Stéphanie Delaune

An Undecidability Result for AGh

Research Report LSV-LSV-06-02

February 2006

Laboratoire Spécification et Vérification

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

An Undecidability Result for AGh *

Stéphanie Delaune

France Télécom, Division R&D
Laboratoire Spécification & Vérification, CNRS UMR 8643, ENS de Cachan

Abstract. We present an undecidability result for the verification of security protocols. Since the *perfect cryptography assumption* is unrealistic for cryptographic primitives with visible algebraic properties, several recent works relax this assumption, allowing the intruder to exploit these properties. We are interested in the *Abelian groups* theory in combination with the homomorphism axiom. We show that the security problem for a bounded number of sessions (expressed by satisfaisability of symbolic deducibility constraints) is undecidable, obtaining in this way the first undecidability result concerning a theory for which unification is known to be decidable.

1 Introduction

Cryptographic protocols are small programs designed to ensure secure communication via a public channel. Many works have been devoted to the use of formal methods in order to automate the proof or the absence of logical attacks on such protocols (*e.g.* [7]). The problem of deciding whether a protocol is secure or not is known to be undecidable in general, even under several restrictions [1, 5, 13]. An interesting decidability result has been obtained by Rusinowitch and Turuani [19], under the assumption that the number of sessions (*i.e.* the number of parallel role instances) is bounded.

In their setting, logical attacks can be characterized by sequences of abstract messages exchanged by honest agents executing the protocol, and by the intruder. Since we consider a bounded number of sessions, there is only a bounded number of symbolic traces. The idea of the algorithm is to guess a symbolic trace in which the messages are represented by terms containing variables. This symbolic trace corresponds to a concrete execution trace if the variables can be instantiated in such a way that, at every moment, a message received by an agent can be deduced by the intruder from the messages seen before. Hence, verifying security of a protocol amounts to a non-deterministic guessing of the symbolic trace plus the resolution of a system of symbolic *deducibility constraints*.

The pioneer work of Rusinowitch and Turuani [19] relies on the so-called *perfect cryptography assumption*, which states that the cryptographic primitives (encryption, hashing, ...) are perfect and can be treated as black boxes. Since

* This work has been partly supported by the RNTL project PROUVÉ 03V360 and the ACI-SI Rossignol.

then, a recent research direction consists in relaxing this assumption by taking into account algebraic properties such as *exclusive or*, *Abelian groups*... Several decision procedures, relying on the constraint solving approach, have been proposed [16, 3, 6, 4, 17, 11]. Moreover, it is well-known that the equational theories we can hope to handle are those for which unification is decidable (*e.g.* [8]). It is also well-admitted that this restriction is not sufficient although, as far as we know, no counterexample has been exhibited.

In this paper, we study the equational theory AGh, whose unification problem is known to be decidable [2], and we prove that the security problem for a bounded number of sessions is undecidable.

2 Preliminaries

2.1 Terms

We use classical notation and terminology on terms, unification and rewriting systems. We write $\mathcal{T}(\mathcal{F}, \mathcal{X})$ for the set of terms built over the finite (ranked) alphabet $\mathcal{F}$ of function symbols and the set of variable symbols $\mathcal{X}$. The set $\mathcal{T}(\mathcal{F}, \emptyset)$ of ground terms (terms without variables) is also written $\mathcal{T}(\mathcal{F})$. The set $\mathcal{F}$ is partitioned into a subset $\mathcal{PF}$ of *private* functions symbols, and a subset $\mathcal{VF}$ of *visible* or *public* functions symbols and we assume that $\mathcal{VF}$ contains classical symbols such as pairing $\langle ., . \rangle$, encryption $\{.\}_.$ and some others such as 0, $h(.)$, $-.$ and $. + .$ related to the equational theory studied in this paper.

The set of positions of a term t is written $\mathcal{O}(t)$. The subterm of $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ at position $p \in \mathcal{O}(t)$ is written $t|_p$. The term obtained by replacing $t|_p$ with s is denoted $t[s]_p$. We refer to any term u that is the same as t everywhere except below p, *i.e.* such that $u[s]_p = t$, as the *linear context* within which the replacement takes place. More precisely, a linear context is a term u with a distinguished position p. The set of variables occurring in t is denoted $vars(t)$.

A *substitution* σ is a mapping from a finite subset of $\mathcal{X}$ called its domain, and written $dom(\sigma)$, to $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Substitutions are extended to endomorphisms of $\mathcal{T}(\mathcal{F}, \mathcal{X})$ as usual. We use a postfix notation for their application.

2.2 Equational Theory

An equational theory E is a set of equations (unordered pairs of terms), we denote by $sig(\mathsf{E})$ the set of all function symbols occurring in E. An E-*context* is a λ-term $\lambda x_1, \ldots, x_n.t$ with $t \in \mathcal{T}(sig(\mathsf{E}), \{x_1, \ldots, x_n\})$, also written $t[x_1, \ldots, x_n]$. The application of a context $t[x_1, \ldots, x_n]$ to arguments $u_1, \ldots, u_n$ is written $t[u_1, \ldots, u_n]$. We denote by $=_\mathsf{E}$ the least congruence on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $u\sigma =_\mathsf{E} v\sigma$ for all pairs $u = v \in \mathsf{E}$ and substitutions σ.

In this paper, we focus on the equational theory $\mathsf{E} = \mathsf{AGh}$, *i.e.* the homomorphism axiom (h), $h(x+y) = h(x)+h(y)$, in combination with the Abelian group theory (AG):

- Associativity-Commutativity (AC): $x + (y + z) = (x + y) + z$, $x + y = y + x$,

– Unit (U): $x + 0 = x$,
– Inverse (Inv): $x + -(x) = 0$.

Let $n \in \mathbb{N}$. The notation $h^n(t)$ (resp. nt) represents the term t (resp. 0) if $n = 0$, and $h(h^{n-1}(t))$ (resp. $t + (n-1)t$) otherwise. Lastly, $-nt$ represents the term $n(-t)$.

2.3 Term Rewriting System

A *term rewriting system* (TRS) is a finite set of *rewrite rules* $l \to r$ where $l \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $r \in \mathcal{T}(\mathcal{F}, vars(l))$. Given a TRS $\mathcal{R}$ and a set of equations E, the relation $\to_{\mathcal{R}/\mathsf{E}}$ (*rewriting modulo* E) is defined as follows: $s \to_{\mathcal{R}/\mathsf{E}} t$ if and only if $s =_{\mathsf{E}} u[l\sigma]_p$ and $u[r\sigma]_p =_{\mathsf{E}} t$, for some linear context u, position p in u, rule $l \to r \in \mathcal{R}$, and substitution σ. The rewrite system is $\mathcal{R}/\mathsf{E}$ is *strongly terminating* if there is no infinite chains $t_1 \to_{\mathcal{R}/\mathsf{E}} t_2 \to_{\mathcal{R}/\mathsf{E}} \dots$ and it is *locally confluent* if for every terms t, s_1 and s_2 such that $t \to_{\mathcal{R}/\mathsf{E}} s_1$, $t \to_{\mathcal{R}/\mathsf{E}} s_2$, there exists a term s such that $s_1 \overset{*}{\to}_{\mathcal{R}/\mathsf{E}} s$, $s_2 \overset{*}{\to}_{\mathcal{R}/\mathsf{E}} s$ where $\overset{*}{\to}_{\mathcal{R}/\mathsf{E}}$ is the reflexive and transitive closure of $\to_{\mathcal{R}/\mathsf{E}}$. A rewrite system $\mathcal{R}/\mathsf{E}$ is said to be E-*convergent* if it is both strongly terminating and locally confluent. A term t is in *normal form* (w.r.t. $\to_{\mathcal{R}/\mathsf{E}}$) if there is no term s such that $t \to_{\mathcal{R}/\mathsf{E}} s$. If $t \overset{*}{\to}_{\mathcal{R}/\mathsf{E}} s$ and s is in normal form then we say that s is a normal form of t. When this form is unique, we write $t \downarrow_{\mathcal{R}/\mathsf{E}}$ or shortly $t \downarrow$ when $\mathcal{R}/\mathsf{E}$ is clear from the context.

We represent the AGh equational theory by an AC-convergent rewrite system. This can be obtained by orienting from left to right the equations (U), (Inv), (h) and by adding the following consequences:

$$\begin{array}{cc} h(0) \to 0 & -(x+y) \to -(x) + -(y) \\ h(-(x)) \to -(h(x)) & -(0) \to 0 \\ & -(-(x)) \to x \end{array}$$

2.4 Factors, Subterms

A term t is *standard* if and only if it is not of the form $f(t_1, \dots, t_n)$ with $f \in sig(\mathsf{E})$. Note that, by definition, every variable is a standard term whereas 0 is not.

Definition 1. *Let t be a term in normal form. We have $t =_{\mathsf{E}} C[t_1, \dots, t_n]$ for some standard terms $t_1, \dots, t_n$ and an E-context C. The set $Fact_{\mathsf{E}}(t)$ of* factors *of t is defined by $Fact_{\mathsf{E}}(t) = \{t_1, \dots, t_n\}$. The set $St_{\mathsf{E}}(t)$ of* subterms *of t is the smallest set such that:*

– $t \in St_{\mathsf{E}}(t)$,
– *if $f(t_1, \dots, t_n) \in St_{\mathsf{E}}(t)$ is standard then $t_1, \dots t_n \in St_{\mathsf{E}}(t)$,*
– *if $s \in St_{\mathsf{E}}(t)$ is not standard then $Fact_{\mathsf{E}}(s) \subseteq St_{\mathsf{E}}(t)$.*

These notations are extended as expected to sets of terms: $Fact_{\mathsf{E}}(T)$ (resp. $St_{\mathsf{E}}(T)$) is the union of the sets $Fact_{\mathsf{E}}(t)$ (resp. $St_{\mathsf{E}}(t)$) for every term t occurring in T. Note that, by definition of $Fact_{\mathsf{E}}$, the factors of any term are necessarily standard.

Example 1. Let $t_1 = 3h^2(a) + -4b + c$, $t_2 = h(\langle -a, b\rangle) + c$ and $t_3 = \langle a + 3b + c, -d\rangle$. We have $Fact_{\mathsf{E}}(t_1) = \{a, b, c\}$, $St_{\mathsf{E}}(t_1) = \{t_1, a, b, c\}$, $Fact_{\mathsf{E}}(t_2) = \{\langle -a, b\rangle, c\}$, and $St_{\mathsf{E}}(t_2) = \{t_2, \langle -a, b\rangle, a, b, c\}$, $Fact_{\mathsf{E}}(t_3) = \{t_3\}$, and $St_{\mathsf{E}}(t_3) = \{t_3, a + 3b + c, -d, a, b, c, d\}$.

We give here some additional definitions in the case of the theory AGh. A polynomial $P(h) \in \mathbb{Z}[h]$ can be written $\sum_{i=0}^{n} c_i h^i$ where $c_i \in \mathbb{Z}$. The product $\odot$ of a polynomial by a term is a term defined as follows:

$$(\sum_{i=0}^{n} c_i h^i) \odot t = \sum_{i=0}^{n} c_i h^i(t).$$

A ground term t such that $Fact_{\mathsf{E}}(t) = \{f_1, \ldots, f_n\}$ can be written $p_{f_1} \odot f_1 + \ldots + p_{f_n} \odot f_n$ for some $p_{f_1}, \ldots, p_{f_n} \in \mathbb{Z}[h]$.

Definition 2. *(number of occurrences) Let t be a ground term and f a ground factor. The number of occurrences of f in t, denoted $\mathcal{N}(f, t)$, is 0 if $f \notin Fact_{\mathsf{E}}(t)$ and $p_f(0)$ otherwise.*

Example 2. Let $p = (3h^2 + -2)$ and $t = a + 2b$. We have:
$p \odot t = 3h^2(a + 2b) + -2(a + 2b) = (3h^2 + -2) \odot a + (6h^2 + -4) \odot b$.
Hence $\mathcal{N}(a, p \odot t) = -2$ and $\mathcal{N}(b, p \odot t) = -4$.

3 Security via Constraint Solving

As explained in the introduction, the symbolic verification of a security protocol can be expressed as a symbolic system of *deducibility constraints* for a certain inference system representing the deduction capabilities of the intruder. More explanations about how to construct the symbolic constraint system from a given protocol can be found in [6, 17]. In this section, we describe the inference systems which are interesting for our purpose, we define precisely our problem and state our undecidability results.

3.1 Intruder Deduction Capabilities

The most widely used deduction relation representing the deduction abilities of an intruder is often referred to as *Dolev-Yao model* [12]. Here we extend the intruder abilities by allowing for equational reasoning modulo a given set E of equational axioms. The deduction system, denoted by $\mathcal{I}_{\mathsf{DY}+\mathsf{E}}$, is given in Figure 1. Equational reasoning is taken into account through the normalization function $\downarrow$ associated to E.

The intended meaning of a *sequent* $T \vdash u$ is that the intruder is able to deduce the term $u \in \mathcal{T}(\mathcal{F})$ from the finite set of terms $T \subseteq \mathcal{T}(\mathcal{F})$. As in the standard Dolev-Yao model, the intruder can compose new terms (C) from known terms, he can also decompose pairs (UL, UR) and decrypt ciphertexts, providing that he can deduce the decryption key (D). Finally, we relax the *perfect cryptography*

$$\text{Unpairing (UL)}\ \frac{T \vdash \langle u, v\rangle}{T \vdash u} \qquad \text{Compose (C)}\ \frac{T \vdash u_1\ \ldots\ T \vdash u_n}{T \vdash f(u_1, \ldots, u_n)}\ \text{with } f \in \mathcal{VF} \setminus sig(\mathsf{E})$$

$$\text{Unpairing (UR)}\ \frac{T \vdash \langle u, v\rangle}{T \vdash v} \qquad \text{Context}(\mathsf{M_E})\ \frac{T \vdash u_1\ \ldots\ T \vdash u_n}{T \vdash C[u_1, \ldots, u_n]\downarrow}\ \text{with } C \text{ an } \mathsf{E}\text{-context}$$

$$\text{Decryption (D)}\ \frac{T \vdash \{u\}_v \quad T \vdash v}{T \vdash u}$$

Fig. 1. Inference Sytem $\mathcal{I}_{\mathsf{DY+E}}$

assumption by allowing the intruder to apply function symbols such as 0, $-$, h and $+$ ($\mathsf{M_E}$). The algebraic properties of these primitives are automatically take into account thanks to the normalization.

After each deduction step, the term u obtained is reduced to its normal form $u\downarrow$. Equivalence modulo AC is easy to decide, so we omit the equality rule for AC and just work with equivalence classes modulo AC. More generally, along this paper, we consider implicitly that terms are kept in normal forms, hence we write u (resp. $u\sigma$) instead of $u\downarrow$ (resp. $u\sigma\downarrow$).

Example 3. The two inferences below are instances of the rule ($\mathsf{M_E}$) obtained by using $C[x_1, x_2] = x_1 + h(x_1) + h^2(x_1) + -2h(x_2)$ and $C[] = 0$.

$$\frac{T \vdash a + h(a) \qquad T \vdash b}{T \vdash a + h^3(a) + -2h(b)}\,(\mathsf{M_E}) \qquad\qquad \frac{}{T \vdash 0}\,(\mathsf{M_E})$$

This deductive system is equivalent in deductive power to a variant of the system in which terms are not automatically normalized, but in which arbitrary equational proofs are allowed at any moment of the deduction (see [10, 14]).

3.2 Proofs

Definition 3. *($\mathcal{I}$-proof)*
Let $\mathcal{I}$ be an inference system. An $\mathcal{I}$-proof *P of $T \vdash u$ is a tree such that:*

- *the root of P is labeled by $T \vdash u$,*
- *every leaf of P labeled by $T \vdash v$ is such that $v \in T$,*
- *for every node labeled by $T \vdash v$ having n sons labeled by $T \vdash v_1, \ldots, T \vdash v_n$, there is an instance of an inference rule of $\mathcal{I}$ with conclusion $T \vdash v$ and hypotheses $T \vdash v_1, \ldots, T \vdash v_n$ such that side conditions are satisfied.*

Definition 4. *(size, minimal)*
The size *of a proof P, denoted by $|P|$, is the number of nodes in P. A proof P of $T \vdash u$ is* minimal *if there is no proof P' of $T \vdash u$ such that $|P'| < |P|$.*

Definition 5. *(decomposition proof)*
A proof P of $T \vdash u$ is a decomposition proof *in any of the following cases:*

- *P is reduced to a leaf,*
- *P ends with an instance of a decomposition rule (*i.e. *(UL, UR, D)),*
- *P ends with an instance of ($\mathsf{M_E}$) and u is a standard term.*

Example 4. Consider the AGh theory, let $T = \{a + h(a), \langle b, c\rangle\}$, the proof P below is a proof of $T \vdash a + h^3(a) + -2h(b)$. This proof contains an instance of the rule ($\mathsf{M_E}$) with $C[x_1, x_2] = x_1 + h(x_1) + h^2(x_1) + -2h(x_2)$.

$$\dfrac{T \vdash a + h(a) \qquad \dfrac{T \vdash \langle b, c\rangle}{T \vdash b}\,(\mathsf{UL})}{T \vdash a + h^3(a) + -2h(b)}\,(\mathsf{M_E})$$

Since $a + h^3(a) + -2h(b)$ is not standard, P is not a decomposition proof. We have $|P| = 4$.

Now, we can state the following *locality* lemma. This notion was first introduced by McAllester [15] in order to characterize theories with a deduction problem decidable in PTIME. This result, proven in [10] for several equational theories and in particular for the theory AGh, allows us to focus on proof trees that involve only some particular terms.

Lemma 1. *(Locality) A minimal proof P of $T \vdash u$ contains only terms in $St_{\mathsf{E}}(T \cup \{u\})$. Moreover, if P is a decomposition proof, then P contains only terms in $St_{\mathsf{E}}(T)$.*

3.3 Deducibility Constraint System

Definition 6. *(deducibility constraint) A constraint is an expression of the form $T \Vdash u$ where T is a finite subset of $\mathcal{T}(\mathcal{F}, \mathcal{X})$, and $u \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. A system of constraints is a sequence of constraints. Given an inference system $\mathcal{I}$, a solution to a system $\mathcal{C}$ of constraints is a substitution σ such that for every $T \Vdash u \in \mathcal{C}$, there exists an $\mathcal{I}$-proof of $T\sigma \vdash u\sigma$.*

In the remainder, we are particularly interested in two inference systems: the inference system $\mathcal{I}_{\mathsf{DY+AGh}}$ described in Figure 1 and the inference system $\mathcal{I}_{\mathsf{AGh}}$ only made up of the rule $\mathsf{M}_{\mathsf{AGh}}$.

Definition 7. *(well-defined) A constraint system $\mathcal{C} = \{T_1 \Vdash u_1, \ldots, T_n \Vdash u_n\}$ is well-defined if:*

1. *(monotonicity) for all $i < n$, $T_i \subseteq T_{i+1}$,*
2. *for all substitution θ, $\mathcal{C}\theta$ satisfies the origination property:*
   $$\forall i < n,\ \forall x \in vars(T_i\theta),\ \exists j < i \text{ such that } x \in vars(u_j\theta).$$

Remember that we consider implicitly that terms are kept in normal forms, hence we write $u\theta$ instead of $u\theta\downarrow$.

Example 5. Consider the following constraint system $\mathcal{C}$ and the equational theory AGh:

$$\begin{array}{c} a,\ 2h(b)+a \Vdash h(x+y) \\ a,\ 2h(b)+a,\ -3h(x)+2b \Vdash y \end{array}$$

Although $\mathcal{C}$ satisfies the monotonicity property, $\mathcal{C}$ is not well-defined. Indeed, by applying $\theta = \{x \mapsto -y\}$ on $\mathcal{C}$, we obtain a constraint system which does not satisfy the origination property.

$$\mathcal{C}\theta = \left\{ \begin{array}{c} a,\ 2h(b)+a \Vdash 0 \\ a,\ 2h(b)+a,\ 3h(y)+2b \Vdash y. \end{array} \right.$$

This notion of well-definedness is due to Millen and Shmatikov. In [17], they show that "reasonable" protocols, in which legitimate protocol participants only execute deterministic steps (up to the generation of random nonces) always lead to a well-defined constraint system.

Theorem 1. *The problem of deciding whether a well-defined constraint system has a solution in $\mathcal{I}_{\mathsf{DY}+\mathsf{AGh}}$ is undecidable.*

The remainder of the paper is devoted to the proof of this result. In fact, the DY part of the intruder model plays no role in this undecidability result. So, in Section 4, we begins to prove the following undecidability result:

Theorem 2. *The problem of deciding whether a well-defined constraint system has a solution in $\mathcal{I}_{\mathsf{AGh}}$ is undecidable.*

This undecidability result is obtained by encoding the Hilbert's 10^{th} problem into a constraint system in which all the terms are built over the restricted signature made up of $\{0, -, h, +, \}$ and a set of constants. In particular, we do not use Dolev-Yao symbol (pairing, encryption,...) in our encoding.

Note that in the inference system $\mathcal{I}_{\mathsf{AGh}}$, all the proofs are reduced either to a leaf or to the application of one instance of the rule $\mathsf{M}_{\mathsf{AGh}}$. This is due to the fact that we can always put together two instances of the rule $\mathsf{M}_{\mathsf{AGh}}$. In such an inference system, deciding whether a system of constraint has a solution can be expressed as a system of quadratic equations of a particular form over $\mathbb{Z}[h]$, the ring of polynomials in one indeterminate over the field $\mathbb{Z}$. This have already been remark in [17] in the case of the theory AG (for which system of equations are over $\mathbb{Z}$) and in [11] for the theory ACUNh, *i.e.* AGh plus the equation $-x = x$ (system of equations are over $\mathbb{Z}/2\mathbb{Z}[h]$). The system, we have to solve, have a particular form due to the well-definedness of the constraint system. Such systems have been shown decidable in the case of AG [17] and ACUNh [11]. Unfortunately, we show that a similar result does not hold for systems over $\mathbb{Z}[h]$.

In Section 5, we prove Theorem 1, by showing that the same encoding works also in the case of $\mathcal{I}_{\mathsf{DY}+\mathsf{AGh}}$. We proceed in two steps. Firstly, we show that if

a constraint system has a solution in $\mathcal{I}_{\mathsf{DY+AGh}}$, then there is one which does not introduce any new structure. Thanks to this, we can easily state that the constraint system obtained with our encoding has a solution in $\mathcal{I}_{\mathsf{AGh}}$ if and only if there is one in $\mathcal{I}_{\mathsf{AGh}}$. This will allow us to conclude.

4 Undecidability for $\mathcal{I}_{\mathsf{AGh}}$

Given an instance S of Hilbert's 10^{th} problem with n variables, we built a well-defined constraint system $\mathcal{C}(S)$, such that S has a solution $(v_1, \ldots, v_n)$ over $\mathbb{Z}$ if and only if $\mathcal{C}(S)$ has a solution in $\mathcal{I}_{\mathsf{AGh}}$. We use the following formulation of Hilbert's 10^{th} problem, known to be undecidable [9].

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 $x_i^2 = x_j$.
OUTPUT: Does S have a solution over $\mathbb{Z}$?

We choose to encode an integer v in a ground term t by $\mathcal{N}(a, t)$ (see Definition 2). Our encoding is made up of two parts. The first one (Section 4.1) is independent of the equations of S. This part is used to introduce our term variables and to ensure some relationships between them after their instantiation by σ, a solution of $\mathcal{C}(S)$ (see Lemma 2). In the second part of our encoding (Section 4.2), we deal with the equations of S: each one is encoded by a constraint.

4.1 Encoding Product

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

$$h^{p+n+2}(a) \Vdash h^{p+n+2}(X_i) \qquad (\tau_1)$$
$$h^{p+n+2}(a) \Vdash h^{p+n+2}(Y_i) \qquad (\tau_1)$$
$$h^{p+n+1}(b),\ h^{p+n+2}(a) \Vdash h^{p+n+1}(X_i') \qquad (\tau_1)$$
$$h^{p+n}(a+b),\ h^{p+n+1}(b),\ h^{p+n+2}(a) \Vdash h^{p+n}(X_i + X_i') \qquad (\tau_2)$$
$$h^{p+n-1}(X_1 + b),\ h^{p+n-2}(X_2 + b),\ \ldots,\ h^{p+n-i}(X_i + b),$$
$$h^{p+n}(a+b),\ h^{p+n+1}(b),\ h^{p+n+2}(a) \Vdash h^{p+n-i}(Y_i + X_i') \qquad (\tau_3)$$

Let $\mathcal{A}_1(n)$ (resp. $\mathcal{A}_2(n)$, $\mathcal{A}_3(n)$) be the constraint system which is made up of the constraints of type τ_1 (resp. τ_2, τ_3).

Example 6. We illustrate the first part of our construction with $n = 3$. We gather together constraints of the same type.

$$\mathcal{A}_1(3) := \begin{cases} h^8(a) \Vdash h^8(X_1) & h^8(a) \Vdash h^8(Y_1) & h^7(b),\ h^8(a) \Vdash h^7(X'_1) \\ h^8(a) \Vdash h^8(X_2) & h^8(a) \Vdash h^8(Y_2) & h^7(b),\ h^8(a) \Vdash h^7(X'_2) \\ h^8(a) \Vdash h^8(X_3) & h^8(a) \Vdash h^8(Y_3) & h^7(b),\ h^8(a) \Vdash h^7(X'_3) \end{cases}$$

$$\mathcal{A}_2(3) := \begin{cases} h^6(a+b),\ h^7(b),\ h^8(a) \Vdash h^6(X_1 + X'_1) \\ h^6(a+b),\ h^7(b),\ h^8(a) \Vdash h^6(X_2 + X'_2) \\ h^6(a+b),\ h^7(b),\ h^8(a) \Vdash h^6(X_3 + X'_3) \end{cases}$$

$$\mathcal{A}_3(3) := \begin{cases} h^5(X_1+b),\ h^6(a+b),\ h^7(b),\ h^8(a) \Vdash h^5(Y_1 + X'_1) \\ h^4(X_2+b),\ h^5(X_1+b),\ h^6(a+b),\ h^7(b),\ h^8(a) \Vdash h^4(Y_2 + X'_2) \\ h^3(X_3+b),\ h^4(x_2+b),\ h^5(X_1+b),\ h^6(a+b),\ h^7(b),\ h^8(a) \Vdash h^4(Y_3 + X'_3) \end{cases}$$

Lemma 2. *Let $n \in \mathbb{N}$ and σ a solution to $\mathcal{A}(n)$ in $\mathcal{I}_{\mathsf{AGh}}$. We have:*

1. *For $1 \leq i \leq n$, $\mathcal{N}(a, X_i\sigma) = \mathcal{N}(b, X'_i\sigma)$,*
2. *For $1 \leq i \leq n$, $\mathcal{N}(a, Y_i\sigma) = \mathcal{N}(a, X_i\sigma)^2$.*

Proof. Let σ be a solution to $\mathcal{A}(n)$. Firstly, constraints of type τ_1 ensure that $\mathcal{N}(b, X_i\sigma) = \mathcal{N}(b, Y_i\sigma) = 0$ and $\mathcal{N}(a, X'_i\sigma) = 0$. Thanks to the constraints of type τ_2, we have that $\mathcal{N}(a, X_i\sigma) + \mathcal{N}(a, X'_i\sigma) = \mathcal{N}(b, X_i\sigma) + \mathcal{N}(b, X'_i\sigma)$. Putting these two results together allow us to conclude for *(1)*. Now, we consider the i^{th} constraint of type τ_3, *i.e.*:

$$h^{p+n-1}(X_1+b),\ h^{p+n-2}(X_2+b),\ \ldots,\ h^{p+n-i}(X_i+b),\ h^{p+n}(a+b),\ h^{p+n+1}(b),\ h^{p+n+2}(a) \Vdash h^{p+n-i}(Y_i + X'_i).$$

This constraint ensures that there exists $z \in \mathbb{Z}$ such that:

- $z \times (1 + \mathcal{N}(b, X_i\sigma)) = \mathcal{N}(b, X'_i\sigma) + \mathcal{N}(b, Y_i\sigma)$, and
- $z \times \mathcal{N}(a, X_i\sigma) = \mathcal{N}(a, X'_i\sigma) + \mathcal{N}(a, Y_i\sigma)$.

Thanks to the fact that $\mathcal{N}(b, X_i\sigma) = \mathcal{N}(b, Y_i\sigma) = \mathcal{N}(a, X'_i\sigma) = 0$ and *(1)*, we conclude for *(2)*. □

4.2 Encoding Equations of S

In this section, we described the part $\mathcal{B}(S)$ of our encoding which really depends on $S = \{e_1, \ldots, e_p\}$. $\mathcal{B}(S)$ contains one constraint per equation, denoted by $\mathsf{d}_1, \ldots, \mathsf{d}_p$. We let $T_0 = \{h^{p+n-j}(X_j + b) \mid 1 \leq j \leq n\},\ h^{p+n}(a+b),\ h^{p+n+1}(b),\ h^{p+n+2}(a)$ (*i.e.* the set of hypotheses obtained at the end of the first part of our encoding) and build the d_k's inductively, depending on the form of e_k. The c_k's are constants distinct from 0, a and b.

- if $e_k =$ "$x_i = m$" then $T_k = T_{k-1},\ h^{p-k}(X_i) + c_k$ and $\mathsf{d}_k = T_k \Vdash h^{p-k}(ma) + c_k$,
- if $e_k =$ "$x_i + x_{i'} = x_j$" then $T_k = T_{k-1},\ h^{p-k}(X_i + X_{i'}) + c_k$ and $\mathsf{d}_k = T_k \Vdash h^{p-k}(X_j) + c_k$,

– if $e_k =$ "$x_i = x_j^2$ " then $T_k = T_{k-1}, h^{p-k}(X_i) + c_k$
and $\mathsf{d}_k = T_k \Vdash h^{p-k}(X'_j) + c_k$.

Example 7. Let $S_e = \{x_1 = 2,\ \ x_2^2 = x_3,\ \ 3x_2 + x_3 = x_1\}$. We obtain:

$$\mathcal{B}(S_e) := \begin{cases} & h^2(X_1) + c_1,\ T_0 \Vdash h^2(2a) + c_1 \\ & h(X_3) + c_2,\ h^2(X_1) + c_1,\ T_0 \Vdash h(Y_2) + c_2 \\ 3X_2 + X_3 + c_3,\ & h(X_3) + c_2,\ h^2(X_1) + c_1,\ T_0 \Vdash h(X_1) + c_3 \end{cases}$$

Proposition 1. *Let S be a set of equations (over n variables) and $\mathcal{C}(S)$ be the constraint system $\mathcal{A}(n) \cup \mathcal{B}(S)$. We have:*

1. *$\mathcal{C}(S)$ is well-defined,*
2. *S has a solution over $\mathbb{Z} \Leftrightarrow \mathcal{C}(S)$ has a solution in $\mathcal{I}_{\mathsf{AGh}}$.*

Proof. 1. The fact that variables have been introduced at the beginning and one by one ensures the well-definedness of the constraint system.

2. ($\Rightarrow$) Let $v_1, \ldots, v_n$ be a solution to S. Let $\sigma = \{X_1 \mapsto v_1 a, \ldots, X_n \mapsto v_n a, X'_1 \mapsto v_1 b, \ldots, X'_n \mapsto v_n b, Y_1 \mapsto v_1^2 a, \ldots, Y_n \mapsto v_n^2 a\}$. We prove that σ is a solution to $\mathcal{C}(S)$. To do this, we have to show that for each constraint $T \Vdash u \in \mathcal{C}(S)$, there exists an $\mathcal{I}_{\mathsf{AGh}}$-proof of $T\sigma \vdash u\sigma$. It is easy to show that such proofs exist. Each time we only have to use the last term introduced in the hypothesis set of the given constraint.
($\Leftarrow$) Let σ be a solution to $\mathcal{C}(S)$. Let $v_i = \mathcal{N}(a, X_i\sigma)$. We show that $v_1, \ldots, v_n$ is a solution to S. From Lemma 2, we have $\mathcal{N}(a, Y_i\sigma) = \mathcal{N}(a, X_i\sigma)^2$. We have to show that $(v_1, \ldots, v_n)$ is a solution to each equation in S. Let e_k be the k^{th} equation of S. Consider the constraint in $\mathcal{B}(S)$ corresponding to this equation. For instance, assume that the equation is of the form "$x_i = x_j^2$" (the others cases are similar). Then the constraint is of the form:
$$T_{k-1}\sigma,\ h^{p-k}(X_i\sigma) + c_k \Vdash h^{p-k}(Y_j\sigma) + c_k$$
Note that (i) c_k only appears in the term $h^{p-k}(X_i\sigma) + c_k$ among all the terms in the hypotheses and (ii) c_k has to appear in the conclusion. We deduce that $\mathcal{N}(a, X_i\sigma) = \mathcal{N}(a, Y_j\sigma)$. By Lemma 2, we know that $\mathcal{N}(a, Y_j\sigma) = \mathcal{N}(a, X_j\sigma)^2$. This allows us to conclude. □

5 Undecidability for $\mathcal{I}_{\mathsf{DY+AGh}}$

In this section, we are going to prove Theorem 1. To do this, we proceed in two steps. Firstly, we show the existence of a *conservative solution* (Lemma 3) which does not introduce any new structure. Since a conservative solution does not introduce new structure, we know that if there exists a solution of the constraint system $\mathcal{C}$ built in Section 4 then there exists one such that $\mathcal{C}\sigma$ only contains terms built over the restricted signature, *i.e.* 0, h, $-$, $+$ and some constants. Thanks to the locality lemma stated in Section 3, we easily deduce that such constraints only involved the rule $\mathsf{M}_{\mathsf{AGh}}$.

Definition 8. *(conservative solution)*
Let $\mathcal{C}$ be a constraint system and σ a solution to $\mathcal{C}$. σ is a conservative solution *to $\mathcal{C}$ if for all $x \in vars(\mathcal{C})$, $Fact_{\mathsf{E}}(x\sigma) \subseteq (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C}))\sigma$.*

Lemma 3. *Let $\mathcal{C}$ be a well-defined constraint system. If there exists a solution σ to $\mathcal{C}$ then there exists a conservative one.*

Before to prove this, we need to introduce a definition and an additional proposition.

Definition 9. *(decomposed) Let P be a proof of $T \vdash u$. We say that a standard term v is decomposed in P if:*

- *either $v = \langle u_1, u_2 \rangle$ and P contains an instance of (UL) or (UR) whose premise is labeled with $T \vdash \langle u_1, u_2 \rangle$.*
- *or $v = \{u_1\}_{u_2}$ and P contains an instance of (D) whose premises are labeled with $T \vdash \{u_1\}_{u_2}$ and $T \vdash u_2$.*

The following proposition has been proved in [19] for the standard Dolev-Yao model. The proof of [19] can be transferred in a straightforward way to our intruder model which comprises in addition to the standard rules the rule $(\mathsf{M_E})$. It will be used in Lemma 3 to ensure the existence of a proof of $T \vdash u$ which respects some conditions.

Proposition 2. *Let P be a proof of $T \vdash u$ and P' be a minimal proof of $T \vdash \gamma$. Moreover, assume that P' ends with an instance of (C). Then, there exists a proof of $T \vdash u$ in which γ is never decomposed.*

Proof. The proof can be done by induction on the number of instances of inference rules in P which decompose γ.
Base case: If there is no such an instance, then P is the expected proof.
Induction case: Assume there are $n+1$ instances of inference rules in P which decompose γ. We can distinguish two cases depending on the fact that γ is a pair (*i.e.* $\langle \gamma_1, \gamma_2 \rangle$) or a ciphertext (*i.e.* $\{\gamma_1\}_{\gamma_2}$). In the first case, this means that there exists an instance of (UL) (or (UR)) whose premise is $\langle \gamma_1, \gamma_2 \rangle$ and conclusion is γ_1 (or γ_2). From P', we can easily extract a proof P_1 of $T \vdash \gamma_1$ (resp. P_2 of $T \vdash \gamma_2$). Note that P_1 (resp. P_2) does not decompose γ by minimality of P'. Hence, such a proof can be plugged to replace the subproof of $T \vdash \gamma_1$ (resp. $T \vdash \gamma_2$) in P which decompose γ. The second case where $\gamma = \{\gamma_1\}_{\gamma_2}$ is similar. We obtain a proof of $T \vdash u$ which contains less instances of inference rules which decompose γ than P. □

Now, we are able to prove Lemma 3. Remember that we consider implicitly that terms are kept in normal forms, hence we write $u\sigma$ instead of $u\sigma\downarrow$.

Proof. (of Lemma 3)
We assume given a linear well-founded ordering $\prec$ on standard terms of $\mathcal{T}(\mathcal{F}, \mathcal{X})$ such that the constant 0 is minimal w.r.t. $\prec$. We shall use below the

multi-set extension $\ll$ of $\prec$ to multi-sets of standard ground terms. For sake of notation, given two solutions σ_1 and σ_2 of a constraint system, we write $\sigma_1 \ll \sigma_2$ if and only if $Fact_{\mathsf{E}}(img(\sigma_1)) \ll Fact_{\mathsf{E}}(img(\sigma_2))$. Let σ be a minimal (w.r.t. $\ll$) solution to $\mathcal{C}$.

We reason by contradiction to show that σ is conservative w.r.t. $\mathcal{C}$. Assume that there exists $x \in vars(\mathcal{C})$ and $v_x \in Fact_{\mathsf{E}}(x\sigma)$ such $v_x \notin (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(C))\sigma$ *i.e.* for all $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \setminus \mathcal{X}$ with $t\sigma =_E v_x$, we have $t \notin St_{\mathsf{E}}(\mathcal{C})$. We will show that under this condition there exists a smaller solution σ' of $\mathcal{C}$. Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ and for each $i \leq k$, let $T_i \Vdash u_i$ be the constraint C_i and $C_i\sigma$ be the constraint obtained from C_i by instantiating (and normalizing) all the terms with σ.

Fact 1 *If $v_x \in St_{\mathsf{E}}(s\sigma)$ for some $s \in T_i$ $(i \leq k)$, then there exists $j < i$ such that $v_x \in St_{\mathsf{E}}(u_j\sigma)$.*

We show this result by contradiction. Assume that $v_x \in St_{\mathsf{E}}(s\sigma)$ for some $s \in T_i$ $(i \leq k)$, and that for all $j < i$, we have $v_x \notin St_{\mathsf{E}}(u_j\sigma)$. Let z be a fresh variable, and ρ be the replacement $\{v_x \mapsto z\}$. Let $\theta := \sigma\rho$. We are going to show that $\mathcal{C}\theta$ is not well-formed, leading to a contradiction with the fact that $\mathcal{C}$ is well-defined. Firstly, since $v_x \notin (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(C))\sigma$, we have $(\mathcal{C}\sigma)\rho = \mathcal{C}(\sigma\rho)$ $(= \mathcal{C}\theta)$. By hypothesis, $v_x \in St_{\mathsf{E}}(T_i\sigma)$, hence $z \in vars(T_i\theta)$. However, for all $j < i$, we have $z \notin vars(u_j\theta)$ since $v_x \notin St_{\mathsf{E}}(u_j\sigma)$.

This allows us to define: $m = \min\{j \mid v_x \in St_{\mathsf{E}}(u_j\sigma)\}$.

Fact 2 *There exists P' a proof of $T_m\sigma \vdash v_x$ which ends with an instance of* (C).

By hypothesis, there exists a minimal proof P of $T_m\sigma \vdash u_m\sigma$. Firstly, we show that there exists in P a node labeled with $T_m\sigma \vdash v_x$. If P contains a node labeled by $T_m\sigma \vdash v_x$, then it is the expected node. Otherwise, we can find recursively a path in P, from the root up to one leaf, where every node which is labeled by $T_m\sigma \vdash u$ is such that $v_x \in St_{\mathsf{E}}(u)$. Thanks to Fact 1, the existence of such a path leads to a contradiction with the minimality of m. Secondly, by definition of m and thanks to Lemma 1 (locality lemma), the subproof P' of P labeled with $T_m\sigma \vdash v_x$ can not be a decomposition proof (otherwise $v_x \in St_{\mathsf{E}}(T_m\sigma)$). Since v_x is necessarily a standard term, this implies that P' ends with an instance of (C).

Now, we let δ be the replacement $\{v_x \mapsto 0\}$. We will show that $\sigma' := \sigma\delta$ is also a solution of $\mathcal{C}$, which is a contradiction since $\sigma' \ll \sigma$ (v_x is a standard term since it is a factor, hence $0 \prec v_x$). For this purpose, we have to build a proof of each $C_i\sigma'$, $i \leq l$. We distinguish two cases.

1. Case $i < m$: By definition of m, $v_x \notin St_{\mathsf{E}}(C_i\sigma)$. In this case, $(C_i\sigma)\delta = C_i\sigma = C_i\sigma'$, *i.e.* σ' is a solution to C_i.
2. Case $i \geq m$: In the remainder, we are going to show that $\sigma' = \sigma\delta$ is also a solution to $C_i = T_i \Vdash u_i$.

Firstly, we may note that $C_i(\sigma\delta) = (C_i\sigma)\delta$ since by hypothesis $v_x \notin (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(C))\sigma$. By hypothesis σ is a solution to C_i, this means that we have a proof P of $T_i\sigma \vdash u_i\sigma$. Moreover, Fact 2 ensures the existence of a proof of $T_i\sigma \vdash v_x$ which ends with (C) in P. σ' is a solution of $\mathcal{C}_i$, it is obvious for $i = m$ and we extend the result for $i > m$ by well-definedness of $\mathcal{C}$ (stability by any substitution that $\mathcal{C}$ is well-formed). Now, we can apply Proposition 2 to obtain a proof P_i of $T_i\sigma \vdash u_i\sigma$ in which v_x is never decomposed. We shall build from P_i a proof P_i' of $(T_i\sigma)\delta \vdash (u_i\sigma)\delta$ by replacing every subtree ended by $\dfrac{T_i\sigma \vdash v_1 \ldots T_i\sigma \vdash v_n}{T_i\sigma \vdash v_x}\,(\mathsf{C})$ with a leaf labeled with $T_i\sigma \vdash v_x$ and then by applying δ to every term of the tree obtained.

Fact 3 *P_i' is a proof of $(T_i\sigma)\delta \vdash (u_i\sigma)\delta$.*

To prove this, we have to show that for every node in P_i' labeled with $T_i\sigma\delta \vdash v_0$ and with n sons labeled respectively by $T_i\sigma\delta \vdash v_1, \ldots T_i\sigma\delta \vdash v_n$, the inference $\dfrac{T_i\sigma\delta \vdash v_1 \ldots T_i\sigma\delta \vdash v_n}{T_i\sigma\delta \vdash v_0}$ is an instance of an inference rule of Figure 1.

We distinguish several cases:

- If the inference is a leaf added by the replacement of an instance of (C) in the construction of P_i' given above, then we have $v_0 = 0$, hence $v_0 \in T_i\sigma\delta$.
- If the inference is not a leaf added by the replacement, then we have a "corresponding" inference in P_i. This means that there exists $\dfrac{T_i\sigma \vdash u_1 \ldots T_i\sigma \vdash u_n}{T_i\sigma \vdash u_0}$ an inference step in P_i such that $v_i = u_i\delta$ for each $0 \leq i \leq n$. Since, by construction of P_i' we know that v_x is never decomposed in P_i and the conclusion of an instance of (C) can not be v_x, we can show by case analysis on the inference rule, that when we apply δ on the inference above, we retrieve another instance of the same inference rule. □

Example 8. Consider the following well-defined constraint system $\mathcal{C}$:

$$\begin{array}{ll} a, h(b) & \Vdash h(x) \\ a, h(b), x & \Vdash \langle a, b\rangle \end{array}$$

One solution is $\sigma = \{x \mapsto \langle a, a\rangle + b\}$. This solution is not conservative w.r.t. $\mathcal{C}$. Indeed $Fact_{\mathsf{E}}(\langle a, a\rangle + b) = \{\langle a, a\rangle, b\}$, and $\langle a, a\rangle$ does not belong to $(St_{\mathsf{E}}(\mathcal{C}) \setminus \{x\})\sigma$ which is equal to $\{0, h(b), b, h(\langle a, a\rangle + b), \langle a, b\rangle, a\}$. However, as it is said in Lemma 3, there is a conservative solution: $\{x \mapsto b\}$.

Now, we are able to prove Theorem 1 stated in Section 3.

Theorem 1. *The problem of deciding whether a well-defined constraint system has a solution in $\mathcal{I}_{\mathsf{DY+AGh}}$ is undecidable.*

Proof. Let S be an instance of the Hilbert's 10^{th} problem and $\mathcal{C}(S)$ be the well-defined constraint system obtained by applying the procedure described in

Section 4. Firstly, it is obvious that if $\mathcal{C}(S)$ has a solution in $\mathcal{I}_{\mathsf{AGh}}$ then $\mathcal{C}(S)$ has also one in $\mathcal{I}_{\mathsf{DY+AGh}}$. Conversly, let σ be a solution of $\mathcal{C}(S)$ in $\mathcal{I}_{\mathsf{DY+AGh}}$. By Lemma 3, we can assume w.l.o.g. that σ is conservative. Hence all the terms which appears in $\mathcal{C}(S)\sigma$ are built over the signature 0, h, $-$, $+$ and some constants. Thanks to the locality lemma (Lemma 1), we know that a minimal proof of every constraint $T \vdash u \in \mathcal{C}(S)\sigma$ only involves terms in $St_{\mathsf{E}}(T \cup \{u\})$. This set only contains terms of T, the term u and some constants. In other words, inference rules such as (UL), (UR), (D) or (C) can not be used. This allows us to conclude that σ is a solution of $\mathcal{C}(S)$ in $\mathcal{I}_{\mathsf{AGh}}$. Hence we have:

$$\mathcal{C}(S) \text{ has a soltuion in } \mathcal{I}_{\mathsf{DY+AGh}} \Leftrightarrow \mathcal{C}(S) \text{ has a soltuion in } \mathcal{I}_{\mathsf{AGh}}.$$

This result together with Proposition 1 allows us to conclude. □

6 Conclusion

In this paper, satisfaisability of well-defined constraint systems is shown undecidable for the theory AGh. This result completes the view of the problem for the three theories ACh (for which unification is undecidable [18]), ACUNh (AGh plus the equation $-(x) = x$) and AGh. The undecidablity result for AGh contrasts with the decidability one obtained for ACUNh [11]. It would now be interesting to have a complete view of the problem for the three theories AC, ACUN and AG. Although results for ACUN and AG are known to be decidable [6, 3, 17], the AC case seems to be very challenging.

References

1. R. Amadio and W. Charatonik. On name generation and set-based analysis in the Dolev-Yao model. In *Proc. International Conference on Concurrency Theory (CONCUR'02)*, volume 2421 of *LNCS*, pages 499–514, Brno (Czech Republic), 2002. Springer-Verlag.
2. F. Baader. Unification in commutative theories, Hilbert's basis theorem, and Gröbner bases. *Journal of the ACM*, 40(3):477–503, 1993.
3. Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with XOR. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 261–270, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.
4. Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani, and L. Vigneron. Deciding the security of protocols with Diffie-Hellman exponentiation and product in exponents. In *Proc. 23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'03)*, volume 2914 of *LNCS*, pages 124–135, Mumbai (India), 2003. Springer-Verlag.
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 V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 271–280, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.

7. V. Cortier. Vérifier les protocoles cryptographiques. *Technique et Science Informatiques*, 24(1):115–140, 2005.
8. V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1):1–43, 2006.
9. 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.
10. S. Delaune. Easy intruder deduction problems with homomorphisms. *Information Processing Letters*, 97(6):213–218, Mar. 2006.
11. S. Delaune, P. Lafourcade, D. Lugiez, and R. Treinen. Symbolic protocol analysis in presence of a homomorphism operator and *exclusive or*. Research Report LSV-05-20, LSV, ENS Cachan, France, 2005.
12. D. Dolev, S. Even, and R. M. Karp. On the security of ping-pong protocols. In *Proc. Advances in Cryptology (CRYPTO'82)*, pages 177–186, Santa Barbara (California, USA), 1983.
13. N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Proc. Workshop on Formal Methods and Security Protocols (FMSP'99)*, Trento (Italy), 1999.
14. P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for AC-like equational theories with homomorphisms. In *Proc. 16th International Conference on Rewriting Techniques and Applications (RTA'05)*, volume 3467 of *LNCS*, pages 308–322, Nara (Japan), 2005. Springer.
15. D. A. McAllester. Automatic recognition of tractability in inference relations. *Journal of the ACM*, 40(2):284–303, 1993.
16. 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)*. ACM Press, 2001.
17. J. Millen and V. Shmatikov. Symbolic protocol analysis with an abelian group operator or Diffie-Hellman exponentiation. *Journal of Computer Security*, 13(3):515–564, 2005.
18. P. Narendran. Solving linear equations over polynomial semirings. In *Proc. 11th Annual IEEE Symposium on Logic in Computer Science (LICS'96)*, pages 466–472, New Brunswick, New Jersey, 1996. IEEE Comp. Soc. Press.
19. M. Rusinowitch and M. Turuani. Protocol insecurity with a finite number of sessions, composed keys is NP-complete. *Theoretical Computer Science*, 1-3(299):451–475, 2003.