INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Combining Intruder Theories

Yannick Chevalier — Michaël Rusinowitch*

N° 5495

February 2005

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--5495--FR+ENG

Combining Intruder Theories

Yannick Chevalier*† , Michaël Rusinowitch*‡

Thème 2 — Génie logiciel
et calcul symbolique
Projet Cassis

Rapport de recherche n° 5495 — February 2005 — 37 pages

Abstract: Most of the decision procedures for symbolic analysis of protocols are limited to a fixed set of algebraic operators associated with a fixed intruder theory. Examples of such sets of operators comprise XOR, multiplication/exponentiation, abstract encryption/decryption. In this report we give an algorithm for combining decision procedures for arbitrary intruder theories with disjoint sets of operators, provided that solvability of ordered intruder constraints, a slight generalization of intruder constraints, can be decided in each theory. This is the case for most of the intruder theories for which a decision procedure has been given. In particular our result allows us to decide trace-based security properties of protocols that employ any combination of the above mentionned operators with a bounded number of sessions.

Key-words: Cryptographic protocols, combination of decision procedures, equational theories

* supported by AVISPA IST-2001-39252, ACI-SI SATIN, ACI-Jeunes Chercheurs Crypto
† IRIT, Team LiLac, Université Paul Sabatier, France. email: ychevali@irit.fr
‡ LORIA-INRIA-Lorraine, France. email: rusi@loria.fr

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy (France)
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Combinaison de théories d'intrus

Résumé : La plupart des procédures de décision pour l'analyse symbolique de protocoles cryptographiques s'appliquent à un ensemble fixé d'opérateurs algébriques liés à une théorie équationnelle fixée. C'est le cas par exemple du OU-exclusif, du couple multiplication/exponentiation, des opérateurs de chiffrement/déchiffrement abstraits. Dans ce rapport nous donnons un algorithme permettant de combiner des procédures de décision pour des systèmes d'intrus et des théories équationnelles arbitraires tant que leurs d'opérateurs sont disjoints et que la satisfaisabilité de contraintes d'intrus étendues par des contraintes d'ordre est décidable dans chaque sous-théorie. C'est le cas pour la plupart des systèmes d'intrus qui ont été étudiés. En particulier notre résultat permet de décider les propriétés de secret et d'authentification des protocoles qui combinent toutes les opérations mentionnées ci-dessus pour un nombre borné de sessions.

Mots-clés : Protocoles cryptographiques, combinaison de procédures de décision, théories équationnelles

1 Introduction

1.1 Algebraic operators for cryptographic protocols analysis

Recently many procedures have been proposed to decide insecurity of cryptographic protocols in the Dolev-Yao model w.r.t. a finite number of protocol sessions [3, 6, 27, 25, 21]. Among the different approaches the symbolic ones [25, 10, 15, 5] are based on reducing the problem to constraint solving in a term algebra. This reduction has proved to be quite effective on standard benchmarks [11] and also permitted to discover new flaws on several protocols [5].

However while most formal analysis of security protocols abstracts from low-level properties, *i.e.* certain algebraic properties of encryption such as the multiplicativity of RSA or the properties induced by chaining methods for block ciphers, many real attacks and protocol weaknesses rely on these properties (for a survey see [16]). For attacks exploiting the *XOR* properties in the context of mobile communications see [8]. Also the specification of *Just Fast Keying* protocol (an alternative to IKE) in [1] employs a set constructor that is idempotent and commutative and a Diffie-Hellman exponentiation operator with the property $(g^y)^z = (g^z)^y$.

At an intermediate level encryption and decryption operations are sometimes defined by explicit constructors and destructors as in [24, 17]. This allows to have a simple communication scheme similar to the applied pi calculus of Abadi and Fournet [2]. This approach has also the advantage of revealing some new flaws in protocols [24]. It can be handled by some analysis tool (at least for basic encryption/decryption theory) such as the NRL analyzer [23] and TRUST system [3].

In this report we present a general procedure for deciding security of protocols in presence of algebraic properties. This procedure relies on the combination of constraint solving algorithm for disjoint intruder theories, provided that solvability of ordered intruder constraints, a slight generalization of intruder constraints, can be decided in each theory. Such combination algorithm already exists for solving E-unification problems [28, 4]. We have extended it in order to solve intruder constraints on disjoint signatures. This extension is non trivial since intruder deduction rules allow one to build *contexts* above terms and therefore add some second-order features to the *standard* first-order E-unification problem.

Our approach is more modular than the previous ones and it allows us to decide interesting intruder theories that could not be considered before by reducing them to simpler and independant theories. For instance it allows one to combine the exponentiation with abelian group theory of [26] with the Xor theory of [9]. This allows one to decide security protocols at a more concrete level where encryption is described by mathematical functions.

1.2 A protocol with several algebraic operators

We consider in this section the Needham-Schroeder Public-Key protocol. This well-known protocol is described in the Alice and Bob notation by the following sequence of messages,

where the comma denotes a pairing of messages and $\{M\}Ka$ denotes the encryption by the public key K_a of A.

$$A \to B : \{A, N_a\}K_b$$
$$B \to A : \{N_a, N_b\}K_a$$
$$A \to B : \{N_b\}K_b$$

Assume now that the encryption algorithm follows El-Gamal encryption scheme. The public key of A is defined by three publicly-available parameters: a modulus p_a, a base g_a and the proper public key $g^a \mod p_a$. The private key of A is a. Denoting $\exp_p$ the exponentiation modulo p and $\times_p$ the multiplication modulo $\varphi(p)$, and with new nonces k_1, k_2 and k_3 we can rewrite the protocol as:

$$A \to B : \exp_{p_b}(g_b, k_1), (A, N_a) \oplus \exp_{p_b}(\exp_{p_b}(g_b, b), k_1)$$
$$B \to A : \exp_{p_a}(g_a, k_2), (N_a, N_b) \oplus \exp_{p_a}(\exp_{p_a}(g_a, a), k_2)$$
$$A \to B : \exp_{p_b}(g_b, k_3), (N_b) \oplus \exp_{p_b}(\exp_{p_b}(g_b, b), k_3)$$

In this simple exemple we would like to model the group properties of the Exclusive-or ($\oplus$), the associativity of exponential ($(x^y)^z = x^{y \times z}$), the group property of the exponents. Several works have already been achieved toward taking into account these algebraic properties for detecting attacks on a bounded number of sessions. Some procedures have been proposed for specific theories like Exclusive-or, abelian goups (with exponential), ... However none of these can analyse protocols combining several algebraic operators like the example above. The algorithm given in this paper will permit to decide the trace-based security properties of such protocols.

1.3 Examples of intruder theories

A convenient way to specify intruder theories in the context of cryptographic protocols is by giving a set L of *deduction rules* that tell how the intruder can construct new messages from the one she already knows and a set of *equational laws* $\mathcal{E}$ that are verified by the functions that are employed in messages. These equations will be further processed as rewrite rules in order to obtain a (possibly infinite) rewrite system R. We give here examples of intruder theories. These examples are developed in Section 7.

1.3.1 Dolev Yao with explicit destructors

The intruder is given with a pairing operator and projections to retrieve the components of a pair. There is a symmetric encryption operator $\mathrm{se}(\_, \_)$ and an operator $\mathrm{sd}(\_, \_)$ for the decryption algorithm too. For conciseness we omit the public-key encryption specification.

$$L_{DY} \left\{ \begin{array}{rcl} x, y & \to & \langle x, y \rangle \\ x & \to & \pi_1(x) \\ x & \to & \pi_2(x) \\ x, y & \to & \mathrm{se}(x, y) \\ x, y & \to & \mathrm{sd}(x, y) \end{array} \right. \qquad \mathcal{E}_{DY} \left\{ \begin{array}{rcl} \pi_1(\langle x, y \rangle) & = & x \\ \pi_2(\langle x, y \rangle) & = & y \\ \mathrm{sd}(\mathrm{se}(x, y), y) & = & x \end{array} \right.$$

Note that this theory is itself the union of two more simple theories, one with the pairing operator and the projections and the other with the explicit encryption and decryption operators.

1.3.2 XOR theory

The theory of the *eXclusive-OR* operator is given over the signature $\mathcal{F}_\oplus = \{0, \oplus\}$ by the following set $L_\oplus$ of deduction rules and equations $\mathcal{E}_\oplus$ over terms.

$$L_\oplus \left\{ \begin{array}{rcl} x, y & \rightarrow & x \oplus y \\ & \rightarrow & 0 \end{array} \right. \qquad \mathcal{E}_\oplus \left\{ \begin{array}{rcl} (x \oplus y) \oplus z & = & x \oplus (y \oplus z) \\ x \oplus y & = & y \oplus x \\ 0 \oplus x & = & x \\ x \oplus x & = & 0 \end{array} \right.$$

1.3.3 Abelian group theory.

This intruder may treat messages as elements of an abelian group. We assume here there is only one such group and that the composition law is $\cdot \times \cdot$, the inverse law is $\mathrm{i}(\cdot)$ and the neutral element is denoted 1.

$$L_\times \left\{ \begin{array}{rcl} & \rightarrow & 1 \\ x & \rightarrow & \mathrm{i}(x) \\ x, y & \rightarrow & x \times y \end{array} \right. \qquad \mathcal{E}_\times \left\{ \begin{array}{rcl} (x \times y) \times z & = & x \times (y \times z) \\ x \times y & = & y \times x \\ 1 \times x & = & x \\ x \times \mathrm{i}(x) & = & 1 \end{array} \right.$$

1.3.4 Exponential and abelian groups

We consider multiplication and exponentiation in a finite but very large ring of order p. We assume $\varphi(p)$ (φ being the Euler function) is publicly known. This is consistent with Diffie-Hellman scheme where p is prime and publicly available. We model this order as a parameter of the $\times_p$ (multiplication modulo $\varphi(p)$), i_p (inverse modulo $\varphi(p)$) and $\exp_p$ (exponentiation modulo p) operation. In the following we fix the parameter p. This permits to simplify notation by omitting this parameter in operator's notations. An in-depths analysis of the exponential taking into account the availability of $\varphi(p)$ is out of the scope of this report.

The equational theory $\mathcal{E}_{\exp}$ we consider on the signature $\mathcal{F}_{\exp} = \{\exp(\_,\_), \mathrm{i}(\_), \_ \times \_\}$ is given by the following axioms (richer axiomatizations lead to undecidability [22]).

$$\mathcal{E}_{\exp} = \left\{ \begin{array}{rcl} \exp(x, 1) & = & x \\ \exp(\exp(x, y), z) & = & \exp(x, y \times z) \\ (x \times y) \times z & = & x \times (y \times z) \\ x \times y & = & y \times x \\ 1 \times x & = & x \\ x \times \mathrm{i}(x) & = & 1 \end{array} \right.$$

Since we assume $\varphi(p)$ is known the intruder may compute products and inverse modulo $\varphi(p)$. The possible deductions are given by the set $L_{\exp}$ of deduction rules.

$$L_{\exp} = \left\{ \begin{array}{rcl} & \rightarrow & 1 \\ x, y & \rightarrow & \exp(x, y) \\ x, y & \rightarrow & x \times y \\ x & \rightarrow & \mathrm{i}(x) \end{array} \right.$$

1.4 Related works

Recently several protocol decision procedures have been designed for handling algebraic properties in the Dolev-Yao model [22, 7, 13, 9]. These works have been concerned by fixed equational theories corresponding to a fixed intruder power. A couple of works only have tried to derive generic decidability results for *class* of intruder theories. For instance, in [17] Delaune and Jacquemard consider the class of *public collapsing* theories. These theories have to be presented by rewrite systems where the right-hand side of every rule is a ground term or a variable, which is a strong restriction. Comon and Treinen [14, 12] have also investigated general conditions on theories for deciding insecurity with passive intruders.

1.5 Outline

In Section 2 we will first define basic notions about terms, substitutions and ordered term rewriting. Then we introduce the notion of subterm values which is a notion of subterms specific to this report, and we give some properties of this notion with respect to replacement.

In Section 3 we first give our modelisation of an intruder in Subsection 3.1. We also prove the existence of special sequences of deductions called *well-formed derivations*. Then we give the model for the cryptographic protocols we plan to analyze (Subsection 3.2) and the reduction of some trace-based security properties to the feasability of an execution. Finally we define constraint systems in Subsection 3.3, the satisfiability of which corresponding to the feasability of an execution of a protocol.

In Section 4 we define with respect to a constraint system $\mathcal{C}$ a special kind of substitutions called *bound substitutions*. We prove that whenever a constraint system $\mathcal{C}$ is satisfiable it is satisfied by a bound substitution. We also prove that these substitutions are conservative with respect to the subterms of $\mathcal{C}$ *i.e.* after application of a bound substitution the number of subterms of $\mathcal{C}$ does not increase.

These results permit to define a combination algorithm for solving constraints systems for the union of two intruders over disjoint signatures in Section 5. We prove its soundness and completeness. The main disadvantage of this algorithm is that the constraint systems in the sub-systems are not necessarily deterministic. This is a major drawback since all decision procedures given so far assume the constraint systems are deterministic (except in [29] but the long version [26] re-introduces a notion that has to be dynamically verified during the verification process.)

This lead us to prove that the combination algorithm can be adapted so that it suffices to decide the satisfiability of deterministic constraint systems in sub-theories. We have prefer to put the definitions and lemmas necessary for this proof aside in Section 6 in order to alleviate the reading of the report.

Finally we give in Section 7 some complexity results for the satisfiability in some intruder theories. The bound we give are tight and permit to extend some previously known complexity results as well as to derive new ones.

2 Terms and subterms

2.1 Basic notions

We consider an infinite set of free constants C and an infinite set of variables $\mathcal{X}$. For all signatures $\mathcal{G}$ (*i.e.* a set of function symbols with arities), we denote by $\mathrm{T}(\mathcal{G})$ (resp. $\mathrm{T}(\mathcal{G},\mathcal{X})$) the set of terms over $\mathcal{G} \cup \mathrm{C}$ (resp. $\mathcal{G} \cup \mathrm{C} \cup \mathcal{X}$). The former is called the set of ground terms over $\mathcal{G}$, while the later is simply called the set of terms over $\mathcal{G}$. Variables are denoted by x, y, terms are denoted by s, t, u, v, and finite sets of terms are written $E, F, ...$, and decorations thereof, respectively. We abbreviate $E \cup F$ by E, F, the union $E \cup \{t\}$ by E, t and $E \setminus \{t\}$ by $E \setminus t$.

In a signature $\mathcal{G}$ a *constant* is either a free constant or a function symbol of arity 0 in $\mathcal{G}$. Given a term t we denote by $\mathrm{Var}(t)$ the set of variables occurring in t and by $\mathrm{Cons}(t)$ the set of constants occurring in t. We denote by $\mathrm{Atoms}(t)$ the set $\mathrm{Var}(t) \cup \mathrm{Cons}(t)$.A substitution σ is an involutive mapping from $\mathcal{X}$ to $\mathrm{T}(\mathcal{G},\mathcal{X})$ such that $\mathrm{Supp}(\sigma) = \{x | \sigma(x) \neq x\}$, the *support* of σ, is a finite set. The application of a substitution σ to a term t (resp. a set of terms E) is denoted $t\sigma$ (resp. $E\sigma$) and is equal to the term t (resp. E) where all variables x have been replaced by the term $x\sigma$. A substitution σ is *ground* w.r.t. $\mathcal{G}$ if the image of $\mathrm{Supp}(\sigma)$ is included in $\mathrm{T}(\mathcal{G})$.

An *equational presentation* $\mathcal{H} = (\mathcal{G}, A)$ is defined by a set A of equations $r = t$ with $r, t \in \mathrm{T}(\mathcal{G},\mathcal{X})$. For any equational presentation $\mathcal{H}$ the relation $=_{\mathcal{H}}$ denotes the equational theory generated by $(\mathcal{G}, A)$ on $\mathrm{T}(\mathcal{G},\mathcal{X})$, that is the smallest congruence containing all instances of axioms of A. By abuse of terminology we do not distinguish between an equational presentation $\mathcal{H}$ over a signature $\mathcal{G}$ and a set A of equations presenting it and denote both $\mathcal{H}$. We will also often refer to $\mathcal{H}$ as an equational theory (meaning the equational theory presented by $\mathcal{H}$). An equational theory $\mathcal{H}$ is *consistent* if there exists at least one model of $\mathcal{H}$ with more than one element. Equivalently a theory $\mathcal{H}$ is consistent if there does not exists two free constants x and y such that $x \neq y$ and $x =_{\mathcal{H}} y$.

The *syntactic subterms* of a term t are denoted $\mathrm{Sub}_{\mathrm{syn}}(t)$ and are defined recursively as follows. If t is a variable or a constant then $\mathrm{Sub}_{\mathrm{syn}}(t) = \{t\}$. If $t = f(t_1, \ldots, t_n)$ then $\mathrm{Sub}_{\mathrm{syn}}(t) = \{t\} \cup \bigcup_{i=1}^{n} \mathrm{Sub}_{\mathrm{syn}}(t_i)$. The *positions* in a term t are sequences of integers defined recursively as follows, ϵ being the empty sequence. The term t is at position ϵ in t. If u is a syntactic subterm of t at position p and if $u = f(u_1, \ldots, u_n)$ then u_i is at position $p \cdot i$ in t

for $i \in \{1, \ldots, n\}$. We denote by $t[p \leftarrow s]$ the term obtained by replacing in t the syntactic subterm at position p by s.

2.2 Definitions for the union of two signatures

In this paper, we consider 2 disjoint signatures $\mathcal{F}_1$ and $\mathcal{F}_2$, a consistent equational theory $\mathcal{E}_1$ (resp. $\mathcal{E}_2$) on $\mathcal{F}_1$ (resp. $\mathcal{F}_2$). We denote by $\mathcal{F}$ the union of the signatures $\mathcal{F}_1$ and $\mathcal{F}_2$, $\mathcal{E}$ the union of the theories $\mathcal{E}_1$ and $\mathcal{E}_2$. A term t in $\mathrm{T}(\mathcal{F}_1, \mathcal{X})$ (resp. in $\mathrm{T}(\mathcal{F}_2, \mathcal{X})$) is called a *pure* 1-term (resp. a pure 2-term). We denote by $\mathrm{Sign}(\cdot)$ the function that associates to each term $t \notin \mathrm{C} \cup \mathcal{X}$ the signature ($\mathcal{F}_1$ or $\mathcal{F}_2$) of its root symbol. For $t \in \mathrm{C} \cup \mathcal{X}$ we define $\mathrm{Sign}(t) = \perp$, with $\perp$ a new symbol. The term s is *alien* to u if $\mathrm{Sign}(s) \neq \mathrm{Sign}(u)$. We now introduce a notion of subterm values of a term t. These are syntactic subterms of t that are either equal to t or a strict maximal alien syntactic subterm of a subterm value of t.

Definition 1 *(factors) The set of* factors *of a term t is denoted* $\mathrm{Factors}(t)$ *and is the set of maximal syntactic strict subterms of t that are either alien to t or atoms.*

Example 1 *Consider $\mathcal{F}_1 = \{\oplus, a, b, c\}$ and $\mathcal{F}_2 = \{f\}$ where f has arity 1. Then*

$$\begin{cases} \mathrm{Factors}(f(f(a)) \oplus (b \oplus c)) &= \{f(f(a)), b, c\} \\ \mathrm{Factors}(f(f(f(b) \oplus c))) &= \{f(b) \oplus c\} \\ \mathrm{Factors}(0) &= \emptyset \end{cases}$$

We now define the notion of *subterm values*.

Definition 2 *(Subterms) Given a term t, the set of its* subterm values *is denoted by* $\mathrm{Sub}(t)$ *and is defined recursively by* $\mathrm{Sub}(t) = \{t\} \cup \bigcup_{u \in \mathrm{Factors}(t)} \mathrm{Sub}(u)$.

By extension, for a set of terms E, the set $\mathrm{Sub}(E)$ is defined as the union of the subterms values of the elements of E.

Example 2 *Consider $\mathcal{F}_1$ and $\mathcal{F}_2$ as in Example 1. Then*

$$\begin{cases} \mathrm{Sub}(f(f(a)) \oplus (b \oplus c)) &= \{f(f(a)) \oplus (b \oplus c), f(f(a)), a, b, c\} \\ \mathrm{Sub}(f(f(f(b) \oplus c))) &= \{f(f(f(b) \oplus c)), f(b) \oplus c, f(b), b, c\} \\ \mathrm{Sub}(0) &= \{0\} \end{cases}$$

This shows the difference with the notion of syntactic subterms.

In the rest of this paper and unless otherwise indicated, *the notion of subterm will refer to subterm values.*

2.3 Congruences and ordered rewriting

In this subsection we shall introduce the notion of *ordered rewriting* [18] which is a useful tool that has been utilized (*e.g.* [4]) for proving the correctness of combination of unification algorithms. Let $<$ be a simplification ordering on $\mathrm{T}(\mathcal{G})$ [1] assumed to be total on $\mathrm{T}(\mathcal{G})$ and such that

- the minimum for $<$ is a constant $\mathrm{c}_{\min} \in \mathrm{C}$;
- non-free constants are smaller than any non-constant ground term.

Given a possibly infinite set of equations $\mathcal{O}$ on the signature $\mathrm{T}(\mathcal{G})$ we define the ordered rewriting relation $\rightarrow_{\mathcal{O}}$ by $s \rightarrow_{\mathcal{O}} s'$ iff there exists a position p in s, an equation $l = r$ in $\mathcal{O}$ and a substitution τ such that $s = s[p \leftarrow g\tau]$, $s' = s[p \leftarrow r\tau]$, and $g\tau > d\tau$.

It has been shown (see [18]) that by applying the *unfailing completion procedure* to a set of equations $\mathcal{H}$ we can derive a (possibly infinite) set of equations $\mathcal{O}$ such that:

1. the congruence relations $=_{\mathcal{O}}$ and $=_{\mathcal{H}}$ are equal on $\mathrm{T}(\mathcal{F})$.
2. the ordered rewrite relation $\rightarrow_{\mathcal{O}}$ is convergent (*i.e.* terminating and confluent) on $\mathrm{T}(\mathcal{F})$.

We shall say that $\mathcal{O}$ is an *o-completion* of $\mathcal{H}$.

From now for sake of conciseness when we will say *"the rewrite system $\rightarrow_{\mathcal{O}}$"* this will mean "the ordered rewrite relation $\rightarrow_{\mathcal{O}}$", when will say *"by convergence of $\mathcal{O}$"*, we will mean "by convergence of $\rightarrow_{\mathcal{O}}$ on ground terms".

The rewrite system $\rightarrow_{\mathcal{O}}$ being convergent on ground terms we can define $(t)\downarrow_{\mathcal{O}}$ as the unique normal form of the ground term t for $\rightarrow_{\mathcal{O}}$. A ground term t is in *normal form*, or *normalized*, if $t = (t)\downarrow_{\mathcal{O}}$. Given a ground substitution σ we denote by $(\sigma)\downarrow_{\mathcal{O}}$ the substitution with the same support such that for all variables $x \in \mathrm{Supp}(\sigma)$ we have $x(\sigma)\downarrow_{\mathcal{O}} = (x\sigma)\downarrow_{\mathcal{O}}$. A substitution σ is *normal* if $\sigma = (\sigma)\downarrow_{\mathcal{O}}$.

Notations in this report. Applying unfailing completion to $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$, it is easy to notice [4] that the set of generated equations R is the the disjoint union of the two systems R_1 and R_2 also obtained by applying unfailing completion procedures to $\mathcal{E}_1$ and to $\mathcal{E}_2$ respectively. (Since $\mathcal{F}_1 \cap \mathcal{F}_2 = \emptyset$ and the $\mathcal{E}_i$ are assumed to be consistent *i.e.* the identity $x =_{\mathcal{E}_i} y$ does not hold in either theory, the critical pair generation will produce only pure equations.) We denote $(t)\downarrow$ the normal form of a term t for the rewrite system $\rightarrow_R$. We denote by $\mathrm{C}_{\mathrm{spe}}$ the set containing the constants in $\mathcal{F}$ and $\mathrm{c}_{\min}$.

First let us show that when normalizing a term with R, we can handle equations introducing new variables by replacing these variables by the minimal constant $\mathrm{c}_{\min}$:

Lemma 1 *Assume that $g = d \in R$, $z \in \mathrm{Var}(d) \setminus \mathrm{Var}(g)$, and $s \rightarrow_R s'$ with $s = s[p \leftarrow g\tau]$, $s' = s[p \leftarrow d\tau]$, $g\tau > d\tau$. Let us define the substitution σ such that for all variables $x \neq z$ we have $x\sigma = x\tau$ and $z\sigma = \mathrm{c}_{\min}$. Then we have also: $s \rightarrow_R s[p \leftarrow d\sigma]$*

[1] by definition $<$ satisfies for all $s, t, u \in \mathrm{T}(\mathcal{G})$ $s < t[s]$ and $s < u$ implies $t[s] < t[u]$

PROOF. We only need to notice that $g\tau > d\tau$ implies $g\sigma > d\sigma$ since $g\tau = g\sigma$ and $d\tau > d\sigma$ by the monotonicity properties of simplification orderings. □

Lemma 2 *If $\mathcal{H}$ is a consistent equational theory then for any equation $g = d$ in a presentation of $\mathcal{H}$ with $g \neq d$ if there exists a substitution τ such that $g\tau > d\tau$ then g is not a variable.*

PROOF. By contradiction assume g is a variable and there exists a ground substitution τ such that $g\tau > d\tau$. By monotony we have $g \notin \mathrm{Var}(d)$. Let τ' be a substitution of support $\mathrm{Var}(d)$ and equal to τ on $\mathrm{Var}(d)$. Then for any ground term t we can build a substitution τ_t of support $\mathrm{Var}(d) \cup \{g\}$:

$$\begin{cases} g\tau_t &= t \\ x\tau_t &= x\tau' \text{ If } x \in \mathrm{Var}(d) \end{cases}$$

The equation $g = d$ then implies that all ground terms t are equal to $d\tau$. By transitivity of the equality all ground terms are equal, which contradicts $\mathcal{H}$ has a model with more than one element. □

The following lemma is an easy to prove but nonetheless fundamental result.

Lemma 3 *Assume a non-constant ground term t has all its factors in normal form. Then either $(t)\downarrow \in \mathrm{C_{spe}} \cup \mathrm{Factors}(t)$ or $\mathrm{Sign}(t) = \mathrm{Sign}((t)\downarrow)$ and $\mathrm{Factors}((t)\downarrow) \subseteq \mathrm{C_{spe}} \cup \mathrm{Factors}(t)$.*

PROOF. The assumption t is a non-constant ground term implies $\mathrm{Factors}(t) \neq \emptyset$ and $\mathrm{Sign}(t) \neq \bot$. If t is in normal form the result is trivial.

Otherwise consider a sequence of applications of rules of R: $t = t_0 \rightarrow_R \ldots \rightarrow_R t_n = t'$ and assume that at each step i for $i \in \{1, \ldots, n\}$ the term t_i is minimal for $<$ among the terms r such that $t_{i-1} \rightarrow_R r$.

Let $0 \leq i \leq n$ be the last step in this sequence such that $t_i \in \mathrm{C_{spe}} \cup \mathrm{Factors}(t)$ or $\mathrm{Sign}(t) = \mathrm{Sign}(t_i)$ and $\mathrm{Factors}(t_i) \subseteq \mathrm{C_{spe}} \cup \mathrm{Factors}(t)$.

By contradiction assume $i < n$ and assume the equation $g = d \in R$ is applied on t_i with substitution τ to yield t_{i+1} with $g\tau > d\tau$. The minimality of t_{i+1} among terms r such that $t_i \rightarrow_R t_{i+1}$ and Lemma 1 imply that for all variables x in $\mathrm{Var}(d) \setminus \mathrm{Var}(g)$ we have $x\tau = \mathrm{c_{min}}$. By construction of R the terms g and d are pure terms and constants in d and g are non-free constants.

By Lemma 2 g is not a variable and thus $\mathrm{Sign}(g\tau) \neq \bot$. The choice of $<$ implies if g is a constant then so is d. In this case they are both in $\mathrm{C_{spe}}$and we are done. Let us assume now the set of factors of $g\tau$ is not empty. Since the factors of t_i are in normal form the rule is applied *above* the factors of t_i. Thus we must have $\mathrm{Sign}(t_i) = \mathrm{Sign}(g)$ and $\mathrm{Factors}(g\tau) \subseteq \mathrm{C_{spe}} \cup \mathrm{Factors}(t_i)$. Thus for each variable $x \in \mathrm{Var}(g)$ either $x\tau \in \mathrm{Factors}(t_i)$ or $\mathrm{Factors}(x\tau) \subseteq \mathrm{Factors}(t_i)$ and $\mathrm{Sign}(x\tau) = \mathrm{Sign}(t_i)$.

The above remark implies that:

- either d is a variable and $d\tau$ is a factor of t or in $\mathrm{C_{spe}}$;

- or the factors of $d\tau$ are either in $\mathrm{C_{spe}}$ or are factors of t.

Now if the rule is applied at position ϵ on t and if we are in the first case we have $t_{i+1} \in \mathrm{Factors}(t_i) \cup \mathrm{C_{spe}}$. Else we necessarily have $\mathrm{Sign}(t_{i+1}) = \mathrm{Sign}(t_i)$ and the above cases imply that $\mathrm{Factors}(t_{i+1}) \subseteq \mathrm{C_{spe}} \cup \mathrm{Factors}(t_i)$. Both cases contradict the maximality of i and thus that $i < n$. Together with $t_n = (t)\!\downarrow$ this implies the Lemma. □

In the rest of this paper we will often used without justification the following consequence of Lemma 3. First we see that if the factors of a term t are in normal form we have:

$$\mathrm{Sub}((t)\!\downarrow) \subseteq (\mathrm{Sub}(t))\!\downarrow \cup \mathrm{C_{spe}}$$

By iterating along a bottom-up normalisation of a ground term t this inclusion also holds for any ground term t. A useful case is when t is not ground but a ground, normalized substitution σ is applied on t. In this case we have $\mathrm{Sub}(t\sigma) \subseteq \mathrm{Sub}(t)\sigma \cup \mathrm{Sub}(\sigma)$. Since the substitution σ is normal we have the following much used inclusion:

$$\mathrm{Sub}((t\sigma)\!\downarrow) \subseteq (\mathrm{Sub}(t)\sigma)\!\downarrow \cup \mathrm{Sub}(\sigma) \cup \mathrm{C_{spe}}$$

2.4 Normalisation and replacements

The following lemma states that if all factors of a term t are in normal form then the replacement of one of these factors commutes with the normalisation of t.

If Π is a set of positions in term t we denote by $t[\Pi \leftarrow v]$ the term obtained by putting v at all positions of t that are in Π. We denote $\delta_{u,v}$ the replacement of u by v such that if u appears at positions Π_u *as a subterm* (*i.e.* as a subterm value) of t then $t\delta_u = t[\Pi_u \leftarrow v]$. We denote δ_u the replacement $\delta_{u,\mathrm{c_{min}}}$.

Lemma 4 *Let t be a term such that its factors are in normal form. Let u and v be two ground terms with $u \in \mathrm{Factors}(t)$ with $u \neq (t)\!\downarrow$ and u, v are terms alien to t. Then $(t\delta_{u,v})\!\downarrow = ((t)\!\downarrow\delta_{u,v})\!\downarrow$.*

Proof. Consider a sequence of rewrite steps: $t \rightarrow_{s_1} t_1 \rightarrow_{s_2} t_2 \cdots \rightarrow_{s_n} t_n = (t)\!\downarrow$, using rules $s_1, \ldots s_n \in R$ and wlog assume $\mathrm{Sign}(t) = \mathcal{F}_1$, $\mathrm{Sign}(u) \neq \mathcal{F}_1$ and $\mathrm{Sign}(v) \neq \mathcal{F}_1$.

Since the terms in $\mathrm{Factors}(t)$ are in normal form we have that all r_i with $i \in \{1, \ldots, n\}$ are in R_1. Thus the left-hand side and the right-hand side of these rules are pure 1-terms. By Lemma 2 the left-hand side of r_i is not a variable x.

Thus u alien to t implies it is alien to all left-hand sides of rewrite rules applied. These remarks lead to the following claim.

Claim 1 *$t_{i-1} \rightarrow_{s_i} t_i$ implies $t_{i-1}\delta_{u,v} =_{R_1} t_i\delta_{u,v}$.*

Proof of the claim. Let π be the position in t_{i-1} at which the rule s_i is applied and let l_i be the left-hand side of r_i and Θ be the set of positions of variables in l_i. Let Π be the set of positions at which u appears as a subterm in t_{i-1}. The above remarks imply u is alien to l_i. Thus for all $p \in \Pi$ we have either π is not a prefix of p or there exists $\theta \in \Theta$ such that $\pi \cdot \theta$ is a prefix of p. This implies the claim. ♢

Iterating the claim along the sequence of rule applications for normalizing t yields $t\delta_{u,v} =_R (t)\downarrow\delta_{u,v}$ by transitivity of $=_R$. Since R is ground convergent this implies $(t\delta_{u,v})\downarrow =_R ((t)\downarrow\delta_{u,v})\downarrow$. □

A ground term s is said to be *bound* by σ to the term t in U if there exists $t \in U$ such that $(t\sigma)\downarrow = s$. A ground term s which is not bound to any term in U is said to be *free* in U. The following lemma permits us to define a new substitution after the replacement of a free subterm of the solution σ.

Lemma 5 *Let t be a term and σ be a normalized substitution. Assume s is free in* $\mathrm{Sub}(t)$ *for σ and let $\sigma' = (\sigma\delta_s)\downarrow$. We have:*

$$((t\sigma)\downarrow\delta_s)\downarrow = (t\sigma')\downarrow$$

Proof. Since R is ground convergent it is sufficient to prove:

$$(t\sigma)\downarrow\delta_s =_R t\sigma'$$

For all variables x we have $x\sigma' =_R x(\sigma\delta_s)$ by definition of σ', and thus:

$$t\sigma' =_R t(\sigma\delta_s)$$

Since s is free and normalized, there is no subterm r of t such that $r\sigma = s$. Thus:

$$t(\sigma\delta_s) =_R (t\sigma)\delta_s$$

Moreover we have $(t\sigma)\downarrow =_R t\sigma$. Since σ is normalized we have $\mathrm{Sub}((t\sigma)\downarrow) \subseteq (\mathrm{Sub}(t)\sigma)\downarrow \cup \mathrm{Sub}(\sigma)$ and $\mathrm{Sub}(t\sigma) \subseteq \mathrm{Sub}(t)\sigma \cup \mathrm{Sub}(\sigma)$. Since s is free and normalized it is neither in $\mathrm{Sub}(t)\sigma$ nor in $(\mathrm{Sub}(t)\sigma)\downarrow$. Thus we have:

$$((t\sigma)\downarrow)\delta_s =_R (t\sigma)\delta_s$$

Hence we have $(t\sigma)\downarrow\delta_s =_R t\sigma'$ which completes the proof. □

Lemma 6 *For all normal substitutions σ, for all terms m and for all $s \in \mathrm{Sub}((m\sigma)\downarrow)$ one of the following holds:*

- $s \in \mathrm{C_{spe}}$*;*
- *There is $u \in \mathrm{Sub}(m)$ such that $(u\sigma)\downarrow = s$ and $\mathrm{Sign}(u) = \mathrm{Sign}(s)$;*
- *There exists $x \in \mathrm{Var}(m)$ such that $s \in \mathrm{Sub}(x\sigma)$.*

Proof. Let m and s be two terms and let σ be a ground substitution such that $s \in \mathrm{Sub}((m\sigma)\downarrow)$. We have

$$\mathrm{Sub}((m\sigma)\downarrow) \subseteq (\mathrm{Sub}(m)\sigma)\downarrow \cup \mathrm{Sub}(\mathrm{Var}(m)\sigma) \cup \mathrm{C_{spe}}$$

Assume there exists no $x \in \mathrm{Var}(m)$ such that $s \in \mathrm{Sub}(x\sigma)$ and $s \notin \mathrm{C_{spe}}$. Let $u \in \mathrm{Sub}(m)$ be minimal for the subterm relation such that $(u\sigma)\downarrow = s$. The above inclusion and $s \notin \mathrm{Sub}(\mathrm{Var}(m)\sigma) \cup \mathrm{C_{spe}}$ imply u is well-defined. If it is a free constant we have necessarily $u = s$ and $\mathrm{Sign}(u) = \mathrm{Sign}(s) = \bot$. Assume now u is neither a constant or a variable and thus $\mathrm{Factors}(u)$ is not empty.

By minimality of u we have $s \notin ((\mathrm{Sub}(u) \setminus \{u\})\sigma)\downarrow$. Thus for all v in $\mathrm{Sub}(u) \setminus \{u\}$ the above inclusion (replacing m by v) imply $s \notin \mathrm{Sub}((v\sigma)\downarrow)$. Consider now a bottom-up normalisation of $u\sigma$ stopping at factors of u and let t be the obtained term. By Lemma 3 and $s \notin \mathrm{C_{spe}} \cup \mathrm{Factors}(t)$ we have $\mathrm{Sign}(t) = \mathrm{Sign}(s)$. By definition of t we have $\mathrm{Sign}(t) = \mathrm{Sign}(u)$ and therefore there exists $u \in \mathrm{Sub}(m)$ such that $(u\sigma)\downarrow = s$ and $\mathrm{Sign}(u) = \mathrm{Sign}(s)$. □

3 Protocols, intruders and constraint systems

Security of a given protocol is assessed with respect to a class of environments in which the protocol is executed. Dolev and Yao [19] have described the environment not in terms of possible attacks on the protocol but by the deduction an intruder attacking a protocol execution is able to perform.

In Subsection 3.1 we define an extension of Dolev-Yao's model to arbitrary operators that models the possible deductions of the intruder. In Subsection 3.2 we describe the execution of a protocol within an hostile environment controlled by the intruder and in Subsection 3.3 we describe how we model this execution by constraint systems.

3.1 Intruder deduction systems

3.1.1 Deduction rules

We shall model messages as ground terms and intruders deduction rules as rewrite rules on sets of messages representing the knowledge of an intruder. The intruder derives new messages from a given (finite) set of messages by applying intruder rules. Since we assume some equational axioms $\mathcal{H}$ are satisfied by functions symbols in the signature, all these derivations have to be considered *modulo* the equational congruence $=_{\mathcal{H}}$ generated by these axioms.

An intruder deduction rule in our setting is specified by a term t in some signature $\mathcal{G}$. Given values for the variables of t the intruder is able to generate the corresponding instance of t.

Definition 3 *An* intruder system $\mathcal{I}$ *is given by a triple* $\langle \mathcal{G}, S, \mathcal{H} \rangle$ *where* $\mathcal{G}$ *is a signature,* $S \subseteq \mathrm{T}(\mathcal{G}, \mathcal{X})$ *and* $\mathcal{H}$ *is a set of equations between terms in* $\mathrm{T}(\mathcal{G}, \mathcal{X})$*. To each* $t \in S$ *we associate a* deduction rule $\mathrm{L}^t : \mathrm{Var}(t) \to t$ *and* $\mathrm{L}^{t,\mathrm{g}}$ *denotes the set of ground instances of the rule* L^t modulo $\mathcal{H}$*:*

$$\mathrm{L}^{t,\mathrm{g}} = \{l \to r \mid \exists \sigma, \textit{ground substitution on } \mathcal{G},\ l = \mathrm{Var}(t)\sigma \text{ and } r =_{\mathcal{H}} t\sigma\}$$

The set of rules $\mathrm{L}_{\mathcal{I}}$ *is defined as the union of the sets* $\mathrm{L}^{t,\mathrm{g}}$ *for all* $t \in S$*.*

Each rule $l \to r$ in $\mathrm{L}_{\mathcal{I}}$ defines an intruder deduction relation $\to_{l\to r}$ between finite sets of terms. Given two finite sets of terms E and F we define $E \to_{l\to r} F$ if and only if $l \subseteq E$ and $F = E \cup \{r\}$. We denote $\to_{\mathcal{I}}$ the union of the relations $\to_{l\to r}$ for all $l \to r$ in $L_{\mathcal{I}}$ and by $\to^*_{\mathcal{I}}$ the transitive closure of $\to_{\mathcal{I}}$. We simply denote by $\to$ the relation $\to_{\mathcal{I}}$ when there is no ambiguity about $\mathcal{I}$.

The next result will allow us to restrict our study to deductions with normalized terms:

Lemma 7 *We assume that R is a rewrite system that is terminating and confluent on ground terms such that $=_R$ and $=_{\mathcal{E}}$ are the same relations. Then given a set of ground terms E and a ground term t, there is a deduction $E \to F$ iff there is a deduction $(E)\downarrow \to (F)\downarrow$.*

For instance we can define $\mathcal{I}_\times = \langle\{\times, \mathrm{i}, 1\}, \{x \times y, \mathrm{i}(x), 1\}, \mathcal{E}_\times\rangle$ and we have $a, b, c \to_{\mathcal{I}_\times} a, b, c, c \times a$.

A *derivation* D of length n, $n \geq 0$, is a sequence of steps of the form $E_0 \to_{\mathcal{I}} E_0, t_1 \to_{\mathcal{I}} \cdots \to_{\mathcal{I}} E_n$ with finite sets of ground terms $E_0, \ldots E_n$, and ground terms $t_1, \ldots, t_n$, such that $E_i = E_{i-1} \cup \{t_i\}$ for every $i \in \{1, \ldots, n\}$. A derivation is *without stutter* if for all $i, j \in \{1, \ldots, n\}$, $t_i = t_j$ implies $i = j$. The term t_n is called the *goal* of the derivation. We define $\overline{E}^{\mathcal{I}}$ to be equal to the set $\{t \,|\, \exists F \text{ s.t. } E \to^*_{\mathcal{I}} F \text{ and } t \in F\}$ *i.e.* the set of terms that can be derived from E. If there is no ambiguity on the deduction system $\mathcal{I}$ we write $\overline{E}$ instead of $\overline{E}^{\mathcal{I}}$.

Let $\mathcal{O}$ be an o-completion of $\mathcal{H}$. By Lemma 7 we will assume from now that all the deduction rules generate terms that are normalized by $\to_{\mathcal{O}}$ and the goal and the initial set are in normal form for $\to_{\mathcal{O}}$.

3.1.2 Union of intruder deduction systems

Given a set of terms $S \subseteq \mathrm{T}(\mathcal{G}, \mathcal{X})$ we define the set of terms $\langle S\rangle$ to be the minimal set such that $S \subseteq \langle S\rangle$ and for all $t \in \langle S\rangle$ and for all substitutions σ with image included in $\langle S\rangle$, we have $t\sigma \in \langle S\rangle$. Hence terms in $\langle S\rangle$ are built by composing terms in S iteratively. We can prove easily that the intruder systems $\mathcal{I} = \langle \mathcal{G}, S, \mathcal{H}\rangle$ and $\mathcal{J} = \langle \mathcal{G}, \langle S\rangle, \mathcal{H}\rangle$ define the same sets of derivable terms, i.e. for all E we have $\overline{E}^{\mathcal{I}} = \overline{E}^{\mathcal{J}}$.

We want to consider now the union of 2 intruder systems: $\mathcal{I}_1 = \langle \mathcal{F}_1, S_1, \mathcal{E}_1\rangle$ and $\mathcal{I}_2 = \langle \mathcal{F}_2, S_2, \mathcal{E}_2\rangle$. In particular we are interested in the derivations obtained by using $\to_{\mathcal{I}_1} \cup \to_{\mathcal{I}_2}$. It can be noticed that $\langle S_1 \cup S_2\rangle = \langle\langle S_1\rangle \cup \langle S_2\rangle\rangle$. Hence by the remarks above the derivable terms using $\langle S_1 \cup S_2\rangle$ or $\langle S_1\rangle \cup \langle S_2\rangle$ are the same. For technical reason it will be more convenient to use $\langle S_1\rangle \cup \langle S_2\rangle$ for defining the union of 2 intruder systems:

Definition 4 *The* union of the two intruder systems $\mathcal{I}_1 = \langle \mathcal{F}_1, S_1, \mathcal{E}_1\rangle$ *and* $\mathcal{I}_2 = \langle \mathcal{F}_2, S_2, \mathcal{E}_2\rangle$ *is the intruder system* $\mathcal{U} = \langle \mathcal{F}_1 \cup \mathcal{F}_2, \langle S_1\rangle \cup \langle S_2\rangle, \mathcal{E}_1 \cup \mathcal{E}_2\rangle$.

From now we assume that deduction steps refer to the intruder system $\mathcal{U} = \langle \mathcal{F}_1 \cup \mathcal{F}_2, \langle S_1\rangle \cup \langle S_2\rangle, \mathcal{E}_1 \cup \mathcal{E}_2\rangle$.

3.1.3 Properties of one-step deductions

First we prove some properties that are *local* to a given deduction in a derivation.

To begin with, we prove that, informally, if the set of subterms changes after a deduction of a term by a rule in $\mathrm{L}^{u,\mathrm{g}}$, then the only change is the addition of t and $\mathrm{Sign}(u) = \mathrm{Sign}(t)$.

Lemma 8 *Let E and F be two finite sets of normalized terms and let $E \rightarrow_{\mathrm{L}^{u,\mathrm{g}}} F$ be a deduction and assume that $\mathrm{Sub}(E) \cup \mathrm{C}_{\mathrm{spe}} \neq \mathrm{Sub}(F) \cup \mathrm{C}_{\mathrm{spe}}$. Then $F = E, s$, with $\mathrm{Sub}(F) = \mathrm{Sub}(E) \cup \{s\}$ and $\mathrm{Sign}(u) = \mathrm{Sign}(s)$.*

Proof. Let the rule applied be:

$$u_1, u_2, \ldots, u_k \rightarrow (u(u_1, \ldots, u_k))\downarrow = t_i$$

and let $v = u(u_1, \ldots, u_k)$. Since E is normalized the u_i are in normal form. Since u is pure, by definition of factors we have $\mathrm{Factors}(v) \subseteq \mathrm{Sub}(u_1, \ldots, u_k) \cup \mathrm{C}_{\mathrm{spe}}$. We can conclude from this:

- $\mathrm{Sub}((v)\downarrow) \subseteq \{(v)\downarrow\} \cup \mathrm{Sub}(u_1, \ldots, u_k) \cup \mathrm{C}_{\mathrm{spe}}$
- If $\mathrm{Sign}(v) \neq \mathrm{Sign}((v)\downarrow)$ we have $(v)\downarrow \in \mathrm{C}_{\mathrm{spe}} \cup \mathrm{Sub}(u_1, \ldots, u_k) \subseteq \mathrm{Sub}(E) \cup \mathrm{C}_{\mathrm{spe}}$ by Lemma 3.

The lemma then follows directly from $\mathrm{Sub}(E, (v)\downarrow) \cup \mathrm{C}_{\mathrm{spe}} \neq \mathrm{Sub}(E) \cup \mathrm{C}_{\mathrm{spe}}$. □

Lemma 9 *Let D be a derivation and $l \rightarrow r \in \mathrm{L}^{u,\mathrm{g}}$ a rule applied in D. Then for each $s \in l$ if there is a rule $l_s \rightarrow s \in \mathrm{L}^{v,\mathrm{g}}$ applied in D we can assume $\mathrm{Sign}(u) \neq \mathrm{Sign}(v)$.*

Proof. By contradiction. Let $\mathcal{D}$ be the set of derivations for which the lemma does not hold and let $D \in \mathcal{D}$ with a minimal number of rules application that does not satisfy the lemma. Let $l \rightarrow r \in \mathrm{L}^{u,\mathrm{g}}$ be the first of these rules. For any term $s \in l$ such that there exists a rule $l_s \rightarrow s \in \mathrm{L}^{v,\mathrm{g}}$ in D we do the following construction:

Since u and v are in the same theory wlog we can assume $u, v \in \langle S_1 \rangle$ and let $v' = v[x_1 \leftarrow u]$. By definition of $\langle S_1 \rangle$ we have $v' \in \langle S_1 \rangle$. In $\mathrm{L}^{v',\mathrm{g}}$ there is a rule

$$l_i, l_j \setminus \{t_i\} \rightarrow t_j$$

By iterating this construction on l we build a rule matching the criterion. This contradicts the minimality of D in $\mathcal{D}$. Thus $\mathcal{D}$ is empty. □

The following lemma is a direct consequence of Lemma 8 that will be used throughout this paper.

Lemma 10 *Let $D : E_0 \rightarrow \ldots \rightarrow E_n$ be a derivation where the E_i are normalized for $i \in \{1, \ldots, n\}$ and assume there exists $s \in \mathrm{Sub}(E_i) \setminus (\mathrm{Sub}(E_0) \cup \mathrm{C}_{\mathrm{spe}})$. Then there exists in D a step $E_{j-1} \rightarrow_{l_s \rightarrow s} E_j$ with $j \leq i$ and $l_s \rightarrow s \in \mathrm{L}^{u,\mathrm{g}}$ with $\mathrm{Sign}(u) = \mathrm{Sign}(s)$.*

PROOF. Consider the minimal indice j such that $s \in \mathrm{Sub}(E_j)$. By hypothesis we have $j > 0$ and $j \leq i$. Moreover by minimality of j we have $\mathrm{Sub}(E_j) \neq \mathrm{Sub}(E_{j-1})$. Since $s \notin \mathrm{C_{spe}}$ Lemma 8 implies that $E_j = E_{j-1}, s$, and that if $E_{j-1} \rightarrow_{l_s \rightarrow s=} E_j$ with $l_s \rightarrow s \in \mathrm{L}^{u,\mathrm{g}}$ then $\mathrm{Sign}(u) = \mathrm{Sign}(s)$. □

Lemma 11 *Let $l \rightarrow r$ be a rule in $\mathrm{L}^{u,\mathrm{g}}$ and $s \in l$ with $\mathrm{Sign}(s) \neq \mathrm{Sign}(u)$ and $s \neq r$. Then $(l\delta_s)\downarrow \rightarrow (r\delta_s)\downarrow$ is also a rule in $\mathrm{L}^{u,\mathrm{g}}$.*

PROOF. Consider $t = u(l)$. Its factors are in normal form, $s, \mathrm{c_{min}}$ are alien to t and s is a factor of t and $s \neq (t)\downarrow$. Therefore by Lemma 4 we have $(t\delta_s)\downarrow = (u((l\delta_s)\downarrow))\downarrow$. □

3.1.4 Well-formed derivations

A derivation $E_0 \rightarrow_{\mathcal{U}} E_0, t_1 \rightarrow_{\mathcal{U}} \cdots \rightarrow_{\mathcal{U}} E_n$ of intruder system $\mathcal{U}$ is *well-formed* if for all $i \in \{1, \ldots, n\}$ we have $t_i \in \mathrm{Sub}(E_0, t_n) \cup \mathrm{C_{spe}}$; in other words every message generated by an intermediate step either occurs in the goal, in the initial set of messages or is a special constant. In next lemma we assume E and t are in normal form and that all terms produced during a derivation are in normal form.

Lemma 12 *A derivation of minimal length starting from E of goal t is well-formed.*

PROOF. Let n be the length of D, and let $\{t_1, \ldots, t_n\}$ be such that $t = t_n$ and

$$D : E \rightarrow E, t_1 \rightarrow E, t_1, t_2 \rightarrow \ldots \rightarrow E, t_1, \ldots, t_n$$

By minimality of D it is without stutter. By contradiction assume that i is the maximal indice such that $t_i \notin \mathrm{Sub}(E_0, t_n) \cup \mathrm{C_{spe}}$. Since $t_i \notin \mathrm{Sub}(E_0, t_n) \cup \mathrm{C_{spe}}$ we have $i < n$.

By Lemma 10 we have $t_i \notin \mathrm{Sub}(E_0) \cup \mathrm{C_{spe}}$ implies $t_i \notin \mathrm{Sub}(E_{i-1}) \cup \mathrm{C_{spe}}$. Let $l_i \rightarrow t_i \in \mathrm{L}^{u,\mathrm{g}}$ such that $E_{i-1} \rightarrow E_i$. By Lemma 8 this implies $\mathrm{Sign}(u) = \mathrm{Sign}(t_i)$.

By the minimality of D the term t_i has to be used in the left-hand side of a subsequent step in the derivation (otherwise the step producing t_i can be avoided). Let us introduce the non-empty set Ω of step indices i where t_i has to be used in left-hand side. More precisely

$$\Omega = \{h \mid E_{h-1} \rightarrow E_h \in D \text{ and } E_{h-1} \setminus \{t_i\} \not\rightarrow E_h \setminus \{t_i\}\}$$

Let j be the minimum element of Ω and let $l_j \rightarrow t_j \in \mathrm{L}^{v,\mathrm{g}}$ be the j-th deduction rule in derivation D. Note that $j > i$ and thus by maximality of i we have $t_j \in \mathrm{Sub}(E_0, t_n)$. Therefore $t_i \notin \mathrm{Sub}(t_j)$. Moreover $j \in \Omega$ implies $t_i \in l_j$. By Lemma 9 we can assume u and v are not in the same theory.

Since $\mathrm{Sign}(u) = \mathrm{Sign}(t_i)$ we have $\mathrm{Sign}(t_i) \neq \mathrm{Sign}(v)$. Thus by Lemma 11 there exists a rule $(l_j\delta_{t_i})\downarrow \rightarrow (t_j\delta_{t_i})\downarrow$ in $\mathrm{L}^{v,\mathrm{g}}$. By minimality of j the only term $r \in E_{j-1}$ such that $t_i \in \mathrm{Sub}(r)$ is t_i itself and thus $(l_j\delta_{t_i})\downarrow = l_j, \mathrm{c_{min}} \setminus \{t_i\}$. Since $t_i \notin \mathrm{Sub}(t_j)$ we have $(t_j\delta_{t_i})\downarrow = t_j$. Thus there exists in $\mathrm{L}^{v,\mathrm{g}}$ a rule $l'_j \rightarrow t_j$ with $l'_j \subseteq E_{j-1}$ and $t_i \notin L'_j$. This contradicts again $j \in \Omega$.

Thus Ω is empty and the lemma follows. □

3.2 Protocol analysis

In this subsection we describe how protocol are modelled. In the following we only model a single session of the protocol since it is well-known how to reduce several sessions to this case. Our semantics follows the one by [17].

In Dolev-Yao's model the intruder can intercept, block and/or redirect all messages sent by honest agents. It is also able to send message by masquerading its identity and honest agents may know its identity and wrongly assume it is honest and communicate with it on that basis.

Thus it has complete control over the communication medium. We model this by considering the intruder *is* the network. Messages sent by honest agents are sent directly to the intruder and messages received by the honest agents are always sent by the intruder. From the intruder's point of vue a finite execution of a protocol is therefore the interleaving of a finite sequence of messages it has to send and a finite sequence of messages it receives (and add to its knowledge).

We also assume the interaction of the intruder with one agent to be an atomic step. The intruder sends a message m to an honest agent, this agent tests the validity of this message and respond to it. Alternatively an agent may initiate an execution and in this case we assume it reacts to a dummy message sent by the intruder.

A *step* is a triplet $(\text{RECV}(x); \text{SEND}(s); \text{COND}(e))$ where $x \in \mathcal{X}$, $s \in \mathrm{T}(\mathcal{G}, \mathcal{X})$ and e is a set of equations between terms of $\mathrm{T}(\mathcal{G}, \mathcal{X})$. The meaning of a step is that upon receiving message x, the honest agent checks the equations in e and sends the message s. An *execution* of a protocol is a finite sequence of steps.

Example 3 *Consider the following simple protocol:*

$$\begin{aligned} &A \to B : \{M \oplus B\}K \\ &B \to A : B \\ &A \to B : K \\ &B \to A : M \end{aligned}$$

Assuming the algebraic properties of $\oplus$*, symmetric encryption* $\mathrm{se}(,)$ *and symmetric decryption* $\mathrm{sd}(,)$ *we model this protocol as:*

$$\begin{aligned} &\text{RECV}(v_1); \text{SEND}(\mathrm{se}(M \oplus B, K)); \text{COND}(v_1 = \mathrm{c_{min}}) \\ &\text{RECV}(v_2); \text{SEND}(B); \text{COND}(\emptyset) \\ &\text{RECV}(v_3); \text{SEND}(K); \text{COND}(v_3 = B) \\ &\text{RECV}(v_4); \text{SEND}(\mathrm{sd}(v_2, v_4) \oplus B); \text{COND}(v_4 = K) \\ &\text{RECV}(v_5); \text{SEND}(\mathrm{c_{min}}); \text{COND}(v_5 = M) \end{aligned}$$

Note that in our setting we can model that at some step i the message must match the pattern t_i by adding an equation $v_i \stackrel{?}{=} t_i$ to $\mathcal{S}$.

In order to define whether an execution of a protocol is feasible we must first define when a substitution σ satisfies a set of equations $\mathcal{S}$.

Definition 5 *(Unification systems) Let $\mathcal{H}$ be a set of equational axioms on $\mathrm{T}(\mathcal{G},\mathcal{X})$. An $\mathcal{H}$-*Unification system *$\mathcal{S}$ is a finite set of couples of terms in $\mathrm{T}(\mathcal{G},\mathcal{X})$ denoted by $(u_i \stackrel{?}{=} v_i)_{i\in\{1,\ldots,n\}}$. It is satisfied by a ground substitution σ, and we note $\sigma \models \mathcal{S}$, if for all $i \in \{1,\ldots,n\}$ $u_i\sigma =_{\mathcal{H}} v_i\sigma$.*

Let $\mathcal{I}=\langle\mathcal{G},S,\mathcal{H}\rangle$ be an intruder system. A *configuration* is a couple $\langle P,N\rangle$ where P is a finite sequence of steps and N is a set of ground terms (the knowledge of the intruder). From the configuration $\langle(\textsc{recv}(x);\textsc{send}(s);\textsc{cond}(e))\cdot P,N\rangle$ a transition to (P',N') is possible iff there exists a ground substitution σ such that $x\sigma \in \overline{N}^{\mathcal{I}}$, $\sigma \models e$, $N' = N \cup \{s\sigma\}$ and $P' = P\sigma$. Trace based-security properties like secrecy can be reduced to the following *Execution feasability* problem.

Execution feasability
Input: an initial configuration $\langle P,N_0\rangle$
Output: SAT iff there exists a reachable configuration $\langle\emptyset,M\rangle$

Protocol insecurity. A major security problem is to decide whether the intruder can deduce a secret m from a finite sequence of message exchange P. This problem can be reduced to the feasability of execution of the protocol by appending a last step

$$S = \textsc{recv}(x);\textsc{send}(\mathrm{c}_{\min});\textsc{cond}(x=m)$$

to P. The problem of insecurity with respect to secrecy of m after P is equivalent to the feasability of the execution of $P\cdot S$. We now reduce the problem of execution feasability to the resolution of constraint systems.

3.3 Constraints systems

We now model an execution of a protocol by a constraint problem $\mathcal{C}$.

Definition 6 *(Constraints systems) Let $\mathcal{I} = \langle\mathcal{G},S,\mathcal{H}\rangle$ be an intruder system. An $\mathcal{I}$-*Constraint system *$\mathcal{C}$ is denoted: $((E_i \rhd v_i)_{i\in\{1,\ldots,n\}},\mathcal{S})$ and it is defined by a sequence of couples $(E_i,v_i)_{i\in\{1,\ldots,n\}}$ with $v_i \in \mathcal{X}$ and $E_i \subseteq \mathrm{T}(\mathcal{G})$ for $i \in \{1,\ldots,n\}$ and $E_{i-1} \subseteq E_i$ for $i \in \{2,\ldots,n\}$ and by an $\mathcal{H}$-unification system $\mathcal{S}$.*

*An $\mathcal{I}$-*Constraint *system $\mathcal{C}$ is satisfied by a ground substitution σ if for all $i \in \{1,\ldots,n\}$ we have $v_i\sigma \in \overline{E_i\sigma}$ and if $\sigma \models_{\mathcal{H}} \mathcal{S}$. If a ground substitution σ satisfies a constraint system $\mathcal{C}$ we denote it by $\sigma \models_{\mathcal{I}} \mathcal{C}$.*

Constraint systems are denoted by $\mathcal{C}$ and decorations thereof. Note that if a substitution σ is a solution of a constraint system $\mathcal{C}$, by definition of constraints and of unification systems the substitution $(\sigma)\downarrow_{\mathcal{O}}$ is also a solution of $\mathcal{C}$. In the context of cryptographic protocols the inclusion $E_{i-1} \subseteq E_i$ means that the knowledge of an intruder does not decrease as the protocol progresses: after receiving a message an honest agent will respond to it. This response can be added to the knowledge of an intruder who listens all communications.

Example 4 *We model the protocol of Example 3 by the following constraint system. First we gather all conditions in an unification system* $\mathcal{S}$

$$\mathcal{S} = \left\{ v_1 \stackrel{?}{=} \mathrm{c_{min}} \ , \ v_3 \stackrel{?}{=} B \ , \ v_4 \stackrel{?}{=} K \ , \ v_5 \stackrel{?}{=} M \right\}$$

The protocol execution for intruder $\mathcal{I}$ *with initial knowledge* $\{\mathrm{c_{min}}\}$ *is then expressed by the constraint:*

$$\begin{array}{rl} \mathcal{C} = ((& \mathrm{c_{min}} \rhd v_1, \\ & \mathrm{c_{min}}, \mathrm{se}(M \oplus B, K) \rhd v_2, \\ & \mathrm{c_{min}}, \mathrm{se}(M \oplus B, K), B \rhd v_3, \\ & \mathrm{c_{min}}, \mathrm{se}(M \oplus B, K), B, K \rhd v_4), \\ & \mathrm{c_{min}}, \mathrm{se}(M \oplus B, K), B, K, \mathrm{sd}(v_2, v_4) \oplus B \rhd v_5, \mathcal{S}) \end{array}$$

We are not interested in general constraint systems but only in those related to protocols. In particular we need to express that a message to be sent at some step i should be built from previously received messages recorded in the variables $v_j, j < i$, and from the initial knowledge. To this end we define:

Definition 7 *(Deterministic Constraints Systems) We say that an* $\mathcal{I}$*-constraint system* $((E_i \rhd v_i)_{i \in \{1,\ldots,n\}}, \mathcal{S})$ *is* deterministic *if* $\forall i \in \{1, \ldots, n\}$ *we have* $\mathrm{Var}(E_i) \subseteq \{v_1, \ldots, v_{i-1}\}$

The decision problems we are interested in are the *satisfiability* and the *ordered satisfiability* of intruder constraint systems.

Satisfiability

Input: an $\mathcal{I}$-constraint system $\mathcal{C}$

Output: SAT iff there exists a substitution σ such that: $\sigma \models_{\mathcal{I}} \mathcal{C}$.

In order to be able to combine solutions of constraints in component theories to get a solution for the full theory these solutions have to satisfy some ordering constraints too. Intuitively, this is to avoid introducing cycle when building a global solution. This motivates the following definition:

Ordered Satisfiability

Input: a constraint system $\mathcal{C}$, X the set of all variables and C the set of all free constants occurring in $\mathcal{C}$ and a linear ordering $\prec$ on $X \cup C$.

Output: SAT iff there exists a substitution σ such that:

$$\begin{cases} \sigma \models_{\mathcal{I}} \mathcal{C} \\ \forall x \in X \text{ and } \forall c \in C, x \prec c \text{ implies } c \notin \mathrm{Sub_{syn}}(x\sigma) \end{cases}$$

The main result of this paper is a modularity result that can be stated as follows:

Theorem 1 *If the ordered satisfiability problem is decidable for two intruders* $\langle \mathcal{F}_1, S_1, \mathcal{E}_1 \rangle$ *and* $\langle \mathcal{F}_2, S_2, \mathcal{E}_2 \rangle$ *for disjoint signatures* $\mathcal{F}_1$ *and* $\mathcal{F}_2$ *then the satisfiability problem is decidable for the intruder* $\langle \mathcal{F}_1 \cup \mathcal{F}_2, \langle S_1 \rangle \cup \langle S_2 \rangle, \mathcal{E}_1 \cup \mathcal{E}_2 \rangle$.

This result is obtained as a consequence of Algorithm 1 solving $\mathcal{U}$-constraints using algorithms for solving *ordered satisfiability* for intruders $\langle \mathcal{F}_1, S_1, \mathcal{E}_1 \rangle$ and $\langle \mathcal{F}_2, S_2, \mathcal{E}_2 \rangle$ given in Section 5. The proof of the soundness and completeness of this algorithm relies on the results of next section. We prove in Section 6 that it suffices to be able to solve deterministic constraint systems in component theories.

4 Bound solutions of constraints systems

We recall that the intruder system $\mathcal{U}$ is the union of the two intruder systems $\langle \mathcal{F}_1, S_1, \mathcal{E}_1 \rangle$ and $\langle \mathcal{F}_2, S_2, \mathcal{E}_2 \rangle$. In this section we let $\mathcal{C} = ((E_i \rhd v_i)_{i \in \{1,\ldots,n\}}, \mathcal{S})$ be a deterministic constraint problem on signature $\mathcal{F}$ and σ be a normal substitution that satisfies $\mathcal{C}$. We assume $\mathrm{c}_{\min} \in E_1$ and $\mathrm{C}_{\mathrm{spe}} \subseteq \mathrm{Sub}(\mathcal{C})$. We are going to show in this section that a solution can be built uniquely from subterms occuring in $\mathcal{C}$.

From now we say a ground term is *bound* if it is bound by σ in $\mathrm{Sub}(\mathcal{C})$, unless otherwise specified. In the same way a term is *free* if it is free in $\mathrm{Sub}(\mathcal{C})$. We say a substitution σ is bound if for all variables x in its support all terms in $\mathrm{Sub}(x\sigma)$ are bound by σ in $\mathrm{Sub}(\mathcal{C})$.

In the rest of this section we first prove in Subsection 4.1 that if s is a free subterm of σ then replacing it by $\mathrm{c}_{\min}$ in derivations yields new derivations (*i.e.* no deduction power is lost.) Then we prove in that if $\mathcal{C}$ is satisfiable there exists a solution σ of $\mathcal{C}$ which is bound (Subsection 4.2).

4.1 Stability of derivations by replacement of free subterms

First we prove that when replacing a free term s in σ by the constant $\mathrm{c}_{\min}$ we still obtain a derivation. The proof of the following lemma relies on the hypothesis above that the constraint system $\mathcal{C}$ is deterministic and σ is normal.

Lemma 13 *Let $s \notin \mathrm{C}_{\mathrm{spe}}$ be a term such that $s \in \mathrm{Sub}((E_k\sigma)\downarrow)$ for some $1 \leq k \leq n$. Then either there exists $i < k$ such that $s \in \mathrm{Sub}(v_i\sigma)$ or there exists $m \in \mathrm{Sub}(E_k)$ such that $(m\sigma)\downarrow = s$ and in that case either $\mathrm{Sign}(s) = \mathrm{Sign}(m)$ or m is a constant.*

Proof. Since the constraint system is deterministic we have

$$\mathrm{Sub}((E_k\sigma)\downarrow) \subseteq (\mathrm{Sub}(E_k)\sigma)\downarrow \cup \mathrm{Sub}(v_1\sigma, \ldots, v_{k-1}\sigma) \cup \mathrm{C}_{\mathrm{spe}}$$

Assume that there is no $i < k$ such that $s \in \mathrm{Sub}(v_i\sigma)$. Then $s \notin \mathrm{C}_{\mathrm{spe}}$ implies there exists $m \in \mathrm{Sub}(E_k)$ such that $(m\sigma)\downarrow = s$. By $s \notin \mathrm{C}_{\mathrm{spe}}$ and Lemma 6 there exists $u \in \mathrm{Sub}(m)$ such that $(u\sigma)\downarrow = s$ and $\mathrm{Sign}(u) = \mathrm{Sign}(s)$. □

The next lemma is a one-step version of Lemma 15.

Lemma 14 *Let G be a finite set of normalized terms with $\mathrm{c}_{\min} \in G$, let r and s be two normalized terms with $s \notin \mathrm{C}_{\mathrm{spe}}$, let l_r be the rule $r_1, \ldots, r_n \to r \in \mathrm{L}^{u,\mathrm{g}}$, l_s be the rule $s_1, \ldots, s_m \to s \in \mathrm{L}^{v,\mathrm{g}}$. Assume moreover $\mathrm{Sign}(v) = \mathrm{Sign}(s)$ and:*

$$G \to_{l_s} G, s \to_{l_r} G, r, s$$

Then either $(r\delta_s)\downarrow \in (G\delta_s)\downarrow$ *or:*

$$(G\delta_s)\downarrow \to_{\mathcal{I}} (G\delta_s)\downarrow, (r\delta_s)\downarrow$$

PROOF. Assume $(r\delta_s)\downarrow \notin (G\delta_s)\downarrow$ and thus $s \neq r$. By Lemma 9 we can safely assume $\mathrm{Sign}(u) \neq \mathrm{Sign}(v)$. Thus the assumption $\mathrm{Sign}(s) = \mathrm{Sign}(u)$ implies $\mathrm{Sign}(s) \neq \mathrm{Sign}(v)$. The result is then a trivial consequence of Lemma 11. □

Lemma 15 will be applied with s a free term in a solution σ in Lemma 16. It permits us to characterise minimal solutions of a constraint satisfaction problem.

Lemma 15 *Let E and F be finite sets of normalized terms with $\mathrm{c}_{\min} \in E$. Let s, t be two normalized terms not in $\mathrm{C}_{\mathrm{spe}}$ with $s \in \overline{E} \setminus \mathrm{Sub}(E)$ and $t \in \overline{E \cup F}$. We have:*

$$(t\delta_s)\downarrow \in \overline{((E \cup F)\delta_s)\downarrow}$$

PROOF. First let us note that if $s \notin \mathrm{Sub}(E, F, t)$ then one has $t = t\delta_s$, $E = E\delta_s$ and $F = F\delta_s$. Since E, F and t are normalized the result is trivial.

Following the hypothesis $s \notin \mathrm{Sub}(E)$ we now assume $s \in \mathrm{Sub}(F, t)$. By Lemma 12 there exists a well-formed derivation D_s without stutter starting from E of goal s:

$$E = E_0 \to_{\mathcal{I}} E_1 \to_{\mathcal{I}} \cdot \to_{\mathcal{I}} E_n$$

with $s \in E_n$. Since $s \notin \mathrm{Sub}(E) \cup \mathrm{C}_{\mathrm{spe}}$ Lemma 8 and the fact that D_s is without stutter imply that the last rule is a rule $l_s \to {}_{\mathrm{L}^{u,\mathrm{g}}} s$ with $s \notin \mathrm{Sub}(E_{n-1})$ and $\mathrm{Sign}(u) = \mathrm{Sign}(s)$. Since the derivation is well-formed we have $l_s \subseteq \mathrm{Sub}(E, s)$. Let $H = E_{n-1}$. We have $E \subseteq H$ and thus $t \in \overline{H \cup F \cup \{s\}}$.

Let D_t be a well-formed derivation without stutter starting from $H \cup F \cup \{s\}$ of goal t and let D be the concatenation of the two sequences of rules in D_s and D_t, possibly removing unnecessary rules (those creating a term already present). Then D defines a derivation from $E \cup F$ of goal t. This derivation is without stutter by construction and well-formed since D_s and D_t are well-formed and $s \in \mathrm{Sub}(F, t)$. Let $(G_i)_{i \in \{1,\ldots,k\}}$ be the sequence of sets appearing in this derivation. Let us prove that when allowing stutters (*i.e.* $(G_i\delta_s)\downarrow = (G_{i+1}\delta_s)\downarrow$) the sequence $(G_0\delta_s)\downarrow \to \cdots \to (G_k\delta_s)\downarrow$ is also a derivation.

By contradiction let $G \to_{l_r \to r} G'$ be the first transition in D such that $(G\delta_s)\downarrow \not\to_{\mathrm{L}} (G'\delta_s)\downarrow$ and $(G'\delta_s)\downarrow \neq (G\delta_s)\downarrow$. The rule has not been applied in D_s since otherwise either:

- $s \in \mathrm{Sub}(l_r, r)$ implies $r = s$ and thus $(G\delta_s)\downarrow = (G'\delta_s)\downarrow$
- $s \notin \mathrm{Sub}(l_r, r)$ implies one can apply the same rule on $(G\delta_s)\downarrow$ and this would lead to $(G'\delta_s)\downarrow$.

Therefore the rule $l \to r$ has been applied in D_t. Consider now the sequence:

$$G \setminus s \to_{l_s \to s} G \to_{l_r \to r} G, r = G'$$

Since $l_s \to s \in \mathrm{L}^{u,\mathrm{g}}$ with $\mathrm{Sign}(u) = \mathrm{Sign}(s)$ Lemma 14 implies that either $(G'\delta_s)\downarrow = (G\delta_s)\downarrow$ or that $(G\delta_s)\downarrow \to (G'\delta_s)\downarrow$ is a valid transition, thus contradicting the choice of G and G'. □

4.2 Existence and properties of bound solutions

We now prove that if $\mathcal{C}$ is satisfiable then it is satisfied by a bound substitution. First we prove it is possible to replace one free term s by the minimal constant.

Lemma 16 *If there exists* $x \in \mathrm{Var}(\mathcal{C})$ *and* $s \in \mathrm{Sub}(x\sigma)$ *such that* s *is free in* $\mathrm{Sub}(\mathcal{C})$ *for* σ *then* $(\sigma\delta_s)\downarrow \models \mathcal{C}$

PROOF. Let $\sigma' = (\sigma\delta_s)\downarrow$. Note that s free implies $s \notin \mathrm{C_{spe}}$.

First let us prove that $\sigma' \models \mathcal{S}$. Since s is free in $\mathrm{Sub}(\mathcal{C})$ Lemma 5 implies that for all equations $s \stackrel{?}{=} t$ in $\mathcal{S}$ we have $(s\sigma)\downarrow = (t\sigma)\downarrow$ implies $(s\sigma')\downarrow = (t\sigma')\downarrow$.

Let us now prove that for all $i \in \{1, \ldots, n\}$ if there is a derivation starting from $(E_i\sigma)\downarrow$ of goal $v_i\sigma$ then there is a derivation starting from $(E_i\sigma')\downarrow$ of goal $v_i\sigma'$. Let $j \in \{1, \ldots, n\}$ and consider the set:

$$\Omega_s = \{i \mid s \in \mathrm{Sub}((E_i\sigma)\downarrow, v_i\sigma)\}$$

If $j \notin \Omega_s$ we have $(E_j\sigma)\downarrow\delta_s = (E_j\sigma)\downarrow$ and $v_j\sigma = v_j\sigma\delta_s$. Since s is free Lemma 5 implies $(E_j\sigma')\downarrow = (E_j\sigma)\downarrow$ and $v_j\sigma' = v_j\sigma$. Thus by assumption there exists a derivation starting from $(E_j\sigma')\downarrow$ of goal $v_j\sigma'$.

Thus if $\Omega = \emptyset$ the Lemma is valid. Otherwise $\Omega \neq \emptyset$ and we can consider the minimum index i_0 in Ω. By minimality of i_0 and by Lemma 13 we have $s \notin \mathrm{Sub}((E_{i_0}\sigma)\downarrow)$ and thus $s \in \mathrm{Sub}(v_{i_0}\sigma)$. By Lemma 10 this implies $s \in \overline{(E_{i_0}\sigma)\downarrow}$.

For $j \in \Omega$ let $F_j = (E_j\sigma)\downarrow \setminus (E_{i_0}\sigma)\downarrow$. By $(E_j\sigma)\downarrow = (E_{i_0}\sigma)\downarrow \cup F_j$ and $s \in \overline{(E_{i_0}\sigma)\downarrow} \setminus \mathrm{Sub}((E_{i_0}\sigma)\downarrow)$ we can apply Lemma 15 to obtain a derivation D'_j starting from $((E_j\sigma)\downarrow\delta_s)\downarrow$ of goal $v_j\sigma'$. Since s is free Lemma 5 implies D'_j is a derivation starting from $(E_j\sigma')\downarrow$ of goal $v_j\sigma'$.

Thus for all $j \in \{1, \ldots, n\}$ there is a derivation starting from $(E_i\sigma')\downarrow$ of goal $v_i\sigma'$ □

The proof of next Proposition 1 is a direct consequence of Lemma 16 and exploits the well-foundedness of the order $<$ to prove it is possible to iteratively replace all free subterms.

Proposition 1 *Let* $\mathcal{C}$ *be a satisfiable constraint system. There exists a normal bound substitution* σ *such that* $\sigma \models \mathcal{C}$.

PROOF. Consider the set Σ of normal substitutions that satisfy $\mathcal{C}$. By hypothesis Σ is not empty. Let σ be a minimal substitution in Σ for the total ordering $<$ on ground terms extended on substitutions seen as multisets of ground terms. Let us prove σ is bound to $\mathcal{C}$.

By contradiction assume there exists s free in $\mathrm{Sub}(\sigma)$ and let $\sigma' = (\sigma\delta_s)\downarrow$. By Lemma 16 we also have $\sigma' \models \mathcal{C}$. By monotony of $<$ we have $\sigma' < \sigma$. By definition of R we have $(\sigma')\downarrow \leq \sigma'$. Thus $(\sigma')\downarrow \in \Sigma$ and $(\sigma')\downarrow < \sigma$ which contradicts the minimality of σ. □

Note that the notion of subterm used throughout this paper implies that for some theories (such as abelian groups) we can have an infinite number of bound substitutions. It is thus not possible to use Proposition 1 to directly guess a substitution satisfying a constraint system $\mathcal{C}$. However this notion is sufficient to permit us to combine decision procedures. In next lemma we prove that instanciating the constraint $\mathcal{C}$ by a bound sunstitution does not introduce any new subterm.

Lemma 17

$$\mathrm{Sub}((\mathrm{Sub}(\mathcal{C})\sigma)\downarrow) = (\mathrm{Sub}(\mathcal{C})\sigma)\downarrow$$

PROOF. Let $S = (\mathrm{Sub}(\mathcal{C})\sigma)\downarrow$. We have $S \subseteq \mathrm{Sub}(S)$. The converse inclusion $\mathrm{Sub}(S) \subseteq S$ follows directly from:

$$\mathrm{Sub}((\mathrm{Sub}(\mathcal{C})\sigma)\downarrow) \subseteq (\mathrm{Sub}(\mathcal{C})\sigma)\downarrow \cup \mathrm{Sub}(\mathrm{Var}(\mathcal{C})\sigma) \cup \mathrm{C}_{\mathrm{spe}}$$

Since σ is bound we have $\mathrm{Sub}(\mathrm{Var}(\mathcal{C})\sigma) \subseteq (\mathrm{Sub}(\mathcal{C})\sigma)\downarrow$ and by hypothesis we have $\mathrm{C}_{\mathrm{spe}} \subseteq \mathrm{Sub}(\mathcal{C})$. □

5 Combination of decision procedures

We introduce Algorithm 1 for solving satisfiability of constraint systems for the union $\mathcal{U}$ of two intruders systems $\mathcal{I}_1 = \langle \mathcal{F}_1, S_1, \mathcal{E}_1 \rangle$ and $\mathcal{I}_2 = \langle \mathcal{F}_2, S_2, \mathcal{E}_2 \rangle$ with disjoint signatures $\mathcal{F}_1$ and $\mathcal{F}_2$. We explain this algorithm in Subsection 5.1, then we prove its soundness (Subsection 5.2) and completeness (Subsection 5.3). Finally we partially prove Theorem 1 in Subsection 5.4.

5.1 Combination algorithm

First let us explain Algorithm 1:

Step 2 The algorithm input is a $\mathcal{U}$-Constraint system $(\mathcal{D}, \mathcal{S})$. An equational system $\mathcal{S}$ is *homogeneous* if for all $u \stackrel{?}{=} v \in \mathcal{S}$, u and v are both pure 1-terms or both pure 2-terms. It is well-known that equational systems can be transformed into equivalent (w.r.t. satisfiability) homonogeneous systems. Thus we can assume that $\mathcal{S}$ is homogeneous without loss of generality.

Step 3 abstracts every subterm t of $\mathcal{C}$ by a new variable $\psi(t)$. A choice of ψ such that $\psi(t) = \psi(t')$ will lead to solutions that identify t and t'.

Steps 4-6 assign non-deterministically a signature to the root symbol of the subterms of $\mathcal{C}$ instanciated by a solution. The choice $th(\psi(t)) = 0$ corresponds to the situation where t gets equal to a free constant.

Steps 7-10 choose and order non-deterministically the intermediate subterms in derivations that witness that the solution satisfies the constraints in $\mathcal{D}$.

Step 11 defines a constraint problem $\mathcal{C}'$ collecting the previous choices on subterms identification, subterms signatures and derivation structures.

Step 12 splits the problem $\mathcal{S}'$ in two pure subproblems.

Step 13 splits non-deterministically the problem $\mathcal{D}'$, that is we select for each $E \rhd v$ in $\mathcal{D}'$ an intruder system to solve it.

Step 14 guesses an ordering on variables: this ordering will preclude the value of a variable from being a subterm of the value of a smaller variable. This is used to avoid cycles in the construction of the solution.

Step 15 solves independantly the 2 pure subproblems obtained at steps 12-13. In $\mathcal{C}_i$ the variables q with $th(q) \neq i$ will be considered as constants.

Algorithm 1 Combination Algorithm

1: **Solve**$_{\mathcal{U}}(\mathcal{C})$
2: **Let** $\mathcal{C} = ((E_i \rhd v_i)_{i\in\{1,\ldots,n\}}, \mathcal{S})$ with $\mathcal{S}$ homogeneous.
3: **Choose** ψ an application from $\mathrm{Sub}(C)$ to $\mathcal{X} \setminus \mathrm{Var}(\mathcal{C})$ and let $Q = \psi(\mathrm{Sub}(C))$
4: **for all** $q \in Q$ **do**
5: Choose a theory $th(q) \in \{0, 1, 2\}$
6: **end for**
7: **for** $i = 1$ to n **do**
8: Choose $Q_i \subseteq Q$
9: Choose a linear ordering over the elements of Q_i say $(q_{i,1}, \ldots, q_{i,k_i})$
10: **end for**
11: **Let** $\mathcal{C}' = (\mathcal{D}', \mathcal{S}')$ where

$$\left\{ \begin{array}{ll} \mathcal{S}' = & \mathcal{S} \cup \left\{ z \stackrel{?}{=} \psi(z) \,|\, z \in \mathrm{Sub}(C) \right\} \\ \mathcal{D}' = & \Delta_1, \ldots, \Delta_i, \ldots \Delta_n \end{array} \right.$$

and $\Delta_i = (K_i, Q_i^{<j} \rhd q_{i,j})_{j\in\{1,\ldots,k_i\}}$, $(K_i, Q_i \rhd \psi(v_i))$ with

$$\left\{ \begin{array}{rcl} K_i & = & \psi(E_i) \cup \bigcup_{j=1}^{i-1} Q_j \\ Q_i^{<j} & = & q_{i,1}, q_{i,2}, \ldots, q_{i,j-1} \end{array} \right.$$

12: **Split** $\mathcal{S}'$ into $\mathcal{S}_1, \mathcal{S}_2$ such that $\mathcal{S}' = \mathcal{S}_1 \cup \mathcal{S}_2$ and:

$$\left\{ \begin{array}{l} \mathcal{S}_1 = \left\{ z \stackrel{?}{=} z' \in \mathcal{S}' \,|\, z, z' \text{are pure 1-terms} \right\} \\ \mathcal{S}_2 = \left\{ z \stackrel{?}{=} z' \in \mathcal{S}' \,|\, z, z' \text{are pure 2-terms} \right\} \end{array} \right.$$

13: **Split** non-deterministically $\mathcal{D}'$ into $\mathcal{D}_1, \mathcal{D}_2$
14: **Choose** a linear ordering $\prec$ over Q.
15: **Solve** $\mathcal{C}_i = (\mathcal{D}_i, \mathcal{S}_i)$ for intruder $\mathcal{I}_i$ with linear ordering $\prec$ for $i \in \{1, 2\}$
16: **if** both are satisfied **then**
17: **Output:** SATISFIED
18: **end if**

As in previous section we assume, with the notations of the algorithm, $\mathrm{C_{spe}} \subseteq \mathrm{Sub}(\mathcal{C})$ and $\mathrm{c_{min}} \in E_1$. Recall that we say a normal substitution σ is *bound* if for all variables x with $x\sigma \neq x$ and for all $t \in \mathrm{Sub}(x\sigma)$ there exists $u \in \mathrm{Sub}(\mathcal{C})$ such that $(u\sigma)\downarrow = t$.

We are now ready to show that Algorithm 1 is sound and complete.

5.2 Correctness

The soundness of Algorithm 1 is a consequence of the two results below. Together they show that if the algorithm produces satisfiable $\mathcal{I}_i$-constraints $(i = 0, 1)$ at *Step*15 then the input $\mathcal{U}$-constraint is satisfiable. The notations below refer to the ones in Algorithm 1.

Lemma 18 *If there exists $\mathcal{C}'$ satisfiable at Step 11 then $\mathcal{C}$ is satisfiable.*

Proof. Assume there exists $\mathcal{C}'$ chosen from $\mathcal{C}$ as in the algorithm and a substitution σ such that $\sigma \models_{\mathcal{U}} \mathcal{C}'$. We can check easily that for all $i \in \{1, \ldots, n\}$:

1. for all elements t of $Q_i\sigma$ there exists a derivation D_t from $\psi(E_i)\sigma$ of goal t;
2. there exists a derivation D_{v_i} starting from $\psi(E_i)\sigma \cup Q_i\sigma$ of goal $\psi(v_i)\sigma$;
3. the concatenation of all former derivations yields a derivation starting in $\psi(E_i)\sigma$ of goal $\psi(v_i)\sigma$.

Since σ is solution of $\mathcal{S}'$ and $\sigma \models \mathcal{S}$, for all $t, t' \in \mathrm{Sub}(\mathcal{C})$ we have $(t\sigma)\downarrow = (t'\sigma)\downarrow$ if $\psi(t) = \psi(t')$. Hence there are derivations starting from $(E_i\sigma)\downarrow$ of goal $v_i\sigma$ for all $i \in \{1, ..n\}$, respectively. Therefore $\sigma \models_{\mathcal{U}} \mathcal{C}$. □

Now we prove that the combination part is sound. This proof follows the lines of the soundness proof for the combination of unification algorithms by [4].

Proposition 2 *Assume that at Step 15 for $i = 1, 2$, σ_i is a solution of $\mathcal{C}_i = (\mathcal{D}_i, \mathcal{S}_i)$ for intruder $\mathcal{I}_i$. Then we can build a solution σ of $\mathcal{C}'$ for intruder $\mathcal{U}$ at Step 11.*

Proof. We can assume up to renaming that σ_i maps every variable of $\mathcal{C}_i$ to a term that contains new variables (away from $\mathcal{C}$) or variables x with $th(x) \neq i$ that are considered as free constants in $\mathcal{C}_i$. Let us define σ by induction on $\prec$. Let x be the least variable for $\prec$ and $i \in \{0, 1, 2\}$ such that $th(x) = i$. We define $x\sigma = x\sigma_i$. Assume now that all $y\sigma$ for $y \prec x$ have been defined and that $th(x) = i$. If $i = 0$ then $x\sigma = x$. Else, since σ_i satisfies the linear order restriction, the variables $y_1, \ldots, y_m$ of index $j \neq i$ that occurs in $x\sigma_i$ (and considered as free constants in $\mathcal{C}_i$) have to be smaller that x with respect to $\prec$. Hence by induction σ is already defined on $y_1, \ldots, y_m$ and we can take $x\sigma = x\sigma_i\sigma$ (considering now the y_i as variables). We can show as in [4] that $\sigma \models \mathcal{S}'$.

Now let us consider $E \triangleright v$ an element of $\mathcal{D}_i$. Since σ_i is a solution of $\mathcal{C}_i$ we have a derivation $E\sigma_i \rightarrow^*_{\mathcal{I}_i} v\sigma_i$. By replacing in this derivation all the variables z such that $th(z) \neq i$ (that were considered as free constants in $\mathcal{C}_i$) by $z\sigma$ we get a derivation $E\sigma \rightarrow^*_{\mathcal{U}} v\sigma$. This reasoning applies to all constraints in $\mathcal{D}_1 \cup \mathcal{D}_2$ and therefore σ is a solution of $\mathcal{D}'$ and finally of $\mathcal{C}'$ too. □

5.3 Completeness

Proposition 3 *If $\mathcal{C}$ satisfiable then there exists $\mathcal{C}_1$ and $\mathcal{C}_2$ satisfiable at Step 15 of the algorithm.*

PROOF. First let us prove that the 11 first steps of the algorithm preserve satisfiability. Assume $\mathcal{C}$ is satisfiable. By Proposition 1 there exists a normal bound substitution σ which satisfies $\mathcal{C}$. Define ψ to be a function such that $\psi(t) = \psi(t')$ if and only if $(t\sigma)\downarrow = (t'\sigma)\downarrow$. Thus by Lemma 17 there exists a bijection ϕ from Q to $\mathrm{Sub}((\mathrm{Sub}(C)\sigma)\downarrow)$. We let $th(q) = i$ if $\mathrm{Sign}(\phi(q)) = \mathcal{F}_i$ and $th(q) = 0$ if $\mathrm{Sign}(\phi(q)) = \perp$. Note that by the construction of $\mathcal{S}'$ and the choice of ψ we can extend σ on Q by $q\sigma = (\psi^{-1}(q)\sigma)\downarrow$.

For each $i \in \{1, \ldots, n\}$ by Lemma 12 we can consider a well-formed derivation D_i starting from $F_i = (E_i\sigma)\downarrow$ and of goal $g_i = v_i\sigma$:

$$D_i : F_i \rightarrow_{\mathcal{U}} F_i, r_{i,1} \rightarrow_{\mathcal{U}} \cdots \rightarrow_{\mathcal{U}} F_i, r_{i,1}, \ldots, r_{i,k_i} \rightarrow_{\mathcal{U}} F_i, r_{i,1}, \ldots, r_{i,k_i}, g_i$$

We have $\mathrm{Sub}(F_i, g_i) \subseteq \mathrm{Sub}((\mathrm{Sub}(\mathcal{C}\sigma))\downarrow)$. Since the derivation is well-formed we have $\{r_{i,1}, \ldots, r_{i,k_i}\} \subseteq \mathrm{Sub}(F_i, g_i)$. By Proposition 1, $\mathrm{Sub}((\mathrm{Sub}(\mathcal{C}\sigma))\downarrow) = (\mathrm{Sub}(\mathcal{C}\sigma))\downarrow$ Thus the function ϕ^{-1} is defined for each $r_{i,j}$. Let $q_{i,j} = \phi^{-1}(r_{i,j})$ and Q_i be the sequence of the $q_{i,j}$.

The algorithm will non-deterministically produce a $\mathcal{C}'$ corresponding to these choices and satisfied by σ (extended over Q by $q\sigma = \tilde{q}$) by construction.

Since $\mathcal{S}$ is satisfiable, following the lines of F. Baader and K. Schulz [4] permits to prove that $\mathcal{S}_1$ and $\mathcal{S}_2$ are satisfiable with the linear constant restriction $\prec$ chosen such that $q \prec q'$ implies $q'\sigma$ is not a subterm of $q\sigma$.

We choose the sequence of constraints in $\mathcal{D}_1$ (resp. $\mathcal{D}_2$) to be the subsequence of constraints $F \rhd q$ from $\mathcal{D}'$ such that the corresponding transition in the solution was performed by a rule in $\mathrm{L}^{u,\mathrm{g}}$ with $\mathrm{Sign}(u) = \mathcal{F}_1$ (resp. $\mathcal{F}_2$). By construction these two systems are satisfiable. □

5.4 Combining solutions of subsystems

We can now prove the main theorem of this article. It is stated for the combination of two intruders but can easily generalized to n intruders over disjoint signatures $\mathcal{F}_1, \ldots, \mathcal{F}_n$.

Theorem 1 *If the ordered satisfiability problem is decidable for two intruders $\langle \mathcal{F}_1, S_1, \mathcal{E}_1 \rangle$ and $\langle \mathcal{F}_2, S_2, \mathcal{E}_2 \rangle$ for deterministic constraint problems over disjoint signatures $\mathcal{F}_1$ and $\mathcal{F}_2$ then the satisfiability problem is decidable for deterministic constraint problems for the intruder $\langle \mathcal{F}_1 \cup \mathcal{F}_2, \langle S_1 \rangle \cup \langle S_2 \rangle, \mathcal{E}_1 \cup \mathcal{E}_2 \rangle$.*

PROOF. Propositions 3 and 2 imply that if the ordered satisfiability problem are decidable for two intruders $\mathcal{I}_1$ and $\mathcal{I}_2$ over disjoint signature and for arbitrary constraint problems then the satisfiability problem is decidable for the union of these two intruders and for *deterministic* constraint problems.

In Section 6 we prove that it is sufficient to assume that ordered satisfiability problems are decidable for each intruder for *deterministic* constraint problems to derive the decidability for their union. □

6 Deriving deterministic constraint systems

We now prove it suffices to solve *deterministic* constraint systems with ordering constraints for intruders $\langle \mathcal{F}_1, S_1, \mathcal{E}_1 \rangle$ and $\langle \mathcal{F}_2, S_2, \mathcal{E}_2 \rangle$ in order to be able to solve deterministic constraint systems for their union $\mathcal{U}$.

6.1 Structure of bound solutions

These results will be used in Proposition 4.

Let $\mathcal{C} = ((E_i \rhd v_i)_{i\in\{1,\ldots,n\}}, \mathcal{S})$ be a deterministic constraint problem on signature $\mathcal{F}$ satisfiable for intruder system $\mathcal{U}$. By Proposition 1 there exists a bound substitution σ such that $\sigma \models_{\mathcal{U}} \mathcal{C}$. As usual we assume $\mathrm{c}_{\min} \in E_1$ and $\mathrm{C}_{\mathrm{spe}} \subseteq \mathrm{Sub}(\mathcal{C})$. We also define:

$$\begin{cases} Z_i &= \mathrm{Sub}((\mathrm{Sub}(E_i)\sigma)\!\downarrow) \cup \mathrm{Sub}(v_i\sigma) \\ Z &= \cup_{i=1}^{n} Z_i \end{cases}$$

Intuitively Z is the set of terms that play a role in the derivations. We say a term m is a *prefix* of a term t if $(m\sigma)\!\downarrow = t$ and $\mathrm{Sign}(m) = \mathrm{Sign}(t)$.

For the purification of a constraint system we will need to order the terms in $(\mathrm{Sub}(\mathcal{C})\sigma)\!\downarrow$ according to the indice of the (instanciated) constraint where they occur for the first time. Given a term $t \in Z$ we introduce its indice $\mathrm{ind}(t)$ which informally marks the first time t appears.

Definition 8 *For all $t \in Z$ we define* $\mathrm{ind}(t)$ *to be the first i such $t \in \mathrm{Sub}((\mathrm{Sub}(E_i)\sigma)\!\downarrow, v_i\sigma)$. If $t \in (\mathrm{Sub}(\mathcal{C})\sigma)\!\downarrow \setminus Z$ we define* $\mathrm{ind}(t) = n + 1$.

Given a term $t \in Z$ we say that t is *past-bound* if there exists a prefix of t in $\mathrm{Sub}(E_i)$. We say a term t is *past-free* if it is not past-bound. Note that $\mathrm{c}_{\min}$ is past-bound of indice 1 by hypothesis. First let us reformulate Lemma 6 with the indice for terms not in $\mathrm{C}_{\mathrm{spe}}$.

Lemma 19 *Let $u \in \mathrm{Sub}(E_j)$ and $s \notin \mathrm{C}_{\mathrm{spe}}$ be in $\mathrm{Sub}((u\sigma)\!\downarrow)$. Then either there exists a prefix of s in $\mathrm{Sub}(u)$ or $\mathrm{ind}(s) < j$.*

Proof. By Lemma 6 and $s \notin \mathrm{C}_{\mathrm{spe}}$ either there exists a prefix of s in $\mathrm{Sub}(u)$ or there exists $x \in \mathrm{Var}(u)$ with $s \in \mathrm{Sub}(x\sigma)$. In the latter case, since the protocol is deterministic we have $x \in \{v_1, \ldots, v_{j-1}\}$ and therefore $\mathrm{ind}(s) \leq j - 1$. □

A direct consequence of this lemma is that if s is past-free of indice $i \leq n$ there is no $w \in \mathrm{Sub}(E_i)$ with $(w\sigma)\!\downarrow = s$. Since the constraint system is deterministic this implies in turn that there is no $w \in \mathrm{Sub}(E_i)$ with $s \in \mathrm{Sub}((w\sigma)\!\downarrow)$. Next Lemma is also a restatement of Lemma 10 with the notions of past-free and past-bound terms.

Lemma 20 *Let $t \notin \mathrm{C}_{\mathrm{spe}}$ be past-free of indice $i \leq n$ and $D_i : F_1 \rightarrow^* F_k$ be a derivation starting from $(E_i\sigma)\!\downarrow$ of goal $v_i\sigma$.*
Then there exists $j \in \{2, \ldots, k\}$ such that $F_{j-1} \rightarrow_{\mathrm{L}^{u,g}} F_j$ with $\mathrm{Sign}(u) = \mathrm{Sign}(t)$ and for all $j' < j$ we have $t \notin \mathrm{Sub}(F_{j'})$.

PROOF. Let t be past-free of indice i. By Lemma 19 there exists no $u \in \mathrm{Sub}(E_i)$ with $t \in \mathrm{Sub}((u\sigma)\downarrow)$. Thus t past-free of indice i implies $t \in \mathrm{Sub}(v_i\sigma) \setminus \mathrm{Sub}((\mathrm{Sub}(E_i)\sigma)\downarrow)$ and thus $t \in \mathrm{Sub}(v_i\sigma)$ and $t \notin \mathrm{Sub}((E_i\sigma)\downarrow)$. Thus the lemma is a direct consequence of Lemma 10. □

Remark. Let $t \in \mathrm{Sub}(\mathcal{C})$ such that $\mathrm{ind}((t\sigma)\downarrow) \leq n$. Let us anticipate on the choices during the split into two constraint problems in Algorithm 3. On the one hand if $(t\sigma)\downarrow$ is past-free then Lemma 20 and σ solution imply that the first time the class of t will appear as a subterm of a constraint it will appears on the right-hand side of a constraint (as a $q_{i,j}$) and this constraint will have to solved in $\mathcal{C}_i$ with $\mathrm{Sign}(t) = \mathcal{F}_i$. On the other hand if t is past-bound we will show in Lemma 21 how t can be partially infered from the subterms of $\mathcal{C}$. Algorithm 2 permits to compute this partial inference once the first choices of Algorithm 3 have been made.

Given a term $t \in Z$ of indice $i \in n$ next lemma states it is possible to compute a special prefix of t.

Lemma 21 *Let $t \in Z \setminus \mathrm{C}_{\mathrm{spe}}$ be a past-bound term of indice $i \leq n$. One can compute from $\mathrm{Sub}(\mathcal{C})$ a prefix m of t such that $\mathrm{Var}(m) \subseteq \{v_1, \ldots, v_{i-1}\}$ and for each $u \in \mathrm{Factors}(m)$ either $(u\sigma)\downarrow$ is past-free and of indice $< i$ or is a prefix of $(u\sigma)\downarrow$.*

PROOF. By contradiction let i be minimal such that there exists a past-bound term of indice i for which the lemma does not hold. Let $m_t \in \mathrm{Sub}(E_i)$ be minimal for the subterm relation such that the lemma does not hold for $t = (m_t\sigma)\downarrow$. By minimality of m_t and $\mathrm{ind}(t) = i$ Lemma 19 implies m_t is a prefix of t. Let Ω_t be the set of prefix of t computable from $\mathcal{C}$ (but not necessarily in $\mathrm{Sub}(\mathcal{C})$). Since the constraint problem is deterministic $m_t \in \Omega_t$ and thus $\Omega_t \neq \emptyset$.

Given $m \in \Omega_t$ let $\mu(m)$ the number of factors u of m such that $(u\sigma)\downarrow$ is past-bound and $\mathrm{Sign}(u) \neq \mathrm{Sign}((u\sigma)\downarrow)$. Let $m \in \Omega_t$ be such that $\mu(m)$ is minimal.

Claim 1 $\mu(m) = 0$.

PROOF OF THE CLAIM. By contradiction assume $\mu(m) > 0$. This imlplies there exists $u \in \mathrm{Factors}(m)$ such that $\mathrm{Sign}(u) \neq \mathrm{Sign}((u\sigma)\downarrow)$. Let j be minimal such that $u \in \mathrm{Sub}(E_j)$ and let $s = (u\sigma)\downarrow$. By definition we have $j \leq i$ and $\mathrm{ind}(s) \leq j$. By Lemma 19 there exists a prefix w of s in $\mathrm{Sub}(E_{\mathrm{ind}(s)})$. Moreover in case $\mathrm{ind}(s) = j$ there is one such prefix in $\mathrm{Sub}(u)$. In both cases by minimality of i and of m_t we can conclude that one can compute a term m_s verifying the lemma for s. Note that either s is a constant (and thus $s = m_s$) or $\mathrm{Sign}(m_s) = \mathrm{Sign}(m_t)$. In both cases m cannot be a factor of m_s. Let m' be the term m where the factor u is replaced by m_s. We have $\mu(m') = \mu(m) - 1 + \mu(m_s) = \mu(m) - 1$ and therefore $\mu(m') < \mu(m)$. This contradicts $\mu(m)$ minimal and non-zero. ♢

As a consequence of Claim 1 $u \in \mathrm{Factors}(m)$ and $\mathrm{Sign}(u) \neq \mathrm{Sign}((u\sigma)\downarrow)$ imply $(u\sigma)\downarrow$ is past-free. Let $s = (u\sigma)\downarrow$. By Lemma 19 either there exists a prefix w of s in $\mathrm{Sub}(u)$ or $s \in \mathrm{Sub}(x\sigma)$ for some $x \in \mathrm{Var}(u)$. In the first case s is past-free imply by definition that $\mathrm{ind}(s) < i$. In the second case we have $x \in \{v_1, \ldots, v_{i-1}\}$ and therefore $\mathrm{ind}(s) < i$. □

Functions defined. A substitution σ defines a mapping f_σ from $\mathrm{Sub}(\mathcal{C})$ to $(\mathrm{Sub}(\mathcal{C}\sigma))\!\downarrow$. Let Q be a finite set of variables away from $\mathcal{C}$ of size $|(\mathrm{Sub}(\mathcal{C}\sigma))\!\downarrow|$. There is a bijection $g_{\sigma,Q}$ from $(\mathrm{Sub}(\mathcal{C}\sigma))\!\downarrow$ to Q. The functions f_σ and $g_{\sigma,Q}$ define a mapping ψ from $\mathrm{Sub}(\mathcal{C})$ to Q such that $\psi(t_1) = \psi(t_2)$ iff $(t_1\sigma)\!\downarrow = (t_2\sigma)\!\downarrow$. We now define some functions on Q. In this context Q_Z is the subset of variables $q \in Q$ such that $g^{-1}_{\alpha,Q}(q) \in Z$.

First we define the function $\mathrm{ind}(\cdot)$on Q with

$$\mathrm{ind}(q) = \begin{cases} \mathrm{ind}(g^{-1}_{\alpha,Q}(q)) & \text{If } q \in Z \\ n+1 & \text{Otherwise} \end{cases}$$

Note that the actual value $(u\sigma)\!\downarrow$ is not needed to compute $\mathrm{ind}(\psi(u))$ for $u \in \mathrm{Sub}(C)$. It suffices to know (or guess) ψ and the subterm relation on $(\mathrm{Sub}(\mathcal{C}\sigma))\!\downarrow$ for the case where $(u\sigma)\!\downarrow$ is a past-free strict subterm of $v_i\sigma$.

We now define a function $\mathrm{type}(\cdot)$ that associates to a variable q of Q a mark depending on whether $g^{-1}_{\alpha,Q}(q)$ is past-free or past-bound. Note that this function may again be computed as soon as the subterm relation on $(\mathrm{Sub}(\mathcal{C}\sigma))\!\downarrow$ is known.

$$\mathrm{type}(q) = \begin{cases} \text{past-bound} & \text{If } q \in Z \text{ and } g^{-1}_{\alpha,Q}(q) \text{ past-bound} \\ \text{past-free} & \text{Otherwise} \end{cases}$$

Note that a function $\varphi_{\mathrm{p}}(\cdot)$ that associates to a variable $q \in Q$ a term $\varphi_{\mathrm{p}}(q)$ satisfying the Lemma 21 may be computed as soon as ψ and $\mathrm{type}(\cdot)$ or known. There exists several possible choices for $\varphi_{\mathrm{p}}(\cdot)$ depending on the order of computation but they all are valid. The Algorithm 2 permits to compute $\varphi_{\mathrm{p}}(\cdot)$ for all terms in Q_Z.

Algorithm 2 Algorithm to compute $\varphi_{\mathrm{p}}(\cdot)$

for all $q \in Q$ with $\mathrm{type}(q) =$ past-free **do**
  $\varphi_{\mathrm{p}}(q) = q$
end for
while there exists $q \in Q$ with $\varphi_{\mathrm{p}}(q)$ undefined **do**
  Let $m \in \mathrm{Sub}(\mathcal{C})$ with $\varphi_{\mathrm{p}}(\psi(m))$ undefined
  if $\varphi_{\mathrm{p}}(\cdot)$defined on all factors of m **then**
    Let $\varphi_{\mathrm{p}}(\psi(m))$ be m where all factors u have been replaced by $\varphi_{\mathrm{p}}(\psi(u))$
  end if
end while

In Algorithm 2 the condition at Step 6 is always satisfied for constants of $\mathrm{Sub}(\mathcal{C})$.

6.2 Revised combination algorithm

Our aim is to prove that at Step 15 of Algorithm 1 it is sufficient to try to solve the $\mathcal{C}_i$ only in the case they are deterministic. In order to do this we restrict the choices in order to demonstrate why this is possible. We also limit the guessing part by adding a line where

the subterm relation $\mathcal{R}$ on $(\mathrm{Sub}(C\sigma))\downarrow$ is guessed. This gives the restricted combination Algorithm 3.

Algorithm 3 Combination Algorithm for CP satisfiability

Solve$_{\mathcal{U}}(\mathcal{C})$
Let $\mathcal{C} = ((E_i \rhd v_i)_{i\in\{1,\ldots,n\}}, \mathcal{S})$ with $\mathcal{S}$ homogeneous.
Choose ψ an application from $\mathrm{Sub}(\mathcal{C})$ to $\mathcal{X} \setminus \mathrm{Var}(\mathcal{C})$ and let $Q = \psi(\mathcal{C})$.
Choose a partial ordering $\mathcal{R}$ over Q^2
for all $q \in Q$ **do**
  Choose a theory $th(q) \in \{0, 1, 2\}$
  Compute the indice $\mathrm{ind}(q) \in \{1, \ldots, n, n+1\}$
  Compute $\mathrm{type}(q) \in \{\text{past-free}, \text{past-bound}\}$
  Compute $\varphi_{\mathrm{p}}(q)$
end for
for $i = 1$ to n **do**
  Choose $Q_i \subseteq Q$
  Choose a linear ordering over the elements of Q_i say $(q_{i,1}, \ldots, q_{i,k_i})$
end for
Let $\mathcal{C}' = (\mathcal{D}', \mathcal{S}')$ where

$$\left\{ \begin{array}{rcl} \mathcal{S}' = & \mathcal{S} \cup \left\{ z \stackrel{?}{=} \psi(z) \,|\, z \in \mathrm{Sub}(C) \right\} \\ \mathcal{D}' = & \Delta_1, \ldots, \Delta_i, \ldots \Delta_n \end{array} \right.$$

and $\Delta_i = (K_i, Q_i^{<j} \rhd q_{i,j})_{j\in\{1,\ldots,k_i\}}$, $(K_i, \varphi_{\mathrm{p}}(Q_i) \rhd \psi(v_i))$ with

$$\left\{ \begin{array}{rcl} K_i = & \varphi_{\mathrm{p}}(\psi(E_i)) \cup \bigcup_{j=1}^{i-1} \varphi_{\mathrm{p}}(Q_j) \\ Q_i^{<j} = & \varphi_{\mathrm{p}}(q_{i,1}), \varphi_{\mathrm{p}}(q_{i,2}), \ldots, \varphi_{\mathrm{p}}(q_{i,j-1}) \end{array} \right.$$

Split $\mathcal{S}'$ into $\mathcal{S}_1, \mathcal{S}_2$ such that $\mathcal{S}' = \mathcal{S}_1 \cup \mathcal{S}_2$ and:

$$\left\{ \begin{array}{l} \mathcal{S}_1 = \left\{ z \stackrel{?}{=} z' \in \mathcal{S}' \,|\, z, z' \text{are pure 1-terms} \right\} \\ \mathcal{S}_2 = \left\{ z \stackrel{?}{=} z' \in \mathcal{S}' \,|\, z, z' \text{are pure 2-terms} \right\} \end{array} \right.$$

Split $\mathcal{D}'$ into $\mathcal{D}_1, \mathcal{D}_2$
Choose a linear ordering $\prec$ over Q.
Solve $\mathcal{C}_i = (Eq_i, \mathcal{S}_i)$ over $\mathcal{F}_i$ with linear restriction $\prec$ for $i \in \{1, 2\}$
if both are deterministic and satisfied **then**
  Output: SATISFIED
end if

We are now ready to show that Algorithm 3 is complete. The correctness follows from Proposition 2 as the added features only restrict the possible choices in the algorithm.

6.3 Completeness

Proposition 4 *If $\mathcal{C}$ is satisfiable then Algorithm 3 will generate 2 deterministic and satisfiable constraint systems $\mathcal{C}_1$ and $\mathcal{C}_2$ for $\mathcal{I}_1$ and $\mathcal{I}_2$ at Step 20.*

Proof. Assume $\mathcal{C}$ is satisfiable. By Proposition 1 it has a normal bound solution σ.

Let $K = (\mathrm{Sub}(\mathcal{C}\sigma))\downarrow$. There exists a surjective function ϕ_1 from $\mathrm{Sub}(C)$ to K such that $\phi(t) = (t\sigma)\downarrow$. Moreover there exists an injection ϕ_2 between K and $\mathcal{X} \setminus \mathrm{Var}(\mathcal{C})$. We denote $\tilde{q} = \phi_2^{-1}(q)$. Consider the following choices:

- we choose $\psi = \phi_2 \circ \phi_1$, and let $Q = \psi(\mathrm{Sub}(\mathcal{C}))$;
- we choose $\mathcal{R}$ such that for any $t, t' \in K$ we have t strict subterm of t' iff $\phi_2(t)\mathcal{R}\phi_2(t')$.
- The theory chosen for $q \in Q$ is $\mathrm{Sign}(\tilde{q})$;
- The computation of the indice and of whether $\tilde{q}$ is past-free or past-bound depends only on ψ and on the subterm relation in K. It can thus be computed once knowing $\mathcal{R}$ and ψ;
- $\varphi_{\mathrm{p}}(q)$ can be computed following Lemma 21;
- For each $i \in \{1, \ldots, n\}$ consider a well-formed derivation D_i starting from $F_i = (E_i\sigma)\downarrow$ and of goal $g_i = v_i\sigma$:

$$D_i : F_i \to F_i, r_{i,1} \to \cdots \to F_i, r_{i,1}, \ldots, r_{i,k_i} \to F_i, r_{i,1}, \ldots, r_{i,k_i}, g_i$$

  Since the derivation is well-formed we have:

$$\begin{cases} \{r_{i,1}, \ldots, r_{i,k_i}\} \subseteq \mathrm{Sub}(F_i, g_i) \\ \mathrm{Sub}(F_i, g_i) \subseteq (\mathrm{Sub}(\mathcal{C}\sigma))\downarrow \end{cases}$$

  The function ϕ_2 is defined for each $r_{i,j}$. Let $q_{i,j} = \phi_2(r_{i,j})$ and Q_i be the sequence of the $q_{i,j}$.

From now on we assume these choices and computations have been performed.

Claim 1 *$\mathcal{C}'$ is satisfiable at Step 15.*

Proof of the claim. Given the choices made it suffices to remark that by construction of $\varphi_{\mathrm{p}}(\cdot)$ we have $q\sigma = (\varphi_{\mathrm{p}}(q)\sigma)\downarrow$ (see Lemma 21).
Thus the algorithm will non-deterministically produce a $\mathcal{C}'$ corresponding to these choices and satisfied by σ (extended over Q by $q\sigma = \tilde{q}$) by construction. $\Diamond$

Since $\mathcal{S}$ is satisfiable, following the lines of F. Baader and K. Schulz [4] permits to prove that $\mathcal{S}_1$ and $\mathcal{S}_2$ are satisfiable with the linear constraint restriction $\prec$ chosen such that $q \prec q'$ implies $\tilde{q}'$ is not a subterm of $\tilde{q}$.

Splitting. We consider the split choice in which the sequence of constraints in Eq_1 (resp. Eq_2) is the subsequence of constraints $F \rhd q$ from $\mathcal{D}'$ such that the corresponding transition in the solution was performed by a rule in $\mathrm{L}^{u,\mathrm{g}}$ with $\mathrm{Sign}(u) = \mathcal{F}_1$ (resp. $\mathcal{F}_2$). By construction and since $(\varphi_{\mathrm{p}}(q)\sigma)\downarrow = q\sigma$ these two systems are satisfiable. Then the $\mathcal{C}_i$ are purified. This means that if $\varphi_{\mathrm{p}}(q)$ is in theory $\mathcal{F}_j$ with $j \neq i$ we replace it by q.

In a system $\mathcal{C}_i$, if $\varphi_{\mathrm{p}}(q) \neq q$ and $th(q) \neq i$ it can be replaced by q before solving $\mathcal{C}_i$. Then the variable symbols q such that $th(q) \neq i$ will be considered as constants when solving the system $\mathcal{C}_i$.

Let us now prove the systems $\mathcal{C}_i$ are deterministic.

Claim 2 *Let $q \in Q$ past-bound, $q' \in \mathrm{Var}(\varphi_{\mathrm{p}}(q))$ with $th(q') = th(q)$. Then $\mathrm{ind}(q') < \mathrm{ind}(q)$ and q' is past-free.*

Proof of the claim. Since q is past-bound $\varphi_{\mathrm{p}}(q)$ is defined and different from q. Let $i = \mathrm{ind}(\tilde{q}) \in \{1, \ldots, n\}$ and let $q' \in \mathrm{Var}(\varphi_{\mathrm{p}}(q))$ with $th(q') = 1$. Let m be the term chosen and $u \in \mathrm{Factors}(m)$ such that $\psi(u) = q'$. We have $\mathrm{Sign}(u) \neq \mathrm{Sign}(m)$. Thus if q' is not past-free we have $th(q') \neq th(q)$. We can conclude that q' is past-free and therefore by Lemma 21 (used on $E_{\mathrm{ind}(q)}$) $\mathrm{ind}(q') < \mathrm{ind}(q)$. $\Diamond$

Claim 3 *The constraint systems $\mathcal{C}_i = (Eq_i, \mathcal{S}_i)$ $(i = 1, 2)$ derived with the above choices are deterministic.*

Proof of the claim. First we note that q is a variable of Eq_i if $th(q) = i$. Let $Eq_i = (E'_j \rhd q_j)_j$ and define $V_j = \{q_1, \ldots, q_j\}$. By contradiction assume the set $\mathcal{J}$ of indices j such that $\mathrm{Var}(E'_j) \not\subseteq Q_{j-1}$ is not empty and let j be the minimum of $\mathcal{J}$ and let $m \in E'_j$ such that $\mathrm{Var}(m) \not\subseteq Q_{j-1}$ and finally let q such that $\varphi_{\mathrm{p}}(q) = m$.
If q is past-free then $m = q$ and thus $q \in E_j \setminus Q_{j-1}$. But the first time a past-free term t appears in a derivation it is deducted with a rule in $\mathrm{L}^{u,\mathrm{g}}$ with $\mathrm{Sign}(u) = \mathrm{Sign}(t)$. Thus by the choice during the split we have a constraint $E_{j'} \rhd q$ in Eq_1 with $j' < j$. This contradicts $q \notin Q_{j-1}$.
Else we have $m \neq q$ but by Claim 2 each $q' \in \mathrm{Var}(\varphi_{\mathrm{p}}(q))$ is past-free and $\mathrm{ind}(q') < \mathrm{ind}(q)$. Thus by the choice during the split there exists a constraint $E_{j'} \rhd q$ in Eq_1 with $j' < j$. Therefore $j \notin \mathcal{J}$ and thus $\mathcal{J} = \emptyset$, which proves the claim. $\Diamond$

□

7 Application to Security Protocols

In order to combine constraint solving algorithms for *subtheories* we only need to show that ordered satisfiability is decidable in each component theory. To illustrate the benefit of our approach we show that this is the case for several theories encoding useful properties of cryptographic primitives (pair, xor, exponential, encryption). A consequence of our main result Theorem 1 is that we can decide the security of (finite sessions of) any protocol employing these primitives even assuming their algebraic properties and **even if they are employed all together**.

7.1 Abelian group operators

We consider in this subsection the case of an intruder $\mathcal{I}_\times = \langle \mathcal{F}_\times, S_\times, \mathcal{E}_\times \rangle$, where $\mathcal{F}_\times$ is the signature $\{\mathrm{i}, \times, 1\}$, $S_\times$ is the set of terms $\{\mathrm{i}(x), x \times y, 1\}$ and with the equational theory:

$$\mathcal{E}_\times \left\{ \begin{array}{rcl} (x \times y) \times z & = & x \times (y \times z) \\ x \times y & = & y \times x \\ 1 \times x & = & x \\ x \times \mathrm{i}(x) & = & 1 \end{array} \right.$$

We reduce decidability of $\mathcal{I}_\times$-constraints to satisfiability of affine systems of equations on Z. This reduction is performed in two steps. First we prove that it suffices to consider ground sets E_i in the constraints, and thus that the v_i are linear combination of ground terms. Second the unification system is translated to a system of affine equations over Z.

Lemma 22 *One can compute $\mathcal{C}' = ((E'_i \rhd v_i)_{i \in \{1,\ldots,n\}}, \mathcal{S})$ from $\mathcal{C}$ such that*

- *$\sigma \models \mathcal{C}$ iff $\sigma \models \mathcal{C}'$*
- *for all $i \in \{1, \ldots, n\}$ the set E'_i is ground*

This leads to next proposition.

Proposition 5 *The ordered satisfiability problem for deterministic constraints and intruder $\mathcal{I}_\times$ is decidable in NPTIME.*

7.2 XOR operator

We consider the intruder $\mathcal{I}_\oplus = \langle \mathcal{F}_\oplus, \{x \oplus y, 0\}, \mathcal{E}_\oplus \rangle$ with the signature $\mathcal{F}_\oplus = \{0, \cdot \oplus \cdot\}$ and with equational theory:

$$\mathcal{E}_\oplus \left\{ \begin{array}{rcl} (x \oplus y) \oplus z & = & x \oplus (y \oplus z) \\ x \oplus y & = & y \oplus x \\ 0 \oplus x & = & x \\ x \oplus x & = & 0 \end{array} \right.$$

Let $\mathcal{C} = ((E_i \rhd v_i)_{i \in \{1,\ldots,n\}}, \mathcal{S})$ be a deterministic constraint problem for $\mathcal{I}_\oplus$. Lemma 22 can be adapted to this case. The main difference is that affine systems are over $(Z/2Z)$. We refer to [20] for a more detailed description of the translation from unification problems to linear systems and of the resolution of such systems.

Proposition 6 *The ordered satisfiability problem for deterministic constraints and intruder $\mathcal{I}_\oplus$ is decidable in PTIME.*

7.3 Exponential operator

For simplicity of exposition we assume that all exponentiations are computed in the same modulus. We consider the signature $\mathcal{F}_{\exp} = \{\exp(\cdot,\cdot), \mathrm{i}(\cdot), \cdot \times \cdot\}$ and the following equational theory over $\mathcal{F}_{\exp}$ to take into account the properties of the exponential:

$$\mathcal{E}_{\exp} \left\{ \begin{array}{rcl} \exp(x,1) & = & x \\ \exp(\exp(x,y),z) & = & \exp(x, y \times z) \\ (x \times y) \times z & = & x \times (y \times z) \\ x \times y & = & y \times x \\ 1 \times x & = & x \\ x \times \mathrm{i}(x) & = & 1 \end{array} \right.$$

The deduction system of the intruder is modelled by $S_{\exp} = \{x \times y, \mathrm{i}(x), \exp(x,y)\}$. We can now define $\mathcal{I}_{\exp} = \langle \mathcal{F}_{\exp}, S_{\exp}, \mathcal{E}_{\exp} \rangle$. Following [26] the satisfiability problem for deterministic constraint systems for this intruder can be reduced to the satisfiability problem for an abelian group operator.

Proposition 7 *The ordered satisfiability problem for deterministic constraints and intruder $\mathcal{I}_{\exp}$ is decidable in NPTIME.*

7.4 Equational Dolev-Yao theory with explicit decryption

We consider now the Dolev-Yao intruder $\mathcal{I}_{DY} = \langle \mathcal{F}_{DY}, S_{DY}, \mathcal{E}_{DY} \rangle$ over the signature $\mathcal{F}_{DY} = \{\langle \cdot,\cdot \rangle, \pi_1(\cdot), \pi_2(\cdot), \mathrm{se}(\cdot,\cdot), \mathrm{sd}(\cdot,\cdot)\}$ with deduction system defined by $S_{DY} = \{\langle x,y \rangle, \pi_1(x), \pi_2(x), \mathrm{se}(x,y), \mathrm{sd}(x,y)\}$ and the equational theory:

$$\mathcal{E}_{DY} \left\{ \begin{array}{rcl} \pi_1(\langle x,y \rangle) & = & x \\ \pi_2(\langle x,y \rangle) & = & y \\ \mathrm{sd}(\mathrm{se}(x,y),y) & = & x \end{array} \right.$$

First we note that we can get from $\mathcal{E}_{DY}$ a convergent and finite rewrite system R_{DY} simply by orienting the axioms from left to right. Thanks to Theorem 8.5. of Schmidt-Schauss [28] satisfiability of equational systems modulo $\mathcal{E}_{DY}$ is decidable even in presence of linear constant restrictions. The idea is that the so-called *narrowing* procedure modulo R_{DY} terminates (since rules right-hand sides are variables) and is complete for solving equations modulo $\mathcal{E}_{DY}$ with linear constant restrictions.

The algorithm of [3] for deciding intruder $\mathcal{I}_{DY}$-constraints can be adapted to generate a finite and complete set of symbolic solutions. Then we can use the constant elimination technique of [28] to solve the ordered satisfiability problem: we apply *narrowing* (i.e. instanciating and rewriting) to the complete set of symbolic solutions provided by [3] and then we eliminate the resulting substitutions that do not satisfy the constant restrictions.

8 Conclusion

We have proposed an algorithm for combining decision procedures for intruder constraints on disjoint signatures. This algorithm allows for a modular treatment of algebraic operators in protocol analysis and a better understanding of complexity issues in the domain. Since only constraint satisfiability is required from the intruder subtheories the approach should permit one to handle more complex operators.

References

[1] M. Abadi, B. Blanchet, and C. Fournet. Just Fast Keying in the Pi Calculus. In David Schmidt, editor, *Proceedings of ESOP'04*, volume 2986 of *Lecture Notes on Computer Science*, pages 340–354, Barcelona, Spain, 2004. Springer Verlag.

[2] M. Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In *POPL '01: Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 104–115. ACM Press, 2001.

[3] R. Amadio, D. Lugiez, and V. Vanackère. On the symbolic reduction of processes with cryptographic functions. *Theor. Comput. Sci.*, 290(1):695–740, 2003.

[4] F. Baader and K. U. Schulz. Unification in the union of disjoint equational theories. combining decision procedures. *J. Symb. Comput.*, 21(2):211–243, 1996.

[5] D. Basin, S. Mödersheim, and L. Viganò. An On-The-Fly Model-Checker for Security Protocol Analysis. In Einar Snekkenes and Dieter Gollmann, editors, *Proceedings of ESORICS'03*, LNCS 2808, pages 253–270. Springer-Verlag, 2003.

[6] M. Boreale. Symbolic trace analysis of cryptographic protocols. In *Proceedings of the 28th ICALP'01*, LNCS 2076, pages 667–681. Springer-Verlag, Berlin, 2001.

[7] M. Boreale and M. Buscemi. Symbolic analysis of crypto-protocols based on modular exponentiation. In *Proceedings of MFCS 2003*, volume 2747 of *Lecture Notes in Computer Science*. Springer, 2003.

[8] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: the insecurity of 802.11. In *Proceedings of MOBICOM 2001*, pages 180–189, 2001.

[9] Y. Chevalier, R. Kuesters, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR. In *Proceedings of the Logic In Computer Science Conference, LICS'03*, June 2003.

[10] Y. Chevalier and L. Vigneron. A Tool for Lazy Verification of Security Protocols. In *Proceedings of the Automated Software Engineering Conference (ASE'01)*. IEEE Computer Society Press, 2001.

[11] J. Clark and J. Jacob. A survey of authentication protocol literature: Version 1.0. Available via `http://www.cs.york.ac.uk/~jac/papers/drareview.ps.gz`, 1997.

[12] H. Comon-Lundh. Intruder theories (ongoing work). In Igor Walukiewicz, editor, *7th International Conference, FOSSACS 2004*, volume 2987 of *Lecture Notes on Computer Science*, pages 1–4, Barcelona, Spain, March 2004. Springer Verlag.

[13] H. Comon-Lundh and V. Shmatikov. Intruder Deductions, Constraint Solving and Insecurity Decision in Presence of Exclusive or. In *Proceedings of the Logic In Computer Science Conference, LICS'03*, pages 271–280, 2003.

[14] H. Comon-Lundh and R. Treinen. Easy intruder deductions. In *Verification: Theory and Practice*, volume 2772 of *Lecture Notes in Computer Science*, pages 225–242, 2003.

[15] Ricardo Corin and Sandro Etalle. An improved constraint-based system for the verification of security protocols. In *SAS*, LNCS, pages 326–341. Springer-Verlag, 2002.

[16] V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. Technical Report LSV-04-15, Laboratoire Spécification et Vérification, ENS Cachan, France, September 2004. 36 pages.

[17] S. Delaune and F. Jacquemard. A decision procedure for the verification of security protocols with explicit destructors. In *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS'04)*, pages 278–287, Washington, D.C., USA, October 2004. ACM Press.

[18] N. Dershowitz and J-P. Jouannaud. Rewrite systems. In *Handbook of Theoretical Computer Science, Volume B*, pages 243–320. Elsevier, 1990.

[19] D. Dolev and A. Yao. On the Security of Public-Key Protocols. *IEEE Transactions on Information Theory*, 2(29), 1983.

[20] G. Guo, P. Narendran, and D. A. Wolfram. Unification and matching modulo nilpotence. *Information and Computation*, 162((1-2)):3–23, 2000.

[21] R. Küsters and T. Wilke. Automata-based Analysis of Recursive Cryptographic Protocols. In *21st Symposium on Theoretical Aspects of Computer Science (STACS 2004)*, Lecture Notes in Computer Science, pages 382–393. Springer-Verlag, 2004.

[22] C. Meadows and P. Narendran. A unification algorithm for the group Diffie-Hellman protocol. In *Workshop on Issues in the Theory of Security (in conjunction with POPL'02), Portland, Oregon, USA, January 14-15*, 2002.

[23] Catherine Meadows. The NRL protocol analyzer: an overview. *Journal of Logic Programming*, 26(2):113–131, 1996.

[24] J. Millen. On the freedom of decryption. *Information Processing Letters*, 86(6):329–333, 2003.

[25] J. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *ACM Conference on Computer and Communications Security*, pages 166–175, 2001.

[26] J. Millen and V. Shmatikov. Symbolic protocol analysis with an abelian group operator or Diffie-Hellman exponentiation. *Journal of Computer Security*, 2005.

[27] M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proc.14th IEEE Computer Security Foundations Workshop*, Cape Breton, Nova Scotia, June 2001.

[28] M. Schmidt-Schauß. Unification in a combination of arbitrary disjoint equational theories. *J. Symb. Comput.*, 8(1/2):51–99, 1989.

[29] V. Shmatikov. Decidable analysis of cryptographic protocols with products and modular exponentiation. In *Proceedings of ESOP'04*, volume 2986 of *Lecture Notes in Computer Science*, pages 355–369,. Springer-Verlag, 2004.

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399