Combining Intruder Theories [*]

Yannick Chevalier , Michaël Rusinowitch

[1] IRIT Université Paul Sabatier, France
email: ychevali@irit.fr
[2] LORIA-INRIA-Lorraine, France
email: rusi@loria.fr

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 paper 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 mentioned operators with a bounded number of sessions.

1 Introduction

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 [2, 5, 18]. Among the different approaches the symbolic ones [16, 10, 4] are based on reducing the problem to constraint solving in a term algebra. This reduction has proved to be quite effective on standard benchmarks and also was able to discover new flaws on several protocols [4].

However while most formal analysis of security protocols abstracts from low-level properties, i.e., certain algebraic properties of encryption, such as the multiplicative properties of RSA or the properties induced by chaining methods for block ciphers, many real attacks and protocol weaknesses rely on these properties. For attacks exploiting the *XOR* properties in the context of mobile communications see [7]. 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 $(g^y)^z = (g^z)^y$.

In this paper 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 [19, 3]. 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 independent theories. For instance it allows one to combine the exponentiation with abelian group theory of [17] with the Xor theory of [8].

Related works. Recently several protocol decision procedures have been designed for handling algebraic properties in the Dolev-Yao model [15, 6, 11, 8]. 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 [12] 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.

2   Motivation

Combination of 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 $\{M\}K_a$ denotes the encryption by the public key K_a of A.

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

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 $(g_a)^a \mod p_a$. The private key of A is a. Denoting $\exp_p$ the exponentiation modulo p, and with new nonces k_1, k_2 and k_3 we can rewrite the protocol as:

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

In this simple example 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 properties 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. However none of these works handles the analysis of 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.

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* describing how the intruder can construct new messages from the ones she already knows and a set of *equational laws* $\mathcal{E}$ verified by the functions employed in messages. We give here two examples of intruder theories. Some other theories are given in [9].

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 $i(\cdot)$ and the neutral element is denoted 1.

$$
L_\times \left\{ \begin{array}{r} \to 1 \\ x \to i(x) \\ x, y \to x \times y \end{array} \right. \qquad \mathcal{E}_\times \left\{ \begin{array}{c} (x \times y) \times z = x \times (y \times z) \\ x \times y = y \times x \\ 1 \times x = x \\ x \times i(x) = 1 \end{array} \right.
$$

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

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

3 Terms and subterms

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 $T(\mathcal{G})$ (resp. $T(\mathcal{G}, \mathcal{X})$) the set of terms over $\mathcal{G} \cup C$ (resp. $\mathcal{G} \cup C \cup \mathcal{X}$). The former is called the set of ground terms over $\mathcal{G}$, while the latter is simply called the set of terms over $\mathcal{G}$. Variables are denoted by x, y, v, terms are denoted by s, t, u, and finite sets of terms are written $E, F, ...$, and decorations thereof, respectively.

A *constant* is either a free constant or a function symbol of arity 0. 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 $\mathrm{Var}(t) \cup \mathrm{Cons}(t)$. We denote by $\mathcal{A}$ the set of all constants and variables. A substitution σ is an involutive mapping from $\mathcal{X}$ to $T(\mathcal{G}, \mathcal{X})$ such that $\mathrm{Supp}(\sigma) = \{x \mid \sigma(x) \neq x\}$, 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 $T(\mathcal{G})$.

In this paper, we consider 2 disjoint signatures $\mathcal{F}_1$ and $\mathcal{F}_2$, and 2 consistent equational theories $\mathcal{E}_1$ and $\mathcal{E}_2$ on $\mathcal{F}_1$ and $\mathcal{F}_2$, resp. 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).

The *syntactic subterms* of a term t are defined recursively as follows and denoted $\mathrm{Sub}_{\mathrm{syn}}(t)$. 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 defined recursively as usual (*i.e.* as sequences of integers), ϵ being the empty sequence. We denote by $t_{|p}$ the syntactic subterm of t at position p. We denote by $t[p \leftarrow s]$ the term obtained by replacing in t the syntactic subterm $t_{|p}$ by s. 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 symbol at position ϵ. For $t \in \mathrm{C} \cup \mathcal{X}$ we define $\mathrm{Sign}(t) = \bot$, with $\bot$ a new symbol. The term s is *alien* to u if $\mathrm{Sign}(s) \neq \mathrm{Sign}(u)$.
Factors. We define the set of *factors* of a term t, and denote $\mathrm{Factors}(t)$, the set of maximal syntactic subterms of t that are either alien to t or atoms and different from t. In particular $\mathrm{Factors}(t) = \emptyset$ for $t \in \mathcal{A}$.
Subterms. We now define the notion of *subterm values*. Given a term t, the set of its subterm values is denoted by $\mathrm{Sub}(t)$ and is defined recursively by: $\mathrm{Sub}(t) = \{t\} \cup \bigcup_{u \in \mathrm{Factors}(t)} \mathrm{Sub}(u)$. For a set of terms E, $\mathrm{Sub}(E)$ is defined as the union of the subterms values of the elements of E.

As an example consider $\mathcal{F}_1 = \{\oplus, 0\}$ and $\mathcal{F}_2 = \{f\}$ where f has arity 1. Then $\mathrm{Sub}(a \oplus (b \oplus 0)) = \{a \oplus (b \oplus 0), a, b, 0\}$. On the other hand $\mathrm{Sub}(f(b \oplus c)) = \{f(b \oplus c), b \oplus c, b, c\}$. 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*.

Congruences and ordered rewriting. We shall introduce the notion of *ordered rewriting* [13], which is a useful technique that has been utilized (e.g. [3]) 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 $\mathrm{c}_{\min} \in \mathrm{C}$. Given a possibly infinite set of equations $\mathcal{O}$ on the signature $\mathrm{T}(\mathcal{G})$ we define the ordered rewriting relation $\to_\mathcal{O}$ by $s \to_\mathcal{O} s'$ iff there exists a position p in s, an equation $l = r$ in $\mathcal{O}$ and a substitution τ such that $s = s[p \leftarrow l\tau]$, $s' = s[p \leftarrow r\tau]$, and $l\tau > r\tau$.

It has been shown (see [13]) that by applying the *unfailing completion procedure* [14] 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. $\to_\mathcal{O}$ is convergent (i.e. terminating and confluent) on $\mathrm{T}(\mathcal{F})$.

We shall say that $\mathcal{O}$ is an *o-completion* of $\mathcal{H}$. The relation $\to_\mathcal{O}$ being convergent on ground terms we can define $(t){\downarrow}_\mathcal{O}$ as the unique normal form of the ground term t for $\to_\mathcal{O}$. 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

---

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

$x(\sigma)\!\downarrow_{\mathcal{O}} = (x\sigma)\!\downarrow_{\mathcal{O}}$. A substitution σ is *normal* if $\sigma = (\sigma)\!\downarrow_{\mathcal{O}}$. We will denote by R an o-completion of $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$. We denote by $\mathrm{C}_{\mathrm{spe}}$ the set containing the constants in $\mathcal{F}$ and $\mathrm{c}_{\min}$.

4 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 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 4.1 we define an extension of Dolev-Yao model to arbitrary operators for modeling the possible deductions of the intruder. In Subsection 4.2 we define the protocol semantics for an execution within an hostile environment controlled by the intruder and in Subsection 4.3 we describe how we represent this execution by a constraint system.

4.1 Intruder systems

We shall model messages as ground terms and intruders deduction rules as rewrite rules on sets of messages representing the knowledge of an intruder. An intruder derives new messages from a given (finite) set of messages by applying intruder rules. Since we assume some equational axioms $\mathcal{H}$ are verified 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 1 *An* intruder system $\mathcal{I}$ *is given by a triple* $\langle \mathcal{G}, T, \mathcal{H} \rangle$ *where* $\mathcal{G}$ *is a signature,* $T \subseteq \mathrm{T}(\mathcal{G}, \mathcal{X})$ *and* $\mathcal{H}$ *is a set of axioms between terms in* $\mathrm{T}(\mathcal{G}, \mathcal{X})$. *To each* $t \in T$ *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:

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

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

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 $\mathcal{I}$.

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 $a, b, c \to_{\mathcal{I}_{\times}} a, b, c, c \times a$ by applying the rule $c, a \to c \times a \in \mathrm{L}^{x \times y, \mathrm{g}}$.

A *derivation* D of length n, $n \geq 0$, is a sequence of steps of the form $E_0 \to_{\mathcal{I}} E_0 \cup \{t_1\} \to_{\mathcal{I}} \cdots \to_{\mathcal{I}} E_n$ 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 $i \in \{1, \ldots, n\}$. The term t_n is called the *goal* of the derivation. We define $\overline{E}^{\mathcal{I}}$ to be equal to the set $\{t \mid \exists F \text{ s.t. } E \rightarrow_{\mathcal{I}}^* F \text{ and } t \in F\}$ *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 $\mathcal{H}$. We will assume from now that all the deduction rules generate terms that are normalized by $\rightarrow_{\mathcal{O}}$ and the goal and the initial set are in normal form for $\rightarrow_{\mathcal{O}}$. It can be shown [9] that this is not restrictive for our main decidability result.

Given a set of terms $T \subseteq \mathrm{T}(\mathcal{G}, \mathcal{X})$ we define the set of terms $\langle T \rangle$ to be the minimal set such that $T \subseteq \langle T \rangle$ and for all $t \in \langle T \rangle$ and for all substitutions σ with image included in $\langle T \rangle$, we have $t\sigma \in \langle T \rangle$. Hence terms in $\langle T \rangle$ are built by composing terms in T iteratively. We can prove easily that the intruder systems $\mathcal{I} = \langle \mathcal{G}, T, \mathcal{H} \rangle$ and $\mathcal{J} = \langle \mathcal{G}, \langle T \rangle, \mathcal{H} \rangle$ 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 the union of the 2 intruder systems $\mathcal{I}_1 = \langle \mathcal{F}_1, T_1, \mathcal{E}_1 \rangle$ and $\mathcal{I}_2 = \langle \mathcal{F}_2, T_2, \mathcal{E}_2 \rangle$. In particular we are interested in the derivations obtained by using $\rightarrow_{\mathcal{I}_1} \cup \rightarrow_{\mathcal{I}_2}$. It can be noticed that $\langle T_1 \cup T_2 \rangle = \langle \langle T_1 \rangle \cup \langle T_2 \rangle \rangle$. Hence by the remarks above the derivable terms using $\langle T_1 \cup T_2 \rangle$ or $\langle T_1 \rangle \cup \langle T_2 \rangle$ are the same. For technical reason it will be more convenient to use $\langle T_1 \rangle \cup \langle T_2 \rangle$ for defining the union of 2 intruder systems:

Definition 2 *The* union of the two intruder systems $\langle \mathcal{F}_1, T_1, \mathcal{E}_1 \rangle, \langle \mathcal{F}_2, T_2, \mathcal{E}_2 \rangle$ *is the intruder system* $\mathcal{U} = \langle \mathcal{F}, \langle T_1 \rangle \cup \langle T_2 \rangle, \mathcal{E} \rangle$.

A derivation $E_0 \rightarrow_{\mathcal{U}} E_0 \cup \{t_1\} \rightarrow_{\mathcal{U}} \cdots \rightarrow_{\mathcal{U}} E_n$ of intruder system $\mathcal{U}$ is *well-formed* if for all $i \in \{1, \ldots, n\}$ we have $t_i \in \mathrm{Sub}(E_0 \cup \{t_n\})$; in other words every message generated by an intermediate step either occurs in the goal or in the initial set of messages. In the following lemma the derivations refer to the intruder system $\mathcal{U} = \langle \mathcal{F}, \langle T_1 \rangle \cup \langle T_2 \rangle, \mathcal{E} \rangle$. For the proof see [9]:

Lemma 1. *A derivation of minimal length starting from E of goal t is well-formed.*

4.2 Protocol analysis

In this subsection we describe how protocols 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 [12].

In Dolev-Yao model the intruder 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 side a finite execution of a protocol is the interleaving of a finite sequence of messages she has to send and a finite sequence of messages she receives (and add to her knowledge).

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

A *step* is a triplet $(\text{RECV}(x); \text{SEND}(s); \text{COND}(e))$ where $x \in \mathcal{X}$, $s \in \text{T}(\mathcal{G}, \mathcal{X})$ and e is a set of equations between terms of $\text{T}(\mathcal{G}, \mathcal{X})$. 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 1. Consider the following toy protocol where K is a symmetric key initially known by A only:

$$A \to B : \{M \oplus B\}_K$$
$$B \to A : B$$
$$A \to B : K$$
$$B \to A : M$$

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

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

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 $v_i \overset{?}{=} t_i$ as a condition for this step.

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 3 *(Unification systems)* Let $\mathcal{H}$ be a set of axioms on $\text{T}(\mathcal{G}, \mathcal{X})$. An $\mathcal{H}$-Unification system $\mathcal{S}$ is a finite set of equations in $\text{T}(\mathcal{G}, \mathcal{X})$ denoted by $(t_i \overset{?}{=} u_i)_{i \in \{1,\dots,n\}}$. It is satisfied by a ground substitution σ, and we note $\sigma \models \mathcal{S}$, if for all $i \in \{1, \dots, n\}$ $t_i \sigma =_{\mathcal{H}} u_i \sigma$.

Let $\mathcal{I} = \langle \mathcal{G}, T, \mathcal{H} \rangle$ 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 $\langle (\text{RECV}(x); \text{SEND}(s); \text{COND}(e)) \cdot P, N \rangle$ a transition to (P', N') is possible iff there exists a ground substitution σ such that $x\sigma \in \overline{N}$, $\sigma \models e$, $N' = N \cup \{s\sigma\}$ and $P' = P\sigma$. Trace based-security properties like secrecy can be reduced to the *Execution feasibility* problem:

Execution feasibility

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

4.3 Constraints systems

We express the execution feasibility of a protocol by a constraint problem $\mathcal{C}$.

Definition 4 *(Constraint systems) Let $\mathcal{I} = \langle \mathcal{G}, T, \mathcal{H} \rangle$ be an intruder system. An $\mathcal{I}$-Constraint system $\mathcal{C}$ is denoted: $((E_i \triangleright v_i)_{i \in \{1,\dots,n\}}, \mathcal{S})$ and it is defined by a sequence of couples $(E_i, v_i)_{i \in \{1,\dots,n\}}$ with $v_i \in \mathcal{X}$ and $E_i \subseteq \mathrm{T}(\mathcal{G}, \mathcal{X})$ for $i \in \{1,\dots,n\}$ and $E_{i-1} \subseteq E_i$ for $i \in \{2,\dots,n\}$ and by an $\mathcal{H}$-unification system $\mathcal{S}$. It is* deterministic *iff for all $i \in \{1,\dots,n\}$, $\mathrm{Var}(E_i) \subseteq \{v_1,\dots,v_{i-1}\}$*

An $\mathcal{I}$-Constraint system $\mathcal{C}$ is satisfied by a ground substitution σ if for all $i \in \{1,\dots,n\}$ we have $v_i \sigma \in \overline{E_i \sigma}$ and if $\sigma \models \mathcal{S}$. We denote that a ground substitution σ satisfies a constraint system $\mathcal{C}$ 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 $\mathcal{C}$ (where $\mathcal{O}$ is an o-completion of H). 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.

The condition defining the *deterministic* constraint systems expresses that a message to be sent at some step i should be built from previously received messages recorded in the variables v_j for $j < i$ and from the initial knowledge.

Example 2. We model the protocol of Example 1 by the following constraint system. First we gather all conditions in a unification system $\mathcal{S}$

$$\mathcal{S} = \left\{ v_1 \stackrel{?}{=} \mathrm{c_{min}} \ , \ v_3 \stackrel{?}{=} B \ , \ v_2 \stackrel{?}{=} \mathrm{se}(x, v_4) \ , \ v_5 \stackrel{?}{=} M \right\}$$

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

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

The *deterministic* condition imposes to write the last message $\mathrm{sd}(v_2, v_4)$ instead of x though both are equivalent with respect to satisfiability.

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. With respect to this use we can always assume $c_{\min}$ is the minimum of $\prec$ in the following definition:

Ordered Satisfiability

> **Input:** an $\mathcal{I}$-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:
> $$\begin{cases} \sigma \models_{\mathcal{I}} \mathcal{C} \\ \forall x \in X \text{ and } \forall c \in C, x \prec c \text{ implies } c \notin \mathrm{Sub}_{\mathrm{syn}}(x\sigma) \end{cases}$$

The main result of this paper is the following modularity result:

Theorem 1 *If the ordered satisfiability problem is decidable for two intruders $\langle \mathcal{F}_1, T_1, \mathcal{E}_1 \rangle$ and $\langle \mathcal{F}_2, T_2, \mathcal{E}_2 \rangle$ for disjoint signatures $\mathcal{F}_1$ and $\mathcal{F}_2$ then the satisfiability problem is decidable for* deterministic *constraint systems for the intruder* $\mathcal{U} = \langle \mathcal{F}, \langle T_1 \rangle \cup \langle T_2 \rangle, \mathcal{E} \rangle$.

This result is obtained as a direct consequence of the next section where we give an algorithm for solving $\mathcal{U}$-constraints using algorithms for solving *ordered satisfiability* for intruders $\langle \mathcal{F}_1, T_1, \mathcal{E}_1 \rangle$ and $\langle \mathcal{F}_2, T_2, \mathcal{E}_2 \rangle$.

5 Combination of decision procedures

We introduce Algorithm 1 for solving satisfiability of constraint systems for the union $\mathcal{U}$ of two intruders systems $\mathcal{I}_1 = \langle \mathcal{F}_1, T_1, \mathcal{E}_1 \rangle$ and $\mathcal{I}_2 = \langle \mathcal{F}_2, T_2, \mathcal{E}_2 \rangle$ with disjoint signatures $\mathcal{F}_1$ and $\mathcal{F}_2$. The completeness of Algorithm 1 is sketched below, and the proofs (for completeness and soundness) are fully detailed in [9]. Let us explain this algorithm:

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) homogeneous 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'.

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

Steps 5–8 choose and order non-deterministically the intermediate subterms in derivations that witness that the solution satisfies the constraints in $\mathcal{D}$.

Step 9 defines a constraint problem $\mathcal{C}'$ collecting the previous choices on subterms identification, subterms signatures and derivation structures.

Step 10 splits the problem $\mathcal{S}'$ in two pure subproblems.

Step 11 splits non-deterministically the problem $\mathcal{D}'$, that is we select for each $E \triangleright v$ in $\mathcal{D}'$ an intruder system to solve it.

Step 12 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 13 solves independently the 2 pure subproblems obtained at steps 10–11. 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** $\mathcal{C} = ((E_i \triangleright v_i)_{i \in \{1,\dots,n\}}, \mathcal{S})$ 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: **Choose** a theory $th(q) \in \{0, 1, 2\}$ **for all** $q \in Q$
5: **for** $i = 1$ to n **do**
6:     Choose $Q_i \subseteq Q$
7:     Choose a linear ordering over the elements of Q_i say $(q_{i,1}, \dots, q_{i,k_i})$
8: **end for**
9: **Let** $\mathcal{C}' = (\mathcal{D}', \mathcal{S}')$ where
$$\begin{cases} \mathcal{S}' = \mathcal{S} \cup \left\{ z \overset{?}{=} \psi(z) \mid z \in \mathrm{Sub}(C) \right\} \\ \mathcal{D}' = \Delta_1, \dots, \Delta_i, \dots \Delta_n \end{cases}$$

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

$$\begin{cases} K_i = \psi(E_i) \cup \bigcup_{j=1}^{i-1} Q_j \\ Q_i^{<j} = q_{i,1}, q_{i,2}, \dots, q_{i,j-1} \end{cases}$$

10: **Split** $\mathcal{S}'$ into $\mathcal{S}_1, \mathcal{S}_2$ such that $\mathcal{S}' = \mathcal{S}_1 \cup \mathcal{S}_2$ and:
$$\begin{cases} \mathcal{S}_1 = \left\{ z \overset{?}{=} z' \in \mathcal{S}' \mid z, z' \text{are pure 1-terms} \right\} \\ \mathcal{S}_2 = \left\{ z \overset{?}{=} z' \in \mathcal{S}' \mid z, z' \text{are pure 2-terms} \right\} \end{cases}$$

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

We assume $\mathrm{C_{spe}} \subseteq \mathrm{Sub}(\mathcal{C})$. Recall that R is the rewrite system associated to $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$. 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 $(u\sigma)\downarrow_R = t$. A key proposition is:

Proposition 1 *If $\mathcal{C}$ is a satisfiable constraint system there exists a bound substitution σ such that $\sigma \models \mathcal{C}$. Moreover $\mathrm{Sub}((\mathrm{Sub}(\mathcal{C})\sigma)\downarrow_R) = (\mathrm{Sub}(\mathcal{C})\sigma)\downarrow_R$.*

5.1   Completeness of the algorithm

Proposition 2 *If $\mathcal{C}$ is satisfiable then there exists $\mathcal{C}_1$ and $\mathcal{C}_2$ satisfiable at Step 13 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 from $\mathrm{Sub}(\mathcal{C})$ to a set of variables Q such that $\psi(t) = \psi(t')$ if and only if $(t\sigma)\!\downarrow_R = (t'\sigma)\!\downarrow_R$. Thus by Proposition 1 there exists a bijection ϕ from Q to $\mathrm{Sub}((\mathrm{Sub}(C)\sigma)\!\downarrow_R)$. We let $th(q) = i$ if $\mathrm{Sign}(\phi(q)) = \mathcal{F}_i$ and $th(q) = 0$ if $\mathrm{Sign}(\phi(q)) = \perp$. By the construction of $\mathcal{S}'$ and the choice of ψ we can extend σ on Q by $q\sigma = (\psi^{-1}(q)\sigma)\!\downarrow_R$ for all $q \in Q$.

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

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

We have $\mathrm{Sub}(F_i, g_i) \subseteq \mathrm{Sub}((\mathrm{Sub}(\mathcal{C}\sigma))\!\downarrow_R)$. 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, $\mathrm{Sub}((\mathrm{Sub}(\mathcal{C}\sigma))\!\downarrow_R) = (\mathrm{Sub}(\mathcal{C})\sigma)\!\downarrow_R$. 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 $\psi(t)\sigma = (t\sigma)\!\downarrow_R$) by construction. Since $\mathcal{S}$ is satisfiable, following the lines of F. Baader and K. Schulz [3] permits to prove that $\mathcal{S}_1$ and $\mathcal{S}_2$ are satisfiable with a 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.

¿From the soundness and completeness of Algorithm 1 we can derive our main result on the combination of two intruders. It can be easily generalized to n intruders over disjoint signatures $\mathcal{F}_1, \ldots, \mathcal{F}_n$.

The main drawback of the combination algorithm that we have presented here is that it requires the solvability of general constraints from sub-theories. However the decision procedures which already exist for fixed intruder theories are limited to *deterministic* constraint systems. Fortunately we have been able to show ([9]) that our combination algorithm can be adapted so that it suffices to decide the solvability of *deterministic* constraints systems in sub-theories.

6   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 we believe that it will contribute to a better understanding of complexity issues in the domain. Since constraint satisfiability is required only from the intruder sub-theories the approach should permit one to handle more complex protocols than with alternative techniques.

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. R. Amadio, D. Lugiez, and V. Vanackère. On the symbolic reduction of processes with cryptographic functions. *Theor. Comput. Sci.*, 290(1):695–740, 2003.
3. 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.
4. 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.
5. M. Boreale. Symbolic trace analysis of cryptographic protocols. In *Proceedings of the 28th ICALP'01*, LNCS 2076, pages 667–681. Springer-Verlag, Berlin, 2001.
6. 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.
7. N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: the insecurity of 802.11. In *Proceedings of MOBICOM 2001*, pages 180–189, 2001.
8. 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.
9. Y. Chevalier and M. Rusinowitch. Combining intruder theories. Technical report, INRIA, 2005. http://www.inria.fr/rrrt/rr-5495.html.
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. 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.
12. 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.
13. N. Dershowitz and J-P. Jouannaud. Rewrite systems. In *Handbook of Theoretical Computer Science, Volume B*, pages 243–320. Elsevier, 1990.
14. J. Hsiang and M. Rusinowitch. On word problems in equational theories. In *ICALP*, volume 267 of *Lecture Notes in Computer Science*, pages 54–71. Springer, 1987.
15. 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.
16. 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.
17. J. Millen and V. Shmatikov. Symbolic protocol analysis with an abelian group operator or Diffie-Hellman exponentiation. *Journal of Computer Security*, 2005.
18. M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proceedings of CSFW 2001*. IEEE, 2001.
19. M. Schmidt-Schauß. Unification in a combination of arbitrary disjoint equational theories. *J. Symb. Comput.*, 8(1/2):51–99, 1989.