrac{T \vdash u}{T \vdash u^{-1}} & \dfrac{T \vdash u \;\; T \vdash v}{T \vdash \mathsf{exp}(u,v)} \\
\forall i \; u_i \neq \mathsf{exp}(u',v') & u \neq \mathsf{exp}(u',v') & u \text{ is not headed with } \cdot,\; v \neq \mathsf{exp}(u',v')
\end{array}
$$

Fig. 3. Attacker's capabilities

monotonic: if $j < i$, then $T_j \subseteq T_i$. Also, since variables represent terms unknown to the recipient, every variable must occur for the first time in some target term u_i (this property is sometimes referred to as *origination*).

A ground substitution σ is a *solution* of $u : T$ (written $\sigma \Vdash u : T$) if $T\sigma \vdash u\sigma$ is derivable using the inference rules of fig. 3. Given a constraint sequence $\mathbf{C} = \{u_1 : T_1, \ldots, u_n : T_n\}$, σ is a solution of the constraint sequence ($\sigma \Vdash \mathbf{C}$) if $\forall i \; T_i\sigma \vdash u_i\sigma$ is derivable using the rules of fig. 3.

If T is a finite set of terms, let $\mathrm{St}(T)$ be the set of subterms defined in the standard way. Let $\mathrm{St}(\mathbf{C}) = \bigcup_{u_i:T_i \in \mathbf{C}} \mathrm{St}(T_i \cup u_i)$, and define $\mathcal{S}(\mathbf{C}) = \mathrm{St}(\mathbf{C}) \setminus \mathrm{Var}(\mathbf{C})$ to be the set of all non-variable subterms of $\mathbf{C}$. Let $\mathcal{S}^p(\mathbf{C})$ be the closure of this set under $\cdot$, inverse, and exponentiation, defined inductively: (i) if $t \in \mathcal{S}(\mathbf{C})$, then $t \in \mathcal{S}^p(\mathbf{C})$, (ii) if $t_{1,2} \in \mathcal{S}^p(\mathbf{C})$ and $t_{1,2} \neq \mathsf{exp}(u,v)$, then $t_1 \cdot t_2, {t_1}^{-1}, {t_2}^{-1} \in \mathcal{S}^p(\mathbf{C})$, (iii) if $t_{1,2} \in \mathcal{S}^p(\mathbf{C})$ and t_1 is not headed with $\cdot$ and $t_2 \neq \mathsf{exp}(u,v)$, then $\mathsf{exp}(t_1,t_2) \in \mathcal{S}^p(\mathbf{C})$.

Running example. We will use the following symbolic trace as an (artificial) running example to illustrate our constraint solving procedure. An event $A \longrightarrow t$ models honest role A sending message t, $B \longleftarrow t'$ models B receiving t', *etc.*

$$
\begin{array}{lll}
1.\; A \longrightarrow a \cdot b & 3.\; A \longrightarrow \{a\}_b & 5.\; B \longrightarrow \langle b \cdot X, \mathsf{exp}(c,a)\rangle \\
2.\; B \longleftarrow a \cdot X \cdot Y & 4.\; B \longleftarrow \{Y\}_b & 6.\; A \longleftarrow \mathsf{exp}(c,a^7)
\end{array}
$$

The goal of symbolic protocol analysis is to determine whether there exists an instantiation of variables X and Y such that every term sent by the attacker and received by an honest participant (*i.e.*, every term t appearing as $P \longleftarrow t$) is derivable using the rules of fig. 3. This is equivalent to deciding whether the following constraint sequence has a solution:

$$
a \cdot X \cdot Y : a \cdot b \;; \quad \{Y\}_b : a \cdot b, \{a\}_b \;; \quad \mathsf{exp}(c,a^7) : a \cdot b, \{a\}_b, \langle b \cdot X, \mathsf{exp}(c,a)\rangle
$$

4 Normal Proofs and Conservative Solutions

We extend the results of [6, 14] to the term algebra with exponentiation.

Definition 1 (Ground proof). *A* proof *of $T \vdash u$ is a tree labeled with sequents $T \vdash v$ and such that (a) every leaf is labeled with $T \vdash v$ such that $v \in T$; (b) every node has n parents $s_1, \ldots, s_n$ such that $\dfrac{s_1 \ \cdots \ s_n}{T \vdash v}$ is an instance of one of the inference rules of fig. 3; (c) the root is labeled with $T \vdash u$.*

Definition 2 (Normal ground proof). *A proof $\mathcal{P}$ of $T \vdash u$ is* normal *if either $u \in \mathrm{St}(T)$ and every node is labeled $T \vdash v$ with $v \in \mathrm{St}(T)$, or $\mathcal{P} = C[\mathcal{P}_1, \ldots, \mathcal{P}_n]$ where every proof $\mathcal{P}_i$ is a normal proof of some $T \vdash v_i$ with $v_i \in \mathrm{St}(T)$ and context C is built using the inference rules (P),(E),(X),(M),(I) only.*

Lemma 1 (Existence of normal ground proof). *If there is a ground proof of $T \vdash u$, then there is a normal ground proof of $T \vdash u$.*

Proposition 1. *If there is a ground proof of $T \vdash u$ that uses only rules (M), (I), and (X), then there exists a proof of $T \vdash u$ of the form $\dfrac{T \vdash u_1 \ T \vdash u_2}{T \vdash u}$ where $u_1 \in T$, and either $u_2 \in T$, or the proof of $T \vdash u_2$ uses rules (M) and (I) only.*

If the constraint sequence $\mathbf{C}$ is solvable, then it has a *conservative* solution [14], in which every variable is instantiated to a product of subterms (and their inverses) that are already present in the original sequence, or to an exponential with a subterm as the base and a product of subterms as the exponent.

Definition 3 (Conservative substitution). *Substitution σ is* conservative *if $\forall x \in \mathrm{Var}(\mathbf{C}) \quad \mathrm{St}(x\sigma) \subseteq \mathcal{S}^p(\mathbf{C})\sigma$.*

Theorem 1 (Existence of conservative solution). *If there exists a solution $\sigma \Vdash \mathbf{C}$, then there exists a conservative solution $\sigma^* \Vdash \mathbf{C}$.*

Lemma 2 (Existence of conservative proof). *If $\sigma \Vdash \mathbf{C}$ is conservative, then $\forall u : T \in \mathbf{C}$ there exists a proof of $T\sigma \vdash u\sigma$ such that for every node labeled $T\sigma \vdash v$, either $v \in \mathrm{St}(\mathbf{C})\sigma$, or node $T\sigma \vdash v$ is obtained by an (M), (I), or (X) inference rule* and *is only used as a premise of an (M), (I), or (X) rule.*

5 Decision Procedure for Symbolic Inference Constraints

For any constraint sequence $\mathbf{C}$, we define a nondeterministic finite reduction $\leadsto$. For each step $\leadsto_i$, we show that there are finitely many $\mathbf{C}_i$ such that $\mathbf{C}_{i-1} \leadsto_i \mathbf{C}_i$, and that $\mathbf{C}_{i-1}$ has a solution if and only if some $\mathbf{C}_i$ has a solution. The final sequence has a solution if and only if a special system of quadratic Diophantine equations has a solution. Quadratic Diophantine equations are undecidable in general, but the system obtained in our case is solvable if and only if a particular linear subsystem is solvable. Since linear Diophantine equations are decidable, this establishes decidability of the symbolic protocol analysis problem.

Following Theorem 1, we will be interested only in conservative solutions. The reduction proceeds as follows (steps 1-3 are essentially the same as in [14]):

1. Guess subterm equalities.
2. For each constraint, guess all derivable subterms and add them to the set of terms available to the attacker.
3. Remove all constraints in which the derivation involves inference rules other than (M), (I), and (X).
4. Guess and instantiate bases of exponential terms.
5. Replace every constraint in which the derivation involves (X) with an equivalent constraint in which the derivation involves only (M) and (I).
6. Substitute all target terms that introduce new variables.
7. Solve a linear Diophantine system to determine whether the final sequence has a solution.

5.1 Determine subterm equalities

Suppose $\mathbf{C}$ has some solution σ. In the first reduction step $\leadsto_1$, we guess the equivalence relation on $\mathrm{St}(\mathbf{C})$ induced by σ, *i.e.*, $\forall s_i, s_j \in \mathrm{St}(\mathbf{C})$, we guess whether $s_i\sigma = s_j\sigma$ or not. Since $\mathrm{St}(\mathbf{C})$ is finite, there are finitely many equivalence relations to consider. Each relation represents a set of unification problems in an Abelian group, which are decidable [2]. There are finitely many most general unifiers consistent with any given equivalence relation. We nondeterministically guess the right unifier θ (in practice, θ would have to be found by by exhaustive enumeration), and let $\mathbf{C}_1 = \mathbf{C}\theta$.

Lemma 3. *$\exists\sigma \Vdash \mathbf{C}$ if and only if $\exists\sigma \Vdash \mathbf{C}_1$ for some $\mathbf{C}_1$ such that $\mathbf{C} \leadsto \mathbf{C}_1$.*

Proposition 2. *$\forall s, s' \in \mathrm{St}(\mathbf{C}_1)$ if $s \neq s'$, then $s\sigma \neq s'\sigma$.*

Running example. In our running example (Section 3), we guess that the only subterm equality is $\{Y\}_b = \{a\}_b$, giving us the unifier $[Y \to a]$ and this $\mathbf{C}_1$:

$$a^2 \cdot X : a \cdot b \ ; \quad \{a\}_b : a \cdot b, \{a\}_b \ ; \quad \mathsf{exp}(c, a^7) : a \cdot b, \{a\}_b, \langle b \cdot X, \mathsf{exp}(c, a)\rangle$$

5.2 Determine order of subterm derivation

Following [14], the second reduction step $\leadsto_2$ guesses which subterms of $\mathbf{C}_1\sigma$ are derivable by the attacker using inference rules of fig. 3, and adds each derivable subterm s to every constraint $u_i : T_i$ such that s is derivable from $T_i\sigma$.

1. Guess $S_\vdash = \{s \in \mathrm{St}(\mathbf{C}_1) \mid \exists u_i : T_i \in \mathbf{C}_1$ s.t. there exists a proof of $T_i\sigma \vdash s\sigma\}$.
2. $\forall s \in S_\vdash$ guess j_s s.t. there exists a proof of $T_{j_s}\sigma \vdash s\sigma$, but not of $T_{j_s-1}\sigma \vdash s\sigma$.
3. Guess linear ordering $\prec$ on $S_\vdash$ such that
   - If $s \prec s'$, then the normal proof of $T\sigma \vdash s\sigma$ does not contain any node labeled with $T\sigma \vdash s'\sigma$.
   - If $j_s < j_{s'}$, then $s \prec s'$.

   Such an ordering always exists [14] and represents the order in which subterms of $\mathbf{C}_1$ are derived.
4. Arrange $s_1, \ldots, s_k \in S_\vdash$ according to the ordering $\prec$, and insert each s in the constraint sequence immediately before the $u_{j_s} : T_{j_s}$ constraint. Let $\mathbf{C}_2$ be the resulting constraint sequence.

Lemma 4. *$\exists\sigma \Vdash \mathbf{C}$ if and only if $\exists\sigma \Vdash \mathbf{C}_2$ for some $\mathbf{C}_2$ such that $\mathbf{C} \leadsto \mathbf{C}_2$.*

Running example. In our running example, we guess that subterms $b \cdot X$ and $\mathsf{exp}(c, a)$ are derivable from T_3. Therefore, we obtain the following $\mathbf{C}_2$:

$a^2 \cdot X : a \cdot b; \quad \{a\}_b : a \cdot b, \{a\}_b;$
$b \cdot X : a \cdot b, \{a\}_b, \langle b \cdot X, \mathsf{exp}(c, a)\rangle; \quad \mathsf{exp}(c, a) : a \cdot b, \{a\}_b, \langle b \cdot X, \mathsf{exp}(c, a)\rangle, b \cdot X;$
$\mathsf{exp}(c, a^7) : a \cdot b, \{a\}_b, \langle b \cdot X, \mathsf{exp}(c, a)\rangle, b \cdot X, \mathsf{exp}(c, a)$

5.3 Eliminate all inferences other than (M), (I), and (X)

Lemma 5. *Consider any $u : T \in \mathbf{C}_2$ and the last inference of the proof of $T\sigma \vdash u\sigma$.*

- *If $u\sigma \in T\sigma$, then $u \in T$.*
- *If $u\sigma$ is obtained by (UL), then $\langle u, t'\rangle \in T$ for some term t'.*
- *If $u\sigma$ is obtained by (UR), then $\langle t', u\rangle \in T$ for some term t'.*
- *If $u\sigma$ is obtained by (D), then $\{u\}_{t'} \in T$ for some term t'.*
- *If $u\sigma$ is obtained by (P), then $u = \langle u_1, u_2\rangle$ and $u_{1,2} \in T$ for some terms $u_{1,2}$.*
- *If $u\sigma$ is obtained by (E), then $u = \{u_1\}_{u_2}$ and $u_{1,2} \in T$ for some terms $u_{1,2}$.*

Lemma 5 implies that all constraints where derivation involves at least one instance of any rule other than (M), (I), or (X) can be discovered by syntactic inspection [14]. Let $\leadsto_3$ consist in eliminating all such constraints, and let $\mathbf{C}_3$ be the resulting constraint sequence.

Proposition 3. *$\forall u : T \in \mathbf{C}_3$, proof of $T\sigma \vdash u\sigma$ uses only inference rules (M), (I), and (X).*

Lemma 6. *$\exists\sigma \Vdash \mathbf{C}$ if and only if $\exists\sigma \Vdash \mathbf{C}_3$ for some $\mathbf{C}_3$ such that $\mathbf{C} \leadsto \mathbf{C}_3$.*

Running example. In our example, we guess the first and fifth constraints were obtained by application of rules (M), (I) and (X) only. We eliminate the middle three constraints, obtaining the following $\mathbf{C}_3$:

$$a^2 \cdot X : a \cdot b \; ; \quad \mathsf{exp}(c, a^7) : a \cdot b, \{a\}_b, \langle b \cdot X, \mathsf{exp}(c, a)\rangle, b \cdot X, \mathsf{exp}(c, a)$$

5.4 Instantiate bases of exponential terms

For each subterm of $\mathbf{C}_3$, $\leadsto_4$ guesses whether the solution σ instantiates it to an exponential, and, if so, nondeterministically chooses the base of exponentiation. Let $\hat{S} = \{s \in \mathrm{St}(\mathbf{C}_3) \mid s\sigma = \mathsf{exp}(b, e)\}$. There are only finitely many subsets of $\mathrm{St}(\mathbf{C}_3)$, thus $\hat{S} = \{s_1, \ldots, s_r\}$ can be computed by exhaustive enumeration of all possibilities. By Definition 3 and because products may not appear in the base, $\forall s_i \in \hat{S}\ b_i \in \mathcal{S}(\mathbf{C}_3)\sigma$. Since $\mathcal{S}(\mathbf{C}_3)$ is finite, there are finitely many possible values for $t_{b_i} \in \mathcal{S}(\mathbf{C}_3)$ such that $t_{b_i}\sigma = b_i$. Let x_i be a fresh variable, and θ_i the (unique) most general unifier of s_i and $\mathsf{exp}(t_{b_i}, x_i)$. Define $\mathbf{C}_4 = \mathbf{C}_3\theta_1 \ldots \theta_r$.

Proposition 4 (Explicit exponentiation bases). *If $s\sigma = \mathsf{exp}(b, e)$ for some $s \in \mathrm{St}(\mathbf{C}_4)$, then $s = \mathsf{exp}(t_b, t_e)$ s.t. $t_b\sigma = b, t_e\sigma = e$.*

Lemma 7. *$\exists\sigma \Vdash \mathbf{C}$ if and only if $\exists\sigma \Vdash \mathbf{C}_4$ for some $\mathbf{C}_4$ such that $\mathbf{C} \leadsto \mathbf{C}_4$.*

Running example. Since exponentiation bases are already explicit, $\mathbf{C}_4 = \mathbf{C}_3$.

5.5 Replace inferences using (X) with those using (M) and (I)

This step is based on the following intuition. Suppose $u = \mathsf{exp}(b, v_1 \cdot \ldots \cdot v_p)$. Under the Diffie-Hellman assumption, the attacker cannot construct $\mathsf{exp}(b, x \cdot y)$ from $\mathsf{exp}(b, x)$ and $\mathsf{exp}(b, y)$, thus $u : T$ has a solution if and only if T contains some term $t = \mathsf{exp}(b, v'_1 \cdot \ldots \cdot v'_q)$ (if $q = 0$, then $t = b$), and $v_1 \cdot \ldots \cdot v_p = v'_1 \cdot \ldots \cdot v'_q \cdot x$ where x is derivable from T. Informally, x can be thought of as the "additional" exponent to which t must be raised in order to obtain u.

Consider all $u_i : T_i \in \mathbf{C}_4$ in order, and define $\rightsquigarrow_5$ as $\overset{(1)}{\rightsquigarrow}_5 \ldots \overset{(N)}{\rightsquigarrow}_5$ where N is the number of constraints in $\mathbf{C}_4$. By Propositions 1 and 3, the proof of $T_i\sigma \vdash u_i\sigma$ is of the form $\dfrac{T_i\sigma \vdash u_{i1} \; T_i\sigma \vdash u_{i2}}{T_i\sigma \vdash u_i\sigma}$ where $u_{i1} \in T_i\sigma$, and either $u_{i2} \in T_i\sigma$, or the proof of $T_i\sigma \vdash u_{i2}\sigma$ is built up using rules (M) and (I) only.

If the last inference rule in the above proof is not (X), then the entire proof is built using inference rules (M) and (I) only. We set $\overset{(i)}{\rightsquigarrow}_5$ to be the identity.

Now suppose the last rule in the proof of $T_i\sigma \vdash u_i\sigma$ is (X). Then $u_i\sigma = \mathsf{exp}(u_{i1}, u_{i2}) \downarrow = \mathsf{exp}(b, e)$ for some ground b, e. Since u_{i1} must have the same base as $u_i\sigma$, $u_{i1} = \mathsf{exp}(b, e_1)$ for some ground e_1. By Proposition 1, $u_{i1} \in T_i\sigma$. Since $\mathrm{St}(\mathbf{C}_4) \subseteq \mathrm{St}(\mathbf{C}_1)$, by Proposition 2 $\exists t_1 \in T_i$ such that $t_1\sigma = u_{i1}$. There are finitely many candidates for t_1, and $\overset{(i)}{\rightsquigarrow}_5$ chooses one nondeterministically.

By Proposition 4, $u_i = \mathsf{exp}(t_b, t_e)$, $t_1 = \mathsf{exp}(t'_b, t_{e_1})$ where $t_b\sigma = t'_b\sigma = b$. Therefore, $t_b = t'_b$. We conclude that (i) $u_i = \mathsf{exp}(t_b, t_e)$; (ii) $t_1 = \mathsf{exp}(t_b, t_{e_1}) \in T_i$; (iii) $t_e\sigma = t_{e_1}\sigma \cdot u_{i2}$ (or, equivalently, $t_{e_1}\sigma^{-1} \cdot t_e\sigma = u_{i2}$); and (iv) proof of $T_i\sigma \vdash u_{i2}$ uses inference rules (M) and (I) only.

By Propositions 1 and 3, $u_i : T_i$ has a solution only if $x_{i2} : T_i$ has a solution where x_{i2} is a fresh variable such that $x_{i2}\sigma = u_{i2} = t_{e_1}\sigma^{-1} \cdot t_e\sigma$. Define $\overset{(i)}{\rightsquigarrow}_5$ to replace $u_i : T_i$ with ${t_{e_1}}^{-1} \cdot t_e : T_i$.

Proposition 5. *$\forall u : T \in \mathbf{C}_5$, proof of $T\sigma \vdash u\sigma$ uses only rules (M) and (I).*

Proposition 6. *$\forall x \in \mathrm{Var}(\mathbf{C}_5)$ let $u_{k_x} : T_{k_x} \in \mathbf{C}_5$ be the constraint in which x occurs for the first time. Then $u_{k_x} = x^{q_x} \cdot \prod_{j \geq 0} u_{k_x j}$ where q_x is an integer, $u_{k_x j}$ are not headed with $\cdot$, and $x \notin \mathrm{St}(T_{k_x} \cup \{u_{k_x j} \mid j \geq 0\})$.*

Lemma 8. *$\exists\sigma \Vdash \mathbf{C}$ if and only if $\exists\sigma \Vdash \mathbf{C}_5$ for some $\mathbf{C}_5$ such that $\mathbf{C} \rightsquigarrow \mathbf{C}_5$.*

Definition 4. *Define $\mathcal{Q}_{max} = \prod_{x \in \mathrm{Var}(\mathbf{C}_5)} q_x$ where q_x is the power of x in the constraint in which it occurs for the first time.*

Running example. Replace $\mathsf{exp}(c, a^7)$ with $a^{-1} \cdot a^7 = a^6$, obtaining this $\mathbf{C}_5$:

$$a^2 \cdot X : a \cdot b \; ; \quad a^6 : a \cdot b, \{a\}_b, \langle b \cdot X, \mathsf{exp}(c, a)\rangle, b \cdot X, \mathsf{exp}(c, a)$$

5.6 Substitute target terms that introduce new variables

The $\rightsquigarrow_6$ step takes each target term in which some variable occurs for the first time, and replaces the entire term with a new variable. In the resulting sequence, every variable appears for the first time as the target term of some constraint. For example, if x occurs for the first time in $a \cdot x^2 : T_i$, let $\theta_i = [x \to \hat{x}^{\frac{1}{2}} \cdot a^{-\frac{1}{2}}]$ where $\hat{x}$ is a new variable, and apply θ_i to the entire constraint sequence.

Let k_x be the index of $u_i : T_i$ in which variable x first occurs. By Proposition 6, $u_i = x^{q_x} \cdot \prod_{j \geq 0} u_{ij}$ for some integer q_x.

Definition 5. *$\forall u_i : T_i \in \mathbf{C}_5$, define*

$$\theta_i = \begin{cases} [x \to \hat{x}^{\frac{1}{q_x}} \cdot \prod u_{ij}^{-\frac{1}{q_x}}] & \textit{if } i = k_x \textit{ for some } x;\ \hat{x} \textit{ is a fresh variable} \\ \emptyset & \textit{otherwise} \end{cases}$$

If more than one variable appears for the first time in u_i, any one of them may be chosen.

Let $\mathbf{C}_6 = \mathbf{C}_5\theta_1 \ldots \theta_{N_5}$ where N_5 is the number of constraints in $\mathbf{C}_5$. Although only integer powers appear in $\mathbf{C}_5$, $\mathbf{C}_6$ may contain rational powers.

Proposition 7. *$\forall \hat{x} \in \mathrm{Var}(\mathbf{C}_6)$ $\hat{x}$ first occurs in $\hat{x} : T \in \mathbf{C}_6$ where $\hat{x} \notin \mathrm{St}(T)$.*

Informally, term sets T_i are well-ordered if terms appearing in multiple sets always appear in the same position. Due to monotonicity (Section 3), if $i < i'$, then $T_i \subseteq T_{i'}$. Without loss of generality, we can assume that $\mathbf{C}_6$ is well-ordered.

Definition 6. *$\mathbf{C}_6$ is* well-ordered *if, $\forall t_{ij} \in T_i$ $\forall t_{i'j} \in T_{i'}$ $t_{ij} = t_{i'j}$.*

Proposition 8. *For any rational r appearing as a power of some term in $\mathbf{C}_6$, $r \cdot \mathcal{Q}_{max}$ is an integer.*

Lemma 9. *$\exists \sigma \Vdash \mathbf{C}$ if and only if $\exists \sigma \Vdash \mathbf{C}_6$ for some $\mathbf{C}_6$ such that $\mathbf{C} \rightsquigarrow \mathbf{C}_6$.*

Running example. In our example, $\theta_1 = [X \to \hat{X} \cdot a^{-2}]$, $\theta_2 = \emptyset$. Therefore, $\mathbf{C}_6 = \mathbf{C}_5\theta_1\theta_2$ is:

$$\hat{X} : a \cdot b\ ; \quad a^6 : a \cdot b, \{a\}_b, \langle b \cdot \hat{X} \cdot a^{-2}, \mathsf{exp}(c,a)\rangle, b \cdot \hat{X} \cdot a^{-2}, \mathsf{exp}(c,a)$$

5.7 Solve system of linear Diophantine equations

We now convert the constraint sequence $\mathbf{C}_6$ into a system of quadratic Diophantine equations which is solvable if and only if $\exists\ \sigma \Vdash \mathbf{C}_6$. We then demonstrate that the quadratic part *always* has a solution as long as a particular *linear* subsystem is solvable. Since linear Diophantine equations are decidable [7], this establishes that the symbolic protocol analysis problem is also decidable.

The key to this result is Lemma 10. Intuitively, we prove that, for every constraint $u : T \in \mathbf{C}_6$, the target term $u\sigma$ must be equal to some product of integer powers of *non-variable* terms appearing in set T. We then represent each power as an integer variable, and convert the derivation problem for each constraint into a system of linear Diophantine equations.

Define $\Phi(t)$ to be the set of all top-level factors of t. If $t = t_1^{r_1} \cdot \ldots \cdot t_n^{r_n}$ where none of t_i are headed with $\cdot$, then $\Phi(t) = \{t_1^{r_1}, \ldots, t_n^{r_n}\}$. For example, $\Phi(a^{-2} \cdot b^{\frac{3}{2}}) = \{a^{-2}, b^{\frac{3}{2}}\}$. Define $\Psi(t) = \{t_i^{r_i} \in \Phi(t) \mid t_i \neq \hat{x} \in \mathrm{Var}(\mathbf{C}_6)\}$ to be the set of all non-variable factors of t. Let $\psi(t) = \prod_{f \in \Psi(t)} f$, *i.e.*, $\psi(t)$ is t with all factors of the form $\hat{x}^r$ removed. For example, $\psi(a \cdot (\{\hat{x}\}_k)^3 \cdot \hat{x}^{\frac{2}{5}}) = a \cdot (\{\hat{x}\}_k)^3$.

Lemma 10. $\forall \hat{x} \in \mathrm{Var}(\mathbf{C}_6)$*, let* k_x *be the index of the constraint in which variable* x *occurs for the first time. Then* $\forall \sigma \Vdash \mathbf{C}_6, \forall u_i : T_i \in \mathbf{C}_6$

$$u_i\sigma = \prod_{t_{ij} \in T_i} (\psi(t_{ij})\sigma)^{\hat{z}[i,j]} \tag{5.1}$$

such that

$$\hat{z}[i,j] = z[i,j] + \sum_{j'>j} (\sum_{\hat{x}^r \in \Phi(t_{ij'})} (\hat{z}[k_x, j] \cdot r \cdot z[i,j'])) \tag{5.2}$$

for some integers $\hat{z}[i,j], z[i,j]$*, where* $1 \leq i \leq |\mathbf{C}_6|$*,* $\forall i \ 1 \leq j, j' \leq |T_i|$*, and* $\forall j > |T_{k_x}| \ \hat{z}[k_x, j] = 0$*.*

Proof. By induction over the length of the constraint sequence. For the induction basis, consider $u_1 : T_1 \in \mathbf{C}_6$. By Proposition 5, the proof of $T_1\sigma \vdash u_1\sigma$ contains only rules (M) and (I). Therefore, $u_1\sigma = \prod_{t_{1j} \in T_1} (t_{1j}\sigma)^{z[1,j]}$ for some integers $z[1,j]$, where $1 \leq j \leq |T_1|$. By Proposition 7, no variables occur in T_1. Therefore, $\forall j \ t_{1j} = \psi(t_{1j})$ and $\forall \hat{x} \in \mathrm{Var}(\mathbf{C}_6), j', r \ \ \hat{x}^r \notin \Phi(t_{1j'})$. Then $\forall j \ \hat{z}[1,j] = z[1,j]$, and we obtain $u_1\sigma = \prod_{t_{1j} \in T_1} (\psi(t_{1j})\sigma)^{\hat{z}[1,j]}$.

Now suppose the lemma is true for all constraints up to and including $u_{i-1} : T_{i-1}$, $i \geq 2$. Applying Proposition 5 to $u_i : T_i$, we obtain that

$$u_i\sigma = \prod_{t_{ij'} \in T_i} (t_{ij'}\sigma)^{z[i,j']} \tag{5.3}$$

Consider any $t_{ij'}$ from the above product. By definition of $\psi(t_{ij'})$, $t_{ij'} = \psi(t_{ij'}) \cdot \hat{x}_1^{r_1} \cdot \ldots \cdot \hat{x}_m^{r_m}$ for some variables $\hat{x}_1, \ldots, \hat{x}_m$ and rational constants $r_1, \ldots, r_m$ where $m \geq 0$. Consider any variable $\hat{x} \in \{\hat{x}_1, \ldots, \hat{x}_m\}$, and let k_x be the index of the first constraint in which x occurs. By Proposition 7, the fact that $\hat{x}$ occurs in T_i implies that $u_i : T_i$ cannot be the first constraint in which $\hat{x}$ occurs. There must be a preceding constraint of the form $\hat{x} : T_{k_x} \in \mathbf{C}_6$ and $k_x < i$. By the induction hypothesis, $\hat{x}\sigma = \prod_{t_{k_x j} \in T_{k_x}} (\psi(t_{k_x j})\sigma)^{\hat{z}[k_x,j]}$. By monotonicity, $T_{k_x} \subseteq T_i$. By Definition 6, $\forall j \leq |T_{k_x}| \ t_{k_x j} = t_{ij}$. Moreover, since $\hat{x}$ occurs in $t_{ij'}$, $|T_{k_x}| < j'$ by Proposition 7. Set $\hat{z}[k_x, j] = 0 \ \forall j > |T_{k_x}|$, and replace each $t_{k_x j}$ with the corresponding t_{ij}, obtaining $\hat{x}\sigma = \prod_{j<j'} (\psi(t_{ij})\sigma)^{\hat{z}[k_x,j]}$.

Substituting values for $\hat{x}_1\sigma, \ldots, \hat{x}_m\sigma$ into equation 5.3, we obtain $u_i\sigma = \prod_{t_{ij'} \in T_i} (\ \psi(t_{ij'})\sigma \cdot \prod_{\hat{x}^r \in \Phi(t_{ij'})} (\prod_{j<j'} (\psi(t_{ij})\sigma)^{\hat{z}[k_x,j] \cdot r}) \ \ )^{z[i,j']}$.

Distributing the exponent $z[i,j']$, obtain that $u_i\sigma$ is equal to

$$\prod_{t_{ij}\in T_i} (\psi(t_{ij})\sigma)^{z[i,j]} \cdot \prod_{t_{ij'}\in T_i} (\prod_{j<j'} ((\psi(t_{ij})\sigma)^{\sum_{\hat{x}^r\in\Phi(t_{ij'})}(\hat{z}[k_x,j]\cdot r\cdot z[i,j'])}))$$

Observing that $\prod_{t_{ij'}\in T_i}(\prod_{j<j'} \dots t_{ij}\dots) = \prod_{t_{ij}\in T_i}(\prod_{j'>j} \dots t_{ij}\dots)$, we conclude that $u_i\sigma = \prod_{t_{ij}\in T_i}(\psi(t_{ij})\sigma)^{z[i,j]+\sum_{j'>j}(\sum_{\hat{x}^r\in\Phi(t_{ij'})}(\hat{z}[k_x,j]\cdot r\cdot z[i,j']))}$, completing the induction.

We now convert each constraint into an equivalent system of linear Diophantine equations. If this system is unsolvable, the constraint cannot be satisfied. If, on the other hand, there exist some values of $\hat{z}[i,j]$ that solve the linear system, we will prove that quadratic equations 5.2 are guaranteed to have a solution.

Consider any $u_i : T_i \in \mathbf{C}_6$. By Lemma 10, $u_i\sigma = \prod_{t_{ij}\in T_i}(\psi(t_{ij})\sigma)^{\hat{z}[i,j]}$. By definition, $\psi(t_{ij})$ does not contain any variables as top-level factors. It is possible that $\hat{x}_k^{p_k} \in \Phi(u_i)$ for some variable $\hat{x}_k$ and rational p_k. Applying Proposition 7 and Lemma 10, we obtain that $\forall\hat{x}_k \in \mathrm{Var}(\mathbf{C}_6)\ \hat{x}_k\sigma = \prod_{t_{k_xj}\in T_{k_x}}(\psi(t_{k_xj})\sigma)^{\hat{z}[k_x,j]}$. Therefore, equation 5.1 can be rewritten as

$$\prod_{\hat{x}_k^{p_k}\in\Phi(u_i)}(\prod_{t_{k_xj}\in T_{k_x}}(\psi(t_{k_xj})\sigma)^{\hat{z}[k,j]})^{p_k} \cdot \prod_{u_{il}\in\Psi(u_i)} u_{il}\sigma = \prod_{t_{ij}\in T_i}(\psi(t_{ij})\sigma)^{\hat{z}[i,j]} \tag{5.4}$$

For any variable $\hat{x}_k$ occurring in u_i, it must be that $k_x \leq i$ since k_x is the index of the first constraint in which $\hat{x}_k$ occurs. According to Definition 6, $\forall\hat{x}_k, t_{k_xj} \in T_{k_x}\ t_{k_xj} = t_{ij}$. Dividing the right-hand side of equation 5.4 by $\prod_{\hat{x}_k^{p_k}\in\Phi(u_i)}(\prod_{t_{k_xj}\in T_{k_x}}(\psi(t_{k_xj})\sigma)^{\hat{z}[k_x,j]})^{p_k}$, we obtain

$$\prod_{u_{il}\in\Psi(u_i)} u_{il}\sigma = \prod_{t_{ij}\in T_i} (\psi(t_{ij})\sigma)^{y[i,j]} \tag{5.5}$$

where

$$y[i,j] = \hat{z}[i,j] - \sum_{\hat{x}_k^{p_k}\in\Phi(u_i)} p_k \cdot \hat{z}[k_x,j] \tag{5.6}$$

Recall that $\hat{z}[k_x,j] = 0$ if $j > |T_{k_x}|$.

Let $\mathbf{F}(\mathbf{C}_6) = \bigcup_{u_i:T_i\in\mathbf{C}_6}(\Psi(u_i) \cup \{\Psi(t_{ij}) \mid t_{ij} \in T_i\})$ be the set of all factors appearing in equations 5.5. Since $\mathbf{F}(\mathbf{C}_6) \subseteq \mathrm{St}(\mathbf{C}_6) \subseteq \mathrm{St}(\mathbf{C}_1)$, by Proposition 2 $\forall t, t' \in \mathbf{F}(\mathbf{C}_6)$ if $t \neq t'$, then $t\sigma \neq t'\sigma$. Therefore, $\forall\mathbf{t} \in \mathbf{F}(\mathbf{C}_6)\ \forall u_i : T_i \in \mathbf{C}_6$, the following system of linear equations must hold:

$$\underbrace{q}_{\substack{\text{if } \mathbf{t}^q \in \Psi(u_i),\\ 0 \text{ otherwise}}} = \sum_{t_{ij}\in T_i} \underbrace{q_j \cdot y[i,j]}_{\substack{\text{if } \mathbf{t}^{q_j} \in \Psi(t_{ij}),\\ 0 \text{ otherwise}}} \tag{5.7}$$

where $y[i,j]$ are integer variables (i ranges over the length of the constraint sequence, and, for each i, j ranges from 1 to $|T_i|$), and $q, q_1, \dots, q_{|T_i|}$ are rational constants. Multiplying equation 5.7 by the lowest common multiplier of the denominators of $q, q_1, \dots, q_{|T_i|}$, we obtain a linear system over $y[i,j]$.

Lemma 11. $\mathbf{C}$ *has a solution if and only if the system of equations 5.7 has a solution in integers for some* $\mathbf{C}_6$ *such that* $\mathbf{C} \rightsquigarrow \mathbf{C}_6$.

Proof. It follows immediately from the reduction in this section that if system 5.7 does not have a solution in integers, then $\mathbf{C}_6$ does not have a solution, either. It is necessary to show that if system 5.7 has a solution in integers, then system 5.6 and, especially, the quadratic system 5.2 also have a solution.

Let $\{y[i,j]\}$ be any solution of system 5.7. First, $\forall \hat{x} \in \mathbf{C}_6\ \forall j$ set $y[k_x, j] = 0$. Since $\forall \hat{x} \in \mathbf{C}_6\ \Psi(u_{k_x}) = \Psi(\hat{x}) = \emptyset$, equation 5.7 degenerates into $0 = \sum_{t_{k_x j} \in T_{k_x}} q_j \cdot y[k_x, j]$ and is still satisfied. By Proposition 7, $u_{k_x} = \hat{x}_k$. Therefore, $\sum_{\hat{x}_k^{p_k} \in \Phi(u_{k_x})} p_k \cdot \hat{z}[k_x, j] = \hat{z}[k_x, j]$, and $y[k_x, j] = \hat{z}[k_x, j] - \hat{z}[k_x, j] = 0$. System 5.6 is thus satisfied by $y[k_x, j] = 0$ as well.

Now, $\forall \hat{x} \in \mathbf{C}_6\ \forall j$ set $\hat{z}[k_x, j] = \mathcal{Q}_{max}$. Recall from Proposition 8 that $\mathcal{Q}_{max}$ is an integer such that $r \cdot \mathcal{Q}_{max}$ is an integer for any rational power r appearing in $\mathbf{C}_6$. We need to show that systems 5.6 and 5.2 are solvable in integers.

First, consider system 5.6. If $i = k_x$ for some x, it becomes $0 = \mathcal{Q}_{max} - \mathcal{Q}_{max}$. If $\forall x\ i \neq k_x$, it becomes $y[i,j] = \hat{z}[i,j] - \sum_{\hat{x}_k^{p_k} \in \Phi(u_i)} p_k \cdot \mathcal{Q}_{max}$, and is solved by setting $\hat{z}[i,j] = y[i,j] + \sum_{\hat{x}_k^{p_k} \in \Phi(u_i)} p_k \cdot \mathcal{Q}_{max}$ since $p_k \cdot \mathcal{Q}_{max}$ is an integer.

It remains to show that the quadratic system 5.2 has a solution in integers. Pick any $u_i : T_i \in \mathbf{C}_6$ and fix it. Proof is by induction over j from $|T_i|$ to 1. For the base case, consider $j = |T_i|$. Because there are no $j' > j$, set $z[i,j] = \hat{z}[i,j]$.

Now suppose the proposition is true for $z[i, j+1], \ldots, z[i, |T_i|]$. To complete the proof, it is sufficient to show that there exists an integer value for $z[i,j]$ that satisfies equation 5.2. Observe that $z[i,j']$ is an integer $\forall j' > j$ (by the induction hypothesis), and $\hat{z}[k_x, j] \cdot r = \mathcal{Q}_{max} \cdot r$ is an integer $\forall \hat{x}$ such that $\hat{x}^r \in \Phi(t_{ij'})$ (by Proposition 8). Therefore, $z[i,j] = \hat{z}[i,j] - \sum_{j'>j}(\sum_{\hat{x}^r \in \Phi(t_{ij'})}(\hat{z}[k_x, j] \cdot r \cdot z[i,j']))$ is an integer solution for equation 5.2. This completes the induction.

Running example. In our running example, we are solving the following $\mathbf{C}_6$:

$$\hat{X} : a \cdot b\ ; \quad a^6 : a \cdot b, \{a\}_b, \langle b \cdot \hat{X} \cdot a^{-2}, \mathsf{exp}(c,a)\rangle, b \cdot \hat{X} \cdot a^{-2}, \mathsf{exp}(c,a)$$

$\mathbf{C}_6$ has a solution *iff* the following system 5.5 is solvable in integers:

$$\begin{aligned} 1 &= (a \cdot b)^{y[1,1]} \\ a^6 &= (a \cdot b)^{y[2,1]} \cdot (\{a\}_b)^{y[2,2]} \cdot (\langle b \cdot \hat{X} \cdot a^{-2}, \mathsf{exp}(c,a)\rangle)^{y[2,3]} \cdot \\ &\quad\ (b \cdot a^{-2})^{y[2,4]} (\mathsf{exp}(c,a))^{y[2,5]} \end{aligned}$$

Since $\Psi(u_1) = \emptyset$, $\prod_{u_{1l} \in \Psi(u_1)} u_{1l}\sigma = 1$, and $\psi(b \cdot \hat{X} \cdot a^{-2}) = b \cdot a^{-2}$.

We set $y[1,1] = 0$ because $k_x = 1$, and convert the second equation into an equivalent linear Diophantine system 5.7, treating all non-atomic terms such as $\langle \ldots, \ldots \rangle$ and $\mathsf{exp}(c,a)$ as constants:

$$\begin{aligned} 6 &= y[2,1] - 2 \cdot y[2,4] & \qquad 0 &= y[2,2] \\ 0 &= y[2,1] + y[2,4] & 0 &= y[2,3] \\ & & 0 &= y[2,5] \end{aligned}$$

The solution of this system is $y[2,1]=2, y[2,4]=-2$. Therefore, the constraint sequence has a solution, and the corresponding symbolic trace is feasible. In this example, $\mathcal{Q}_{max}=1$, therefore, $\hat{z}[1,1]=1$, and $\hat{X}=(a\cdot b)^{\hat{z}[1,1]}=a\cdot b$. Reconstructing the values of original variables, we obtain $X=\hat{X}\cdot a^{-2}=a^{-1}\cdot b$.

Theorem 2 (Soundness and completeness). *Symbolic constraint sequence* $\mathbf{C}$ *has a solution if and only if the system of linear equations 5.7 has a solution in integers for some* $\mathbf{C}_6$ *such that* $\mathbf{C} \rightsquigarrow \mathbf{C}_6$.

6 Conclusions

We have presented a decision procedure for symbolic analysis of cryptographic protocols employing Abelian group operators and modular exponentiation from arbitrary bases, assuming the number of protocol sessions is bounded. Decidability is proved by reducing the symbolic constraint satisfiability problem to the solvability of a particular system of linear Diophantine equations.

This result enables fully automated analysis of a wide class of cryptographic protocols, such as those based on group Diffie-Hellman, that cannot be analyzed in the standard Dolev-Yao model. The next step is development of practical protocol analysis techniques. Instead of nondeterministically guessing subterm equalities and the order of subterm derivation, the analysis tool would search for a solution by inductively analyzing the structure of target terms, similar to the techniques of [13]. We expect that this approach will result in better average-case complexity than the generic decision procedure presented here.

Acknowledgments. Thanks to the anonymous reviewers for insightful comments, and to Jon Millen whose collaboration was invaluable in developing the symbolic protocol analysis framework on which the results presented in this paper are based.

References

1. R. Amadio and D. Lugiez. On the reachability problem in cryptographic protocols. In *Proc. 11th International Conference on Concurrency Theory (CONCUR '00)*, volume 1877 of *LNCS*, pages 380–394, 2000.
2. F. Baader and W. Snyder. Unification theory. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, chapter 8, pages 445–532. Elsevier Science, 2001.
3. M. Boreale. Symbolic trace analysis of cryptographic protocols. In *Proc. 28th International Colloquium on Automata, Languages and Programming (ICALP '01)*, volume 2076 of *LNCS*, pages 667–681, 2001.
4. M. Boreale and M. Buscemi. On the symbolic analysis of low-level cryptographic primitives: modular exponentiation and the Diffie-Hellman protocol. In *Proc. Workshop on the Foundations of Computer Security (FCS)*, 2003.
5. Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the security of protocols with Diffie-Hellman exponentiation and products in exponents. Technical Report IFI-Report 0305, CAU Kiel, 2003.

6. H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proc. 18th Annual IEEE Symposium on Logic in Computer Science (LICS '03)*, pages 271–280, 2003.
7. E. Contejean and H. Devie. An efficient algorithm for solving systems of Diophantine equations. *Information and Computation*, 113(1):143–172, 1994.
8. N. Durgin, P.D. Lincoln, J.C. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Proc. FLOC Workshop on Formal Methods in Security Protocols*, 1999.
9. M. Fiore and M. Abadi. Computing symbolic models for verifying cryptographic protocols. In *Proc. 14th IEEE Computer Security Foundations Workshop*, pages 160–173, 2001.
10. D. Kapur, P. Narendran, and L. Wang. An e-unification algorithm for analyzing protocols that use modular exponentiation. In *Proc. 14th International Conference on Rewriting Techniques and Applications (RTA '03)*, volume 2706 of *LNCS*, pages 165–179, 2003.
11. C. Meadows and P. Narendran. A unification algorithm for the group Diffie-Hellman protocol. In *Proc. Workshop of Issues in Theory of Security (WITS)*, 2002.
12. A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
13. J. Millen and V. Shmatikov. Constraint solving for bounded process cryptographic protocol analysis. In *Proc. 8th ACM Conference on Computer and Communications Security (CCS '01)*, pages 166–175, 2001.
14. J. Millen and V. Shmatikov. Symbolic protocol analysis with products and Diffie-Hellman exponentiation. In *Proc. 16th IEEE Computer Security Foundations Workshop*, pages 47–61, 2003.
15. L. Paulson. Mechanized proofs for a recursive authentication protocol. In *Proc. 10th IEEE Computer Security Foundations Workshop*, pages 84–95, 1997.
16. O. Pereira and J.-J. Quisquater. A security analysis of the Cliques protocols suites. In *Proc. 14th IEEE Computer Security Foundations Workshop*, pages 73–81, 2001.
17. M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proc. 14th IEEE Computer Security Foundations Workshop*, pages 174–190, 2001.
18. P. Ryan and S. Schneider. An attack on a recursive authentication protocol: A cautionary tale. *Information Processing Letters*, 65(1):7–10, 1998.
19. M. Steiner, G. Tsudik, and M. Waidner. Diffie-Hellman key distribution extended to group communication. In *Proc. 3rd ACM Conference on Computer and Communications Security (CCS '96)*, pages 31–37, 1996.