An Undecidability Result for AGh

Stéphanie Delaune

France Télécom R&D,
Lab. Spécification & Vérification, CNRS & ENS de Cachan, France.

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 satisfiability of symbolic deductibility constraints) is undecidable, obtaining in this way the first undecidability result concerning a theory for which unification is known to be decidable [2].

Key words: formal methods, security protocols, constraint solving

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 of the absence of logical attacks on such protocols (*e.g.* [6]).

The problem of deciding whether a protocol is secure or not is known to be undecidable in general, even under several restrictions [1,4,12]. An interesting decidability result has been obtained by Rusinowitch and Turuani [15], under the assumption that the number of sessions (*i.e.* the number of parallel role instances) is bounded. This pioneer work 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.

Email address: `delaune@lsv.ens-cachan.fr` (Stéphanie Delaune).

Since 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 (*e.g.* [6]). It is well-known that the equational theories we can hope to handle are those for which unification is decidable. 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, *i.e.* the *Abelian groups* theory (AG) in combination with the axiom (h): $\mathsf{h}(x+y) = \mathsf{h}(x)+\mathsf{h}(y)$, whose unification problem is known to be decidable [2]. We prove that the security problem for a bounded number of sessions is undecidable. The question of the decidability of the protocol security problem in that case is interesting since some protocols relies on these algebraic properties. A well-known example is the TMN protocol [16] on which an attack, due to Simmons, makes use of the homomorphic property of RSA encryption: $\{x \times y\}_{\mathsf{pub}(S)} = \{x\}_{\mathsf{pub}(S)} \times \{y\}_{\mathsf{pub}(S)}$. Such a protocol, in which RSA encryption is only used with the public key of the server, can be modeled in our settings assuming that the decryption key of the server is a trusted key.

2 Preliminaries

2.1 Basic Definitions

We use classical notation and terminology on terms (see [10] for details). 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 $\mathcal{X}$ of variables. $\mathcal{T}(\mathcal{F}, \emptyset)$ 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, $\mathsf{h}(.)$, $-.$ and $. + .$ related to the equational theory studied in this paper. The set of variables occurring in t is noted $vars(t)$.

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. If E is a set of equations (unordered pair of terms), we note $sig(\mathsf{E})$ for the set of function symbols occurring in E and 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. An E-*context* is a λ-term $\lambda x_1, \dots, x_n.t$ with $t \in \mathcal{T}(sig(\mathsf{E}), \{x_1, \dots, x_n\})$, also written $t[x_1, \dots, x_n]$. The application of $t[x_1, \dots, x_n]$ to arguments $u_1, \dots, u_n$ is written $t[u_1, \dots, u_n]$.

A *term rewriting system* is a finite set of *rewrite rules* $l \rightarrow r$ where $l \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $r \in \mathcal{T}(\mathcal{F}, vars(l))$. Given a term rewriting system $\mathcal{R}$ and a set of equations E, the relation $\rightarrow_{\mathcal{R}/\mathsf{E}}$ (*rewriting modulo* E) is defined as follows: $s \rightarrow_{\mathcal{R}/\mathsf{E}} t$ if and only if $s =_{\mathsf{E}} u[l\sigma]_p$ and $u[r\sigma]_p =_{\mathsf{E}} t$, for some context u, position p in u, rule $l \rightarrow r \in \mathcal{R}$, and substitution σ. We denote by $\xrightarrow{*}_{\mathcal{R}/\mathsf{E}}$ the reflexive and transitive closure of $\rightarrow_{\mathcal{R}/\mathsf{E}}$. A rewrite system $\mathcal{R}/\mathsf{E}$ is said to be E-*convergent* if there is no infinite chains $t_1 \rightarrow_{\mathcal{R}/\mathsf{E}} t_2 \rightarrow_{\mathcal{R}/\mathsf{E}} \ldots$ and for every three terms t, s_1 and s_2 such that $t \rightarrow_{\mathcal{R}/\mathsf{E}} s_1$ and $t \rightarrow_{\mathcal{R}/\mathsf{E}} s_2$, there exists a term s such that $s_1 \xrightarrow{*}_{\mathcal{R}/\mathsf{E}} s$ and $s_2 \xrightarrow{*}_{\mathcal{R}/\mathsf{E}} s$. A term t is in *normal form* (w.r.t. $\rightarrow_{\mathcal{R}/\mathsf{E}}$) if there is no term s such that $t \rightarrow_{\mathcal{R}/\mathsf{E}} s$. If $t \xrightarrow{*}_{\mathcal{R}/\mathsf{E}} s$ and s is in normal form then we say that s is a normal form of t.

2.2 Dolev-Yao Model Extended with an Equational Theory

The most widely used deduction relation representing the deduction abilities of an intruder is often referred to as the Dolev-Yao model [11]. In addition, we give to the intruder the power to use equational reasoning modulo a set E of equational axioms (see Figure 1).

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

$$\text{Unpairing (UR)}\ \frac{T \vdash \langle u, v\rangle}{T \vdash v} \qquad \text{Decryption (D)}\ \frac{T \vdash \{u\}_v \quad T \vdash v}{T \vdash u}$$

$$\text{Equality (Eq)}\ \frac{T \vdash u}{T \vdash v}\ \text{ if } u =_{\mathsf{E}} v$$

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

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 assumption* through the rule (Eq) allowing the intruder to exploit the algebraic properties of cryptographic primitives.

Definition 1 ($\mathcal{I}$-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.*

2.3 Equational Theory AGh

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

- Associativity & Commutativity (AC): $x+(y+z) = (x+y)+z$, $x+y = y+x$,
- Unit (U) & Inverse (Inv): $x+0 = x$, $x + -(x) = 0$.

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:

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

In the remainder of the paper, we always assume that terms are kept in normal form (w.r.t. $\to$).

Example 2 *Let $T = \{a + \mathsf{h}(a), -\mathsf{h}(\mathsf{h}(a)) + b\}$. The proof tree below is an $\mathcal{I}_{\mathsf{DY+AGh}}$-proof of $T \vdash h(a) + b$.*

$$
\dfrac{\dfrac{\dfrac{T \vdash a + \mathsf{h}(a)}{T \vdash \mathsf{h}(a+\mathsf{h}(a))}\,(\mathsf{C}) \qquad T \vdash -\mathsf{h}(\mathsf{h}(a)) + b}{T \vdash \mathsf{h}(a+\mathsf{h}(a)) + -\mathsf{h}(\mathsf{h}(a)) + b}\,(\mathsf{C})}{T \vdash \mathsf{h}(a) + b}\,(\mathsf{Eq})
$$

A term t is *standard* if it is not of the form $f(t_1, \ldots, t_n)$ with $f \in sig(\mathsf{E})$.

Definition 3 (factors) *Let t be a term (in normal form), $t =_{\mathsf{AC}} C[t_1, \ldots, t_n]$ for some standard terms $t_1, \ldots, t_n$ (in normal forms) and an E-context C. The set $Fact_{\mathsf{E}}(t)$ of* factors *of t is defined by $Fact_{\mathsf{E}}(t) = \{t_1, \ldots, t_n\}$.*

Note that, since the term rewriting system considered is AC-convergent, the set $Fact_{\mathsf{E}}(t)$ is uniquely determined up to associativity and commutativity of $+$. For example, let $t_1 = \mathsf{h}^2(a) + b + c$ and $t_2 = \mathsf{h}(\langle a, b\rangle) + c$. The terms t_1 and t_2 are not standard. We have $Fact_{\mathsf{E}}(t_1) = \{a, b, c\}$ and $Fact_{\mathsf{E}}(t_2) = \{\langle a, b\rangle, c\}$.

Let $n \in \mathbb{N}$. The notation $\mathsf{h}^n(t)$ (resp. nt) represents the term t (resp. 0) if $n = 0$, and $\mathsf{h}(\mathsf{h}^{n-1}(t))$ (resp. $t + (n-1)t$) otherwise. Lastly, $-nt$ represents the term $n(-t)$. A polynomial $P(\mathsf{h}) \in \mathbb{Z}[\mathsf{h}]$ can be written $\sum_{i=0}^{n} c_i \mathsf{h}^i$ where $c_i \in \mathbb{Z}$. The product $\odot$ of a polynomial by a term is a term defined as follows:

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

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

Definition 4 (number of occurrences) *Let t be a ground term and f a ground standard term. The number of occurrences of f in t, denoted $\mathcal{N}(f, t)$, is 0 if $f \notin \mathit{Fact}_{\mathsf{E}}(t)$ and $p_f(0)$ otherwise.*

Note that an occurrence of a constant c under a symbol h in a term t plays no role in $\mathcal{N}(c, t)$. For instance, let $p = (3\mathsf{h}^2 + -2)$ and $t = a + 2b$. We have: $p \odot t = 3\mathsf{h}^2(a + 2b) + -2(a + 2b) = (3\mathsf{h}^2 + -2) \odot a + (6\mathsf{h}^2 + -4) \odot b$. Hence $\mathcal{N}(a, p \odot t) = -2$ and $\mathcal{N}(b, p \odot t) = -4$.

3 Constraints Solving

3.1 Security via Constraint Solving

In our 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 *deductibility constraints*.

More explanations about how to construct the symbolic constraint system from a given protocol can be found in [5,13].

3.2 Deductibility Constraint System

Definition 5 (deductibility constraint) *A deductibility 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 constraint system $\mathcal{C}$ is a substitution σ such that for every $T \Vdash u \in \mathcal{C}$, there exists an $\mathcal{I}$-proof of $T\sigma \vdash u\sigma$.

Definition 6 (well-defined) *A constraint system $\mathcal{C} = \{T_1 \Vdash u_1, \ldots, T_n \Vdash u_n\}$ is well-defined if:*

(1) for all $i < n$, $T_i \subseteq T_{i+1}$,
(2) for all substitution θ, $\mathcal{C}\theta$ satisfies the following requirement:

$$\forall i \leq n,\ \forall x \in vars(T_i\theta),\ \exists j < i \text{ such that } x \in vars(u_j\theta).$$

This notion of well-definedness is due to Millen and Shmatikov. In [13], 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 7 *The problem of deciding whether a well-defined constraint system has a solution in $\mathcal{I}_{\mathsf{DY+AGh}}$ 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. More precisely, the encoding proposed in Section 4 works both in $\mathcal{I}_{\mathsf{DY+AGh}}$ and in the inference system made up of (Eq) and (C) where $\mathcal{VF} = \{0, +, \mathsf{h}, -\}$. In [8], it is formally shown that for constraint systems involving no function symbols outside $sig(\mathsf{E})$ (as the one built in our encoding), undecidability in the latter inference system implies undecidability in $\mathcal{I}_{\mathsf{DY+AGh}}$.

4 Undecidability for AGh

We use the following formulation of Hilbert's 10^{th} problem, known to be undecidable [7]. 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 $x_i^2 = x_j$.
OUTPUT: Does S have a solution over $\mathbb{Z}$?

Given an instance S of Hilbert's 10^{th} problem with n free 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 $\mathcal{I}_{\mathsf{DY+AGh}}$.

We choose to encode an integer v in a ground term t by $\mathcal{N}(a, t)$ (see Definition 4). 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 9). In the second part of our encoding (Section 4.2), we deal with the equations of S: each one is encoded by a deductibility 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 $i = 1, \dots, n$, the constraint system $\mathcal{A}(n)$ contains the following five deductibility constraints whose free variables are X_i, Y_i, and Z_i:

$$\mathsf{h}^{p+n+2}(a) \Vdash \mathsf{h}^{p+n+2}(X_i) \qquad (\tau_1)$$

$$\mathsf{h}^{p+n+2}(a) \Vdash \mathsf{h}^{p+n+2}(Z_i) \qquad (\tau_1)$$

$$\mathsf{h}^{p+n+1}(b),\ \mathsf{h}^{p+n+2}(a) \Vdash \mathsf{h}^{p+n+1}(Y_i) \qquad (\tau_1)$$

$$\mathsf{h}^{p+n}(a+b),\ \mathsf{h}^{p+n+1}(b),\ \mathsf{h}^{p+n+2}(a) \Vdash \mathsf{h}^{p+n}(X_i + Y_i) \qquad (\tau_2)$$

$$\begin{aligned}&\mathsf{h}^{p+n-i}(X_i + b),\ \dots,\ \mathsf{h}^{p+n-2}(X_2 + b),\ \dots,\ \mathsf{h}^{p+n-1}(X_1 + b),\\ &\qquad \mathsf{h}^{p+n}(a+b),\ \mathsf{h}^{p+n+1}(b),\ \mathsf{h}^{p+n+2}(a) \Vdash \mathsf{h}^{p+n-i}(Z_i + Y_i) \qquad (\tau_3)\end{aligned}$$

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).

The idea of our encoding is first to ensure that if σ is a solution to $\mathcal{A}(n)$, the terms $X_i\sigma$ and $Z_i\sigma$ contain no occurrence of b, and the terms $Y_i\sigma$ contain no occurrence of a. This is ensured by the deductibility constraints of type τ_1. Then, thanks to the constraints of type τ_2, we ensure that $\mathcal{N}(a, X_i\sigma) = \mathcal{N}(b, Y_i\sigma)$. Lastly, the constraints of type τ_3 allows us to encode the products. Note that, each time, only the last term introduced on the left-hand side of the deductibility constraint is relevant to build the target term. Indeed only the terms of the form $\mathsf{h}^{k'}(\dots)$ with $k' \leq k$ are relevant to build a term of the form $\mathsf{h}^k(t)$. The terms $\mathsf{h}^{k'}(\dots)$ with $k' > k$ have no impact: they can not contribute to $\mathcal{N}(c, t\sigma)$ for any constant c.

Example 8 *We illustrate the first part of our construction with $n = 3$. We gather together constraints of the same type.*

$$\mathcal{A}_1(3) := \begin{cases} \mathsf{h}^8(a) \Vdash \mathsf{h}^8(X_1) & \mathsf{h}^8(a) \Vdash \mathsf{h}^8(Z_1) & \mathsf{h}^7(b),\ \mathsf{h}^8(a) \Vdash \mathsf{h}^7(Y_1) \\ \mathsf{h}^8(a) \Vdash \mathsf{h}^8(X_2) & \mathsf{h}^8(a) \Vdash \mathsf{h}^8(Z_2) & \mathsf{h}^7(b),\ \mathsf{h}^8(a) \Vdash \mathsf{h}^7(Y_2) \\ \mathsf{h}^8(a) \Vdash \mathsf{h}^8(X_3) & \mathsf{h}^8(a) \Vdash \mathsf{h}^8(Z_3) & \mathsf{h}^7(b),\ \mathsf{h}^8(a) \Vdash \mathsf{h}^7(Y_3) \end{cases}$$

$$\mathcal{A}_2(3) := \begin{cases} \mathsf{h}^6(a+b),\ \mathsf{h}^7(b),\ \mathsf{h}^8(a) \Vdash \mathsf{h}^6(X_1+Y_1) \\ \mathsf{h}^6(a+b),\ \mathsf{h}^7(b),\ \mathsf{h}^8(a) \Vdash \mathsf{h}^6(X_2+Y_2) \\ \mathsf{h}^6(a+b),\ \mathsf{h}^7(b),\ \mathsf{h}^8(a) \Vdash \mathsf{h}^6(X_3+Y_3) \end{cases}$$

$$\mathcal{A}_3(3) := \begin{cases} \mathsf{h}^5(X_1+b),\ \mathsf{h}^6(a+b),\ \mathsf{h}^7(b),\ \mathsf{h}^8(a) \Vdash \mathsf{h}^5(Z_1+Y_1) \\ \mathsf{h}^4(X_2+b),\ \mathsf{h}^5(X_1+b),\ \mathsf{h}^6(a+b),\ \mathsf{h}^7(b),\ \mathsf{h}^8(a) \Vdash \mathsf{h}^4(Z_2+Y_2) \\ \mathsf{h}^3(X_3+b),\ \mathsf{h}^4(x_2+b),\ \mathsf{h}^5(X_1+b),\ \mathsf{h}^6(a+b),\ \mathsf{h}^7(b),\ \mathsf{h}^8(a) \Vdash \mathsf{h}^3(Z_3+Y_3) \end{cases}$$

Lemma 9 *Let $n \in \mathbb{N}$ and σ a solution to $\mathcal{A}(n)$ in $\mathcal{I}_{\mathsf{DY+AGh}}$. We have:*

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

PROOF. Let σ be a solution to $\mathcal{A}(n)$. Firstly, constraints of type τ_1 ensure that $\mathcal{N}(b, X_i\sigma) = \mathcal{N}(b, Z_i\sigma) = 0$ and $\mathcal{N}(a, Y_i\sigma) = 0$. Thanks to the constraints of type τ_2, we have that $\mathcal{N}(a, X_i\sigma) + \mathcal{N}(a, Y_i\sigma) = \mathcal{N}(b, X_i\sigma) + \mathcal{N}(b, Y_i\sigma)$. Putting these two results together allow us to conclude for *(1)*. Now, we consider the i^{th} constraint of type τ_3. This constraint ensures that there exists $z \in \mathbb{Z}$ such that: $z \times (\mathcal{N}(b, X_i\sigma) + 1) = \mathcal{N}(b, Y_i\sigma) + \mathcal{N}(b, Z_i\sigma)$ and $z \times \mathcal{N}(a, X_i\sigma) = \mathcal{N}(a, Y_i\sigma) + \mathcal{N}(a, Z_i\sigma)$. Thanks to *(1)* and the fact that $\mathcal{N}(b, X_i\sigma) = \mathcal{N}(b, Z_i\sigma) = \mathcal{N}(a, Y_i\sigma) = 0$, we conclude for *(2)*. □

4.2 Encoding Equations of S

In this section, we describe the part $\mathcal{B}(S)$ of our coding which really depends on $S = \{e_1, \ldots, e_p\}$. $\mathcal{B}(S)$ contains one deductibility constraint per equation, denoted by $\mathsf{d}_1, \ldots, \mathsf{d}_p$. We let T_0 the knowledge that the intruder has obtained at the end of the first part of our coding, *i.e.*

$$T_0 = \{\mathsf{h}^{p+n-j}(X_j + b) \mid 1 \leq j \leq n\} \cup \{\mathsf{h}^{p+n}(a+b),\ \mathsf{h}^{p+n+1}(b),\ \mathsf{h}^{p+n+2}(a)\}.$$

We 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. Their role are just to prevent addition. They ensure that we only use one time the term introduced with this constant to build the target term of the corresponding deductibility constraint.

- if e_k='$x_i = m$" then $T_k = T_{k-1} \cup \{\mathsf{h}^{p-k}(X_i) + c_k\}$ and $\mathsf{d}_k = T_k \Vdash \mathsf{h}^{p-k}(ma) + c_k$,
- if e_k="$x_i + x_{i'} = x_j$" then $T_k = T_{k-1} \cup \{\mathsf{h}^{p-k}(X_i + X_{i'}) + c_k\}$
  and $\mathsf{d}_k = T_k \Vdash \mathsf{h}^{p-k}(X_j) + c_k$,
- if e_k="$x_i = x_j^2$ " then $T_k = T_{k-1} \cup \{\mathsf{h}^{p-k}(X_i) + c_k\}$, $\mathsf{d}_k = T_k \Vdash \mathsf{h}^{p-k}(Z_j) + c_k$.

Example 10 *Let $S_e = \{x_1 = 2,\ \ x_2^2 = x_3,\ \ x_2 + x_3 = x_1\}$. We obtain:*

$$\mathcal{B}(S_e) := \begin{cases} \hfill \mathsf{h}^2(X_1) + c_1,\ T_0 \Vdash \mathsf{h}^2(2a) + c_1 \\ \hfill \mathsf{h}(X_3) + c_2,\ \mathsf{h}^2(X_1) + c_1,\ T_0 \Vdash \mathsf{h}(Z_2) + c_2 \\ X_2 + X_3 + c_3,\ \mathsf{h}(X_3) + c_2,\ \mathsf{h}^2(X_1) + c_1,\ T_0 \Vdash \mathsf{h}(X_1) + c_3 \end{cases}$$

Proposition 11 *Let S be a set of equations (over n variables) and $\mathcal{C}(S)$ be the constraint system $\mathcal{A}(\mathsf{n}) \cup \mathcal{B}(S)$. 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 $\mathcal{I}_{\mathsf{DY+AGh}}$.

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 $\sigma = \{X_1 \mapsto v_1 a, \ldots, X_n \mapsto v_n a, Y_1 \mapsto v_1 b, \ldots, Y_n \mapsto v_n b, Z_1 \mapsto v_1^2 a, \ldots, Z_n \mapsto v_n^2 a\}$, we prove that σ is a solution to $\mathcal{C}(S)$. To do this, we show that for each constraint $T \Vdash u \in \mathcal{C}(S)$, there exists an $\mathcal{I}_{\mathsf{DY+AGh}}$-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 9, we have $\mathcal{N}(a, Z_i\sigma) = \mathcal{N}(a, X_i\sigma)^2$. 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:

$$T_{k-1}\sigma,\ \mathsf{h}^{p-k}(X_i\sigma) + c_k \Vdash \mathsf{h}^{p-k}(Z_j\sigma) + c_k$$

Note that c_k only appears in the term $\mathsf{h}^{p-k}(X_i\sigma) + c_k$ among all the terms in the hypotheses and c_k has to appear in the conclusion. We deduce that $\mathcal{N}(a, X_i\sigma) = \mathcal{N}(a, Z_j\sigma)$. From Lemma 9, we have $\mathcal{N}(a, Z_j\sigma) = \mathcal{N}(a, X_j\sigma)^2$ and we conclude. $\square$

5 Conclusion

In this paper, satisfiability 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 [14]), ACUNh (AGh plus the equation $-(x) = x$) and AGh. The undecidability result for AGh contrasts with the decidability one obtained for ACUNh [9]. 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 [5,3,13], the AC case seems to be very challenging.

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

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*, p. 499–514. Springer, 2002.

[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. 18th Symposium on Logic in Computer Science (LICS'03)*, p. 261–270. IEEE Comp. Soc. Press, 2003.

[4] H. Comon and V. Cortier. Tree automata with one memory set constraints and cryptographic protocols. *Theoretical Computer Science*, 331(1):143–214, 2005.

[5] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proc. 18th Symposium on Logic in Computer Science (LICS'03)*, p. 271–280. IEEE Comp. Soc. Press, 2003.

[6] 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.

[7] 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*, p. 323–378, 1976.

[8] S. Delaune. An undecidability result for AGh. Research Report LSV-06-02, Laboratoire Spécification et Vérification, ENS Cachan, France, 2006.

[9] S. Delaune, P. Lafourcade, D. Lugiez, and R. Treinen. Symbolic protocol analysis in presence of a homomorphism operator and *exclusive or*. In *Proc. International Colloquium on Automata, Languages and Programming (ICALP'06) — Part II*, volume 4052 of *LNCS*, p. 132–141. Springer, 2006.

[10] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 6. Elsevier and MIT Press, 1990.

[11] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory, 29(2):198-208*, 1983.

[12] N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Proc. Workshop on Formal Methods and Security Protocols (FMSE'99)*, 1999.

[13] 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.

[14] P. Narendran. Solving linear equations over polynomial semirings. In *Proc. 11th Symposium on Logic in Computer Science (LICS'96)*, p. 466–472. IEEE Comp. Soc. Press, 1996.

[15] 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.

[16] M. Tatebayashi, N. Matsuzaki, and D. B. Newman. Key distribution protocol for digital mobile communication systems. In *Proc. 9th Annual International Cryptology Conference (CRYPTO'89)*, volume 435 of *LNCS*, p. 324–333, Santa Barbara (California, USA), 1989. Springer.