Associative-Commutative Deducibility Constraints*

Sergiu Bursuc[1], Hubert Comon-Lundh[1], and Stéphanie Delaune[1,2]

[1] Laboratoire Spécification & Vérification, ENS de Cachan & CNRS, France
{bursuc,comon,delaune}@lsv.ens-cachan.fr
[2] School of Computer Science, University of Birmingham, UK

Abstract. We consider deducibility constraints, which are equivalent to particular Diophantine systems, arising in the automatic verification of security protocols, in presence of associative and commutative symbols. We show that deciding such Diophantine systems is, in general, undecidable. Then, we consider a simple subclass, which we show decidable. Though the solutions of these problems are not necessarily semi-linear sets, we show that there are (computable) semi-linear sets whose minimal solutions are not too far from the minimal solutions of the system. Finally, we consider a small variant of the problem, for which there is a much simpler decision algorithm.

1 Introduction

During the past ten years there has been a lot of work devoted to the logical verification of security protocols (as opposed to computational security). In general, the security problem is undecidable. Many authors designed subclasses for which there are decision algorithms (*e.g.* [14, 1, 5, 12]). However, these techniques assume most of the time a *perfect cryptography*: the algebra of messages must be a free algebra. On the other hand, many protocols rely on algebraic properties of some primitives, for instance the Abelian Groups properties of the multiplication of exponents in modular exponentiation or the associativity, commutativity and nilpotence of exclusive or. Sometimes, the protocols cannot be executed when these properties are not considered.

Another research direction consists, instead of considering subclasses of protocols, to bind the search for an attack to a limited number of protocol instances. We get a reasonable confidence in the protocol security if we show that, say, after any 10 plays of the protocol there is no attack. In the free algebra case, the security problem for such a fixed number of sessions is co-NP-complete [15]. This result has then been extended beyond the perfect cryptography assumption: for exclusive or [7, 3], for some properties of modular exponentiation [2, 13], for properties combining exclusive or and homomorphisms [10] and other combinations [4].

* This work was partly supported by the RNTL project PROUVÉ 03V360, ACI-SI Rossignol, and EPSRC project EP/E029833.

However, it seems that every new protocol comes with its own relevant algebraic properties. That is what happened to us when we were faced to a case study of electronic purse submitted to us by France Télécom (see [9]). In [6] two of us gave a result allowing to reduce many equational theories to associativity and commutativity only, at the price of considering many particular instances of the protocol. In particular, the above-mentioned properties of modular exponentiation, the exclusive-or properties, and the properties used in the electronic purse protocol can be reduced to associativity and commutativity alone. Now, this raises the problem of designing an algorithm solving the security problem in an algebra of messages, modulo associativity and commutativity. This were the reasons why we came to the problem, which is studied in this paper.

We consider the simplest instance of the problem (and we hope to be able to reduce many other situations to this one using combination techniques such as those described in [4]). A (AC)-*deducibility constraint* is a conjunction of expressions $T \Vdash u$ where u is a simple term and T is a finite set of simple terms. A simple term is an expression $\lambda_1 x_1 + \ldots + \lambda_k x_k + w$ where $\lambda_1, \ldots, \lambda_k \in \mathbb{N}$ and $w \in \mathbb{N}^m$. A solution of $T \Vdash u$ is an assignment σ of the variables $x_1, \ldots, x_k$ to vectors of $\mathbb{N}^m$ such that there is a linear combination of the simple terms $t\sigma$ of T, which equals $u\sigma$. It is not difficult to see that such problems can be reduced to (non-linear) Diophantine systems. Unfortunately, the converse is also true: AC-deducibility constraints are undecidable. Then, we consider a subclass of such constraints: first we assume a (classical) monotonicity condition between the sets T_i, which corresponds to the increasing of intruder's knowledge. We also assume that every term u in a constraint $T \Vdash u$ contains at most one variable. This ensures the determinacy of protocol executions, but there might be weaker conditions which ensure both determinacy and decidability. The core of our paper is a decision algorithm for this class of Diophantine equations.

Our decidability proof works as follows. We first define a relation between constraints, which is derived from an occurrence relation on variables, and consider the strongly connected components for this relation. In each of such components we show that, if there is a solution, then there is one, which is not far (w.r.t. Euclidian distance) of a solution of some (finitely many) computable semilinear sets. Then the last step consists in proving that the restrictions of minimal solutions of the whole system to minimal strongly connected components are not far from minimal solutions of the minimal strongly connected components. This allows to derive an algorithm (in NEXPTIME), which solves our constraints.

In a last part of the paper, we consider another interpretation of the constraint system. In this interpretation, the intruder is not only allowed to add messages, but also to substract them. In this interpretation, we show that there is a much simpler decision procedure.

2 AC-deducibility constraints & Diophantine equations

As explained in the introduction, several algebraic properties have been studied recently [7, 3, 2]. In [6], we gave a result allowing us to reduce many relevant equational theories to associativity and commutativity only. In this section, we focus on this particular equational theory. We consider simple messages built from constants, variables and the symbol + only. Moreover, the only intruder capability consists in adding messages. We believe that many other intruder inference systems can be reduced to this simple one, by combination techniques.

2.1 Basic definitions

Let $\mathcal{A}$ be a finite set of constants, $\mathcal{X}$ be a set of variable symbols disjoint from $\mathcal{A}$ and $+$ be an associative and commutative (AC) symbol, which will be assumed later to have a neutral element 0. *Terms* are expressions

$$n_1 \cdot X_1 + \ldots + n_k \cdot X_k + m_1 \cdot a_1 + \ldots + m_l \cdot a_l$$

where $n_1, \ldots, n_k, m_1, \ldots, m_l$ are positive (non null) integers, $X_1, \ldots, X_k$ are distinct variable symbols and $a_1, \ldots, a_l$ are distinct constant symbols. Other writings of terms (*e.g.* with repeated variables or constants or with some null coefficients) are normalized into the above canonical form.

Let t be a term and u be a constant or a variable. The *number of occurrences* of u in t, denoted by $|t|_u$, is 0 if u does not occur in t, and the coefficient of u in t otherwise. $|t|_{\max}$ the integer $\max(\{|t|_a \mid a \in \mathcal{A}\})$ and by $t^{\mathcal{X}}$ (resp. t^0) the term such that $|t^{\mathcal{X}}|_a = 0$ for every $a \in \mathcal{A}$ (resp. $|t^0|_X = 0$ for every $X \in \mathcal{X}$) and $t = t^{\mathcal{X}} + t^0$. A *ground term* t is a term such that $t^0 = t$. It can also be viewed as a vector of non-negative integers, whose dimension is $|\mathcal{A}|$. A *substitution* (resp. a *ground substitution*) is a mapping from a finite subset of $\mathcal{X}$, called its *domain* to the set of terms (resp. ground terms). Substitutions are extended, as usual, to endomorphisms of the term algebra. We write $\{X_1 \mapsto t_1, \ldots, X_p \mapsto t_p\}$ the substitution σ whose domain is $\{X_1, \ldots, X_p\}$ and such that, $\forall i = 1, .., p, X_i\sigma = t_i$. A ground substitution $\{X_1 \mapsto t_1, \ldots, X_p \mapsto t_p\}$ can be represented by a p-columns matrix whose i^{th} column is $X_i\sigma$.

Example 1. Let $\mathcal{A} = \{a, b, c, d\}$ and let t be the ground term $2a + b + c$. The representation of t as a vector is described below. We have that $|t|_{\max} = 2$. The substitution $\sigma = \{X_1 \mapsto t, X_2 \mapsto 2d\}$ can be represented as follows.

$$t := \begin{pmatrix} 2 \\ 1 \\ 1 \\ 0 \end{pmatrix} \qquad \sigma := \begin{pmatrix} 2 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 2 \end{pmatrix}$$

$\mathbb{N}^n$ is ordered with the product ordering: if $\Lambda = (\lambda_1, \ldots, \lambda_n), \Lambda' = (\lambda'_1, \ldots, \lambda'_n) \in \mathbb{N}^n$, $\Lambda \leq \Lambda'$ if and only if $\forall i \in \{1..n\}, \lambda_i \leq \lambda'_i$. $\Lambda < \Lambda'$ if and only if $\Lambda \leq \Lambda'$ and $\Lambda \neq \Lambda'$. This ordering is a well-ordering: in any infinite sequence of vectors, there is an infinite increasing subsequence. Following the vector representation of ground terms, this ordering is also used to compare ground terms. For instance $a + b < 2a + b + c$. It can also be extended to ground substitutions: $\sigma \leq \sigma'$ iff $dom(\sigma) = dom(\sigma')$ and $\forall X \in dom(\sigma), X\sigma \leq X\sigma'$. The following definition expresses the intruder deduction capabilities: given the messages $t_1, \ldots, t_n$, he is able to build any combination of them:

Definition 1 (AC-deducibility). *Given terms $t_1, \ldots, t_n, u$, we write $t_1, \ldots, t_n \vdash u$ if there are non-negative integers $\lambda_1, \ldots, \lambda_n$ such that $\lambda_1 \cdot t_1 + \ldots + \lambda_n \cdot t_n = u$.*

Example 2. $2a + x, b + x, a + c \vdash 7a + 2x + 3c$ with $\lambda_1 = 2$, $\lambda_2 = 0$ and $\lambda_3 = 3$.

Definition 2 (AC-deducibility constraint). *An AC-deducibility constraint is an expression $T \Vdash u$ where T is a finite set of terms and u is a term. An AC-deducibility constraint system $\mathcal{C}$ is a finite conjunction of such constraints. A ground substitution σ is a* solution *of $t_1, \ldots, t_n \Vdash u$ if its domain contains the variables of $t_1, \ldots, t_n, u$ and $t_1\sigma, \ldots, t_n\sigma \vdash u\sigma$. σ is a solution of a constraint system $\mathcal{C}$ if it is a solution of every individual AC-deducibility constraint.*

The previous definition allows to express the ability to mount an attack in a given number of steps: initially, the intruder knows a finite set of messages T_0 and must be able to build an instance of the message u_1 expected by some honest agent. He replies by sending a corresponding message v_1, which increases the intruder knowledge. Again, from T_0 and v_1, the intruder must be able to build an instance of u_2 and gets the corresponding instance of v_2,... and after n such steps the intruder can deduce a message s, which was supposed to be secret. This translates into the constraint system

$$T_0 \Vdash u_1 \quad \wedge \quad T_0, v_1 \Vdash u_2 \quad \wedge \quad \ldots \quad \wedge \quad T_0, v_1, \ldots, v_n \Vdash s$$

The details of this formalism are reported in many papers (see *e.g.* [15]). Variables in the terms represent the pieces of the messages that cannot be analysed by the agent: it could be nonces or cyphertexts whose key is unknown. The agent will accept any message in place of this variable. That is how many classical logical attacks are constructed: the intruder, at some point, replaces the expected message with a message, which only differs from the correct one in the non-analyzable parts. Hence, finding a solution to the above AC-deducibility constraint system amounts to find (constructible) fake instances allowing to retrieve the secret after n steps.

Example 3. Consider the system $a \Vdash X \;\wedge\; a, X + b \Vdash Y$. Typical solutions of this systems are $\{X \mapsto 2a, Y \mapsto 5a + 2b\}, \{X \mapsto k_1 a; Y \mapsto k_2 k_1 a + k_2 b\}$, with $k_1, k_2 \geq 0$.

Putting together Definitions 1 and 2, we get the following problem, whose decision is the subject of this paper:

> Given an AC-deducibility constraint system $T_1 \Vdash u_1, \ldots T_n \Vdash u_n$, does there exist a substitution σ such that,
>
> $$\exists (\lambda_{i,t})_{i \in \{1..n\}, t \in T_i} \in \mathbb{N}^{|T_1| + \ldots + |T_n|}. \bigwedge_{i=1}^{n} \sum_{t \,\in\, T_i} \lambda_{i,t} t\sigma = u_i \sigma$$

Definition 3 (minimal solution). *Let $\mathcal{C}$ be an AC-deducibility constraint system and σ be a solution to $\mathcal{C}$. σ is a* minimal solution *of $\mathcal{C}$ if for every solution σ' of $\mathcal{C}$, $\sigma' \not< \sigma$.*

2.2 From AC-deducibility constraints to Diophantine equations ...

In the above problem, if the set of constants is $\{a_1, \ldots, a_m\}$, the equality holds iff the coefficients of every a_i are identical on both sides. Assuming that $X_1, \ldots, X_p$ are the variables of the system $\mathcal{C}$ of n AC-deducibility constraints, $\mathcal{A} = \{a_1, \ldots, a_m\}$ and σ is a solution to $\mathcal{C}$, we consider the variables $x_{i,j}$, representing the coefficient of a_j in $X_i\sigma$ and the coefficients $\lambda_{k,t}$ of the term $t \in T_k$ in the k^{th} constraint $T_k \Vdash u_k$. Then $\mathcal{C}$ has a solution iff there is a solution to the conjunction, for $k = 1..n, j = 1..m$ of the equations

$$\sum_{t\,\in\, T_k} \lambda_{k,t}(|t|_{a_j} + \sum_{i=1}^{p} |t|_{X_i}\; x_{i,j}) = |u_k|_{a_j} + \sum_{i=1}^{p} |u_k|_{X_i}\; x_{i,j} \tag{1}$$

This is a system of $n \times m$ quadratic Diophantine equations, whose variables are $x_{i,j}, \lambda_{k,t}$. In addition, we add equations ruling out the solutions $X_i = 0$.

Example 4. Consider the constraint system $2a \Vdash X_1 \,\wedge\, 2a,\; X_1 + b \Vdash 3X_2 + a$ and assume that a, b are the only two constants. This constraint can be translated into the equivalent Diophantine system

$$\exists\, \lambda_{1,1}, \lambda_{2,1}, \lambda_{2,2}. \;\left\{\begin{array}{r} 2\lambda_{1,1} = x_{1,1} \\ 0 = x_{1,2} \\ 2\lambda_{2,1} + \lambda_{2,2}x_{1,1} = 3x_{2,1} + 1 \\ \lambda_{2,2}x_{1,2} + \lambda_{2,2} = 3x_{2,2} \end{array}\right.$$

which can also be expressed in matricial notation:

$$\exists\, \Lambda. \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 2 & x_{1,1} \\ 0 & 0 & x_{1,2}+1 \end{pmatrix} \cdot \Lambda = \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ 3x_{2,1}+1 \\ 3x_{2,2} \end{pmatrix}$$

We also ensure that $x_{1,1} \neq 0$ and $x_{2,1}, x_{2,2}$ are not both 0, adding $x_{1,1} = 1 + z_1$ and $x_{2,1} + x_{2,2} = 1 + z_2$.

This shows that we can reduce our problem to some Diophantine systems. Such systems seem to be particular ones. Unfortunately, this is not true, as we show in the next section.

2.3 ... and back

We show here that we can also go back from Diophantine systems: any Diophantine system can be encoded in an AC-deducibility constraint system. Hence:

Proposition 1. *The problem of deciding whether a system of deducibility constraints has a solution is undecidable.*

To prove this result, we use the following formulation of Hilbert's 10^{th} problem, known to be undecidable [8]. Note that we can simulate the product by using the identity $(u + v)^2 = u^2 + v^2 + 2uv$.

INPUT: a finite set S of Diophantine equations where each equation is of the form: $x_i = m$, $x_i + x_{i'} = x_j$, or $x_j = x_i^2$.
OUTPUT: Does S have a solution in $\mathbb{N}$?

Given an instance S of Hilbert's 10^{th} problem with n variables $x_1, \ldots, x_n$ we built an AC-constraint system $\mathcal{C}(S)$, such that S has a solution iff $\mathcal{C}(S)$ has a solution. We use three constants a, b, c and assume first that variables of $\mathcal{C}$ can be mapped to 0 (neutral element). This is not a restriction as we may then reduce the problem to systems of AC-deducibility constraints by guessing first which variables are assigned to 0.

Our encoding. Let n (resp. p) be the number of equations (resp variables) in S. We describe below how we build the first part $\mathcal{A}(p)$ of our constraint system. For every $i = 1, \ldots, p$, the constraint system $\mathcal{A}(p)$ contains the following five deducibility constraints whose free variables are X_i, Y_i, and Z_i:

$$a \Vdash X_i \quad b \Vdash Y_i \quad a \Vdash Z_i \quad a + b \Vdash X_i + Y_i \quad X_i + b \Vdash Z_i + Y_i$$

Lemma 1. *Let $p \in \mathbb{N}$ and σ a solution of $\mathcal{A}(p)$. For $i = 1, ..p$, $|Z_i\sigma|_a = |X_i\sigma|_a^2$.*

The part $\mathcal{B}(S) = \{\mathsf{d}_1, \ldots, \mathsf{d}_n\}$ of our coding which depends on $S = \{\mathsf{e}_1, \ldots, \mathsf{e}_n\}$ and contains one deducibility constraint per equation and is built as follows:

- if e_k is $x_i = m$ then d_k is $X_i + c \Vdash m \cdot a + c$
- if e_k is $x_i + x_{i'} = x_j$ then d_k is $X_i + X_{i'} + c \Vdash X_j + c$,
- if e_k is $x_i = x_j^2$ then d_k is $X_i + c \Vdash Z_j + c$.

Example 5. Let $S_e = \{x_1 = 2,\ x_3 = x_2^2,\ x_2 + x_3 = x_1\}$. We obtain for $\mathcal{B}(S_e)$

$$X_1 + c \Vdash 2a + c \quad X_3 + c \Vdash Z_2 + c \quad X_2 + X_3 + c \Vdash X_1 + c$$

3 A decidable class of AC-deducibility constraints

AC-deducibility constraints are undecidable. However, fortunately, we may impose relevant restrictions on our constraint system.

3.1 Well-formed and simple constraint system

Given a set of terms T, we let T^0 be the ground terms of T and $T^{\mathcal{X}}$ be the non-ground terms of T, so that $T = T^0 \uplus T^{\mathcal{X}}$. The first restriction we consider is *monotonicity*. As we have seen before when we sketched how the security problem is expressed, the left members of the constraints were increasing, w.r.t. inclusion. This is not by chance: this corresponds to the assumption that the intruder never forgets any information. We add now this assumption:

Definition 4. *A system $\mathcal{C}$ is* monotone *(resp.* monotone w.r.t. ground terms*) if its constraints can be ordered $T_1 \Vdash u_1, \ldots, T_n \Vdash u_n$ in such a way that $T_i \subseteq T_{i+1}$ (resp. $T_i^0 \subseteq T_{i+1}^0$) for any i such that $1 \leq i < n$.*

The next restriction corresponds to the way in which variables are bounded: messages $u_1, \ldots, u_n$ are sent in this order over the network. When it is sent, the corresponding instance of u_i is determined (and not before). Hence, each time a variable occurs first in some u_i, it must not have occurred before in some T_j, with $j \leq i$. Furthermore, the protocol must be deterministic: upon reception of u_i (or its instance), the agent must not have any choice in sending its message. For instance, a protocol rule, which, upon receiving $X + Y$, states that X must be replied, is ambiguous on many instances, and cannot be implemented in a reasonable way. Determinacy is ensured by introducing the variables one by one:

Definition 5 (well-formed). *Let $\mathcal{C} = \{T_1 \Vdash u_1, \ldots, T_n \Vdash u_n\}$ be a monotone constraint system. We say that $\mathcal{C}$ is well-formed if it satisfies*

1. *(origination property) for every $i \leq n$ for every $X \in vars(T_i)$, there exists $j < i$ such that $X \in vars(u_j)$. We will write $\min(X)$ the index of the constraint in which X appears for the first time.*
2. *(deterministic) for all $X, Y \in vars(\mathcal{C})$, if $X \neq Y$ then $\min(X) \neq \min(Y)$.*

The notion of origination and the notation $\min(X)$ are defined in a similar way on constraint systems which are monotone w.r.t. ground terms. The hypotheses introduced so far are not considered as real restriction and have already been used in [3, 2]. We will actually require a stronger property, implying determinacy:

Definition 6 (simple). *A deducibility constraint $T \Vdash u$ is said* simple *if u is of the form $\beta X + u_0$ for some variable X, some $\beta \in \mathbb{N}$ and some ground term u_0. A constraint system $\mathcal{C}$ is said simple if all the constraints in $\mathcal{C}$ are simple.*

By convention, $\beta = 0$ means that $u = u^0$ and $u^0 = 0$ means that $u = \beta X$.

Example 6. The system $\mathcal{C}$ described below is simple and well-formed.

$$2a \Vdash X_1 + a \ \ \wedge \ \ 2a,\ X_1 + b \Vdash 3X_2 + b \ \ \wedge \ \ 2a,\ X_1 + b,\ X_2 \Vdash a$$

As in Section 2.2, each ground substitution can be viewed as a $p \times m$ tuple of integers if there are p variables in its domain and m constants. Then, to each AC-deducibility constraint system $\mathcal{C}$, we can associate the set of tuples of integers $S(\mathcal{C})$ corresponding to its set of solutions.

Sets of integers defined by solutions of AC-deducibility constraint systems strictly include semi-linear sets. Using an example similar to Example 3, we can define, using AC-deducibility constraints, the set of triples $\{(u, v, uv{+}w) \mid u, v, w \in \mathbb{N}\}$, which is not semi-linear and might even not be semi-polynomial [11].

The remainder of this section is devoted to the proof of the following theorem.

Theorem 1. *The problem of deciding whether a simple and well-formed AC-deducibility constraint system has a solution is decidable.*

We will allow assignments to 0, a neutral element for $+$. This is not a restriction, since we can force variables to be distinct of 0 by guessing, for each variable of the system, a constant c_X and replacing X with $c_X + X'$ in the system. This replacement preserves simplicity and well-formedness.

3.2 The algorithm

From now, by "constraint system" we mean a simple well-formed AC-deducibility constraint system.

First step. In a first phase, given a constraint system $T_i \Vdash u_i$, we guess what are the useful terms $\mathcal{U}_i$ in each $T_i^{\mathcal{X}}$, *i.e.* we guess which coefficients are assigned 0.

Definition 7 (solution compatible with $\mathcal{U}$). *Let $\mathcal{C} = \{T_1 \Vdash u_1, \ldots, T_n \Vdash u_n\}$ be a constraint system. Let $\mathcal{U}$ be the sequence $(\mathcal{U}_1, \ldots, \mathcal{U}_n)$ with $\mathcal{U}_i \subseteq T_i^{\mathcal{X}}$. Let σ be a solution of $\mathcal{C}$. We say that σ is* compatible with $\mathcal{U}$ *if there exists a tuple of $\lambda_{k,t} \in \mathbb{N}$, one for each $k \in \{1, ..., n\}, t \in T_k$ such that:*

$$\forall k \leq n. \begin{cases} \sum_{t \in T_k} \lambda_{k,t} t\sigma = u_k\sigma \\ \forall t \in T_k^{\mathcal{X}}.\ \lambda_{k,t} \neq 0 \quad \Leftrightarrow t \in \mathcal{U}_k \end{cases}$$

Example 7. Consider the constraint system $\mathcal{C}$ described in Example 6. Let $\mathcal{U}$ be the sequence $(\emptyset, \{X_1 + b\}, \{X_2\})$. Consider the substitution $\sigma = \{X_1 \mapsto a, X_2 \mapsto a\}$. We claim that σ is a solution of $\mathcal{C}$ compatible with $\mathcal{U}$. Indeed, we may choose $\lambda_{1,2a} = \lambda_{2,2a} = \lambda_{2,X_1+b} = \lambda_{3,X_2} = 1$ and $\lambda_{i,t} = 0$ otherwise.

Second step. This step consists in constructing a dependency graph on variables:

Definition 8. *Let $\mathcal{C} = \{T_1 \Vdash u_1, \ldots, T_n \Vdash u_n\}$ be a constraint system. Let $\mathcal{U}$ be a sequence $(\mathcal{U}_1, \ldots, \mathcal{U}_n)$ with $\mathcal{U}_i \subseteq T_i^{\mathcal{X}}$. The relation $\mathcal{R}_{\mathsf{occ}}^{\mathcal{U}}$ on $vars(\mathcal{C})$ is defined by:*

$$X\ \mathcal{R}_{\mathsf{occ}}^{\mathcal{U}}\ Y \Leftrightarrow\ \exists i. \begin{cases} Y \in vars(u_i), \text{ and} \\ X \in vars(t) \text{ for some term } t \in \mathcal{U}_i. \end{cases}$$

We consider the equivalence relation $=_{\mathsf{occ}}^{\mathcal{U}}$. We have $X =_{\mathsf{occ}}^{\mathcal{U}} Y$ if, and only if, $X \prec_{\mathsf{occ}}^{\mathcal{U}} Y$ and $Y \prec_{\mathsf{occ}}^{\mathcal{U}} X$ where $\prec_{\mathsf{occ}}^{\mathcal{U}}$ is the transitive closure of $\mathcal{R}_{\mathsf{occ}}^{\mathcal{U}}$. We denote by $[=_{\mathsf{occ}}^{\mathcal{U}}]$ the equivalence classes induced by $=_{\mathsf{occ}}^{\mathcal{U}}$. $\prec_{\mathsf{occ}}^{\mathcal{U}}$ is then an ordering on $[=_{\mathsf{occ}}^{\mathcal{U}}]$. In the last example, $X_1 \prec_{\mathsf{occ}}^{\mathcal{U}} X_2$ and $[=_{\mathsf{occ}}^{\mathcal{U}}] = \{\{X_1\}, \{X_2\}\}$.

Third step. Now, we choose one of the minimal classes (minimal strongly connected component in the graph) and solve the subsystem consisting of variables in that class.

Definition 9. *Let $\mathcal{C} = \{T_1 \Vdash \beta_1 X_1 + u_1, \ldots, T_n \Vdash \beta_n X_n + u_n\}$ be a simple constraint system. Let $\mathcal{U}$ be the sequence $(\mathcal{U}_1, \ldots, \mathcal{U}_n)$ with $\mathcal{U}_i \subseteq T_i^{\mathcal{X}}$ and M be a minimal class of $[=_{\mathsf{occ}}^{\mathcal{U}}]$. We let $\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})$ be the constraint system defined as follows.*

$$\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U}) = \{T_i^0 \cup \mathcal{U}_i \Vdash \beta_i X_i + u_i \mid X_i \in \mathsf{M}\}.$$

Example 8. Consider the system $\mathcal{C}$ described in Example 6 and the sequence $\mathcal{U}$ described in Example 7. We have $\mathsf{M} = \{X_1\}$ and $\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U}) = \{2a \Vdash X_1 + a\}$.

We first show that this subsystem inherits good properties of the original system. However, note that $\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})$ is not always monotone.

Lemma 2. *Let $\mathcal{C} = \{T_1 \Vdash u_1, \ldots, T_n \Vdash u_n\}$ be a simple and well-formed constraint system. Let $\mathcal{U}$ be the sequence $(\mathcal{U}_1, \ldots, \mathcal{U}_n)$ with $\mathcal{U}_i \subseteq T_i^{\mathcal{X}}$ and M be a minimal class of $[=^{\mathcal{U}}_{\mathsf{occ}}]$. The minimal component $\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})$ is simple, monotone w.r.t. ground terms and satisfies the origination property.*

Then, we prove that the set of solutions of $\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})$ can be represented by a semi-linear set. We also show that we can compute a bound $\delta(\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U}))$ on minimal solutions of such systems. This is detailed in section 3.3.

Fourth step. We prove (this is the subject of section 3.4) that a minimal solution of the system is "not far" from a minimal solution of $\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})$, for which we computed a bound. Then we guess a partial substitution, on the variables of the minimal class, within the computed bound, and replace it in the system. At this point, we eliminated at least one variable, while keeping the set of minimal solutions. We only have to iterate the process, until all variables are eliminated.

Summary of the procedure.
Let $\mathcal{C} = \{T_1 \Vdash u_1, \ldots, T_n \Vdash u_n\}$ be a simple and well-formed constraint system.

1. Guess $\mathcal{U}$
2. Compute $[=^{\mathcal{U}}_{\mathsf{occ}}]$
3. While $[=^{\mathcal{U}}_{\mathsf{occ}}] \neq \emptyset$:
   (a) extract a system $\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})$
   (b) Compute the minimal solutions S of this system
   (c) Guess a replacement in $\mathcal{C}$ of the variables of $\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})$, which is at a distance bounded by δ from a solution in S.

This procedure yields a finite set of AC-deducibility constraint systems in which every term is ground. The satisfiability of such systems can be decided in non-deterministic polynomial time by reducing it to linear Diophantine equations.

3.3 The case of a strongly connected variable graph

In this section, we show that the solutions of $\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})$ is a semi-linear set.

Lemma 3. *Let $\mathcal{C}' = \mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U}) = \{T_1 \Vdash \beta_1 X_1 + u_1, \ldots, T_n \Vdash \beta_n X_n + u_n\}$ There exists a bound $\eta(\mathcal{C}') \in \mathbb{N}$, effectively computable from $\mathcal{C}'$, such that for every solution σ of $\mathcal{C}'$ compatible with $(T_1^{\mathcal{X}}, \ldots, T_n^{\mathcal{X}})$, there exist a tuple of $\lambda_{i,t} \in \mathbb{N}$, one for each $k \in \{1, \ldots, n\}, t \in T_i$, such that:*

$$(t \in T_i^{\mathcal{X}} \Rightarrow \lambda_{i,t} \leq \eta(\mathcal{C}')) \wedge \sum_{t \in T_i} \lambda_{i,t} t\sigma = \beta_i X_i \sigma + u_i$$

To prove this lemma, we simply use, for each variable, a non trivial cycle on it in the graph $\mathcal{R}^{\mathcal{U}}_{occ}$. Assuming X occurs in t and putting together all inequalities along the cycle we get $\lambda_{i,t} \leq (\prod_{j \in c} \beta_j)(1 + \sum_{j \in c} |u_j|_{\max})$ where c is the cyclic sequence of indices starting from i.

Now the coefficients of non-ground terms are bounded, we are back to linear Diophantine systems: we can guess a value $r_{i,t}$, within the above-computed bound, for the coefficients $\lambda_{i,t}$ when t is not ground, and get an equivalent (disjunction of) systems

$$\bigwedge_{i=1}^{n} \sum_{t \in T_i^0} \lambda_{i,t} t + \sum_{t \in T_i^{\mathcal{X}}} r_{i,t} t = \beta_i X_i + u_i \tag{2}$$

whose variables are the remaining $\lambda_{i,t}$ and the X_i. We also let the homogeneous system be:

$$\bigwedge_{i=1}^{n} \sum_{t \in T_i^0} \lambda_{i,t} t + \sum_{t \in T_i^{\mathcal{X}}} r_{i,t} t^{\mathcal{X}} = \beta_i X_i \tag{3}$$

Lemma 4. *For the system $\mathcal{C}' = \mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})$, the solutions of (1) form a semi-linear set. Given an assignment θ of the $\lambda_{i,t}$ (t non ground) to $r_{i,t}$, we let $\Sigma_0(\theta)$ be the minimal solutions of (2) and $\Sigma_h(\theta)$ be the minimal non-null solutions of (3). Each solution of $\mathcal{C}'$ assigns the $\lambda_{i,t}$, for t not ground, to some $r_{i,t} \leq \eta(\mathcal{C}')$, which defines a substitution θ. Then the remaining solutions assign the variables to the vectors $V_0 + \sum_{i=1}^{N} \mu_i V_i$ for $V_0 \in \Sigma_0(\theta)$, $\Sigma_h(\theta) = \{V_1, \ldots, V_N\}$ and $\mu_1, \ldots, \mu_N$ are arbitrary non-negative integers.*

For the next step, we need to compute a distance within which the restriction of the minimal solutions of $\mathcal{C}$ to variables of $\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})$ lie. Then let

$$\delta(\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U}),\theta) = \sum_{\sigma \,\in\, \Sigma_0(\theta)} \sigma \;+\; \beta(\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})) \cdot \sum_{\sigma \,\in\, \Sigma_h(\theta)} \sigma$$

where $\beta(\mathcal{C}) = \prod_{X \in vars(\mathcal{C})} \beta_{\min(X)}$. Let $\delta(\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})) = \max_{\theta \leq \theta^0}(\delta(\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U}),\theta))$ where θ^0 assigns $\eta(\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U}))$ to all variables.

3.4 The projections of global minimal solutions are not far from minimal solutions of the minimal classes

In this section, we show that if a simple and well-formed constraint system $\mathcal{C}$ has a solution compatible with a given sequence $\mathcal{U}$, then there is one such σ satisfying $\sigma|_{\mathsf{M}} \leq \delta(\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U}))$. The proof relies on the above bound,

Proposition 2. *Let $\mathcal{C} = \{T_1 \Vdash \beta_1 X_1 + u_1, \ldots, T_n \Vdash \beta_n X_n + u_n\}$ be a simple and well-formed constraint system. Let $\mathcal{U}$ be the sequence $(\mathcal{U}_1, \ldots, \mathcal{U}_n)$ with $\mathcal{U}_i \subseteq T_i^{\mathcal{X}}$ and M be a a minimal class of $[=_{\mathsf{occ}}^{\mathcal{U}}]$. If σ is a minimal solution of $\mathcal{C}$ compatible with $\mathcal{U}$ then $\sigma|_{\mathsf{M}} \;\leq\; \delta(\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U}))$.*

This also concludes the proof of the main theorem: the algorithm is roughly described in Section 3.2 and we can now complete the last step of the loop: we guess an assignment $\sigma|_M$ of the variables of $\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U})$, within a finite set, bounded by $\delta(\mathscr{C}_{\mathsf{M}}(\mathcal{C},\mathcal{U}))$.

4 Another deducibility system

In this section, we consider again AC-deducibility constraint systems, but with a different interpretation of the deducibility relation. More precisely, we keep the same definitions as in Section 2, except for Definition 1 which becomes:

Definition 10. *$t_1, \ldots, t_n \vdash u$ iff $\exists \lambda_1, \ldots, \lambda_n \in \mathbb{Z}$ such that $\sum_{i=1}^{n} \lambda_i t_i = u$.*

The main difference is now the ability of using negative coefficients. Note however that we do not have any opposite symbol: variables can only be substituted by positive combinations of constants. This new inference system, denoted by $\mathcal{I}_{\pm}$, allows us to obtain a procedure, which is simpler than the one presented in the previous section and also to deal with a broader class of constraint systems. Moreover, the additional capabilities given to the attacker through this inference system is realistic in most of our applications.

Theorem 2. *The problem of whether a well-formed constraint system has a solution w.r.t. $\mathcal{I}_{\pm}$ is decidable.*

Note that we allow here right hand sides to contain more than one variable. For the proof of this theorem, we perform a variable elimination. Considering a minimal constraint (w.r.t. inclusion of left hand sides), it has the form $T \Vdash \beta X + u_0$: by origination, T can only contain ground terms and, by determinacy, the right hand side contains at most one variable. We eliminate X by showing that there is a bound on the coefficients of $t \in T$ in a solution:

Lemma 5. *Let $\mathcal{C}$ be a well-formed satisfiable (w.r.t. $\mathcal{I}_{\pm}$) constraint system. Let $T \Vdash \beta X + u_0$ be a constraint of $\mathcal{C}$ with a minimal left hand side, with $\beta \in \mathbb{N}$ and u_0 a ground term. Then there exists a solution σ of $\mathcal{C}$ and coefficients $\lambda_{1,t}$ for $t \in T$ such that $\sum_{t \in T} \lambda_{1,t} t = \beta X\sigma + u_0$ and $\forall t \in T. 0 \leq \lambda_{1,t} \leq \beta + |u_0|_{\max}$.*

This lemma heavily relies on the ability to substract. For instance, if the coefficient of some t is negative in a solution, then we increase the coefficient and the corresponding value of X. Then, this is compensated in other constraints by substracting to coefficients what is added by the new contribution of X.

5 Conclusion

We have shown the decidability of two deducibility constraint systems modulo associativity and commutativity. These results are a first step towards a general decision procedure for security protocols in a bounded number of sessions. Our results have several weaknesses. The first one is algorithmic complexity. An analysis of the algorithms show that they are in NEXPTIME. It is not clear whether this would be applicable in practice. There is a hope still, since security protocols are in general very small (up to 6 protocol rules). Only an implementation would prove the usefulness of the method. There is however a long way before implementing the techniques. We need first to establish a combination result, which would allow to handle more complicated constructions and inference systems. The last weakness of our results is the additional condition (right hand side only contain one variable) we have in Theorem 1. It is not clear that it is necessary. Though protocols generally satisfy this condition, it might not be the case for the constraints which are computed using the procedure of [6].

References

1. B. Blanchet and A. Podelski. Verification of Cryptographic Protocols: Tagging Enforces Termination. *Theoretical Computer Science*, 333(1-2):67–90, 2005.
2. Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the security of protocols with Diffie-Hellman exponentiation and product in exponents. In *Proc. 23rd Conf. on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'03)*, vol. 2914 of *LNCS*, pages 124–135. Springer-Verlag, 2003.
3. Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with XOR. In *Proc. 18th IEEE Symp. Logic in Computer Science (LICS'03)*, pages 261–270. IEEE Comp. Soc. Press, 2003.
4. Y. Chevalier and M. Rusinowitch. Hierarchical combination of intruder theories. In *Proc. 17th International Conference on Rewriting Techniques and Applications (RTA'06)*, volume 4098 of *LNCS*, pages 108–122. Springer, 2006.
5. H. Comon and V. Cortier. Tree automata with one memory, set constraints and cryptographic protocols. *Theoretical Computer Science*, 331(1):143–214, 2005.
6. H. Comon-Lundh and S. Delaune. The finite variant property: How to get rid of some algebraic properties. In *Proc. 16th Int. Conf. on Rewriting Techniques and Applications (RTA'05)*, vol. 3467 of *LNCS*, pages 294–307. Springer, 2005.
7. H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proc. 18th IEEE Symp. Logic in Computer Science (LICS'03)*, pages 271–280. IEEE Comp. Soc. Press, 2003.
8. M. Davis, Y. Matijasevich, and J. Robinson. Hilbert's tenth problem, Diophantine equations: positive aspects of a negative solution. In *Proc. of Symposia in Pure Maths*, pages 323–378, 1976.
9. S. Delaune. *Vérification des protocoles cryptographiques et propriétés algébriques.* Thèse de doctorat,ENS Cachan, France, 2006.
10. S. Delaune, P. Lafourcade, D. Lugiez, and R. Treinen. Symbolic protocol analysis in presence of a homomorphism operator and *exclusive or*. In *Proc. 33rd International Colloquium on Automata, Languages and Programming (ICALP'06)*, volume 4052 of *LNCS*, pages 132–141. Springer, 2006.
11. W. Karianto, A. Krieg, and W. Thomas. On intersection problems for polynomially generated sets. In *Proc. 33rd International Colloquium on Automata, Languages and Programming (ICALP'06)*, volume 4052 of *LNCS*. Springer, 2006.
12. G. Lowe. Towards a completeness result for model checking of security protocols. *J. Computer Security*, 7(2–3):89–146, 1999.
13. J. Millen and V. Shmatikov. Symbolic protocol analysis with an abelian group operator or Diffie-Hellman exponentiation. *J. Computer Security*, 13(3):515–564, 2005.
14. R. Ramanujam and S. P. Suresh. Decidability of context-explicit security protocols. *J. Computer Security*, 13(1), 2005.
15. M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proc.14th IEEE Computer Security Foundations Workshop*, 2001.