A Simple Constraint-solving Decision Procedure for Protocols with Exclusive or

Yannick Chevalier

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Simple Constraint-solving Decision Procedure for Protocols with Exclusive or

Y. Chevalier

N° 5224

Juin 2004

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--5224--FR+ENG

A Simple Constraint-solving Decision Procedure for Protocols with Exclusive or*

Y. Chevalier

Thème 2 — Génie logiciel
et calcul symbolique
Projet Cassis

Rapport de recherche n° 5224 — Juin 2004 — 35 pages

Abstract: We present a procedure for deciding security of protocols employing the Exclusive or operator. This procedure relies on a direct combination of a constraint solver for security protocol with a unification algorithm for the exclusive-or theory. Hence compared to the previous ones it is much simpler and easily amenable to automation. The principle of the approach can be applied to other theories too.

Key-words: Security Protocols, Decision Procedure, Unification, Rewriting, Exclusive-or

* This work was partly supported by the RNTL project PROUVE-03V360 and the European project AVISPA IST-2001-39252.

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

Une procédure de décision simple pour la résolution de contraintes avec ou-exclusif[†]

Résumé : Nous présentons une procédure permettant de décider la sécurité de protocoles cryptographiques dans le cadre d'un nombre borné d'acteurs en présence du *ou-exclusif*. Cette procédure est basé sur des techniques de combinaison d'algorithmes d'unifications. Par rapport aux travaux précédents, elle a une présentation beaucoup plus simple et est implantable efficacement. L'approche développée se généralise aux cas d'autres théories.

Mots-clés : Protocoles de sécurité, procédure de décision, unification, réécriture, ou-exclusif.

1 Introduction

Security protocol analysis has been intensively studied [23, 25, 29, 15, 20, 19] motivated by the threats on internet communications and their dramatic consequences. 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 [1, 5, 17, 28, 24, 2]. Among the different approaches the symbolic ones [24, 11, 27, 4, 21] are based on reducing the problem to constraint solving in a term algebra. Constraint solving has proved to be quite effective on standard benchmarks [13] and also was able to discover new flaws on several protocols [12].

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. In particular for the XOR operator (which is frequently used in protocol design) Ryan and Schneider [30] give a simple attack on Bull's recursive authentication protocol: the protocol is used to distribute a connected chain of keys linking all the nodes from originator to the server, but if one key is compromised the others can be compromised too thanks to the property of XOR. Conversely, if XOR is considered as a free operator then, as shown by L. Paulson using the Isabelle prover [26], the protocol is secure. For attacks exploiting the XOR properties in the context of mobile communications see [7].

Therefore several attempts have been made to extend the protocol decision procedures to incorporate algebraic properties in the Dolev-Yao model and relax in that way the so-called *perfect encryption assumption* (i.e. One needs a decryption key to extract the plaintext from the ciphertext, and also, a ciphertext can be generated only with the appropriate key and message). To our knowledge only two such procedures have been proposed for handling the properties of the XOR operator: [9] gives a decision procedure for XOR with an optimal NP complexity and [14] describes a constraint solving procedure for a more general class of protocols but with a higher complexity. Both procedures seem difficult to be turned into effective verification tools. The former is based on guessing terms of some height and may lead to combinatorial explosion. The latter has also hard complexity and is intricate.

In this paper we present a new procedure for deciding security of protocols employing the XOR operator. This procedure relies on a direct combination of a constraint solver for security protocol with a unification algorithm for the XOR theory. Hence compared to the previous ones it is much simpler and easily amenable to automation. The principle of the approach can be applied to other theories too. An advantage of our approach is that it reuses as much as possible results from unification theory and especially combination techniques, instead of reconstructing them in ad-hoc way for security constraints.

Structure of the paper. In the following section, we provide an example illustrating the role of XOR in attacks. We then present the theoretical framework of our study, and our model of protocols (in Section 4) and of the hostile environment in Section 5. After this, we define the Simultaneous Construction Problems, and the translation from protocols to

SCPs in Section 6, and transformations on SCPs in Sections 7 and 8. We give an example demonstrating how this system permits to automatically find an attack on a protocol in Section 9, and prove its termination in Section 10 and its completeness in Section 11. Finally, we deduce from a decision procedure for protocol insecurity in presence of *xor* in the case of a finite number of session in Section 12.

2 Related work

The main result presented in this article is the decidability of the search for attacks on a protocol within a finite number of sessions and no bounds on message length. This result is very similar to the one presented by V. Shmatikov and H. Comon-Lundh in [14], the main difference being that the protocol we consider are only a subset of the protocol they take into account. The reason for this is that we impose some conditions on the occurrences of the variables in the rules of the protocol. However, these conditions are no real restrictions in practice.

To explain why, let us consider the reception of a message m (a ground term) by an agent. Upon receiving this message, the agent matches it with a pattern t, and assigns values to the variables of t according to the result of this matching. We define a protocol to be *deterministic* when given m and t, there is at most one possible assignement, *i.e.* at most one substitution τ such that $t\tau \equiv m$. As is noted in [31], not every deterministic protocol satisfies our restrictions. However, as was already noted in [9], and proven in [8] every *deterministic* protocol can be effectively transformed in a protocol that satisfies these restrictions. We have not considered this larger class of deterministic protocols in order to avoid the introduction of an additionnal ordering on the variables as is done in [1], this additionnal ordering being encoded within the order on messages.

To sum up, we consider the result presented in this article is equivalent for practical purposes to the one presented in [14]. But we claim that this article is very original because the algorithm presented and the proof of it are very different. In [14], but also in all recent papers considering the problem of cryptographic protocols analysis in presence of an algebraic operators [9, 10, 31, 6], the equalities induced by the theory of the operator are hard-wired into the deduction system and into the proof. The proofs are technically involved when only one algebraic operator is considered, and we do not know any attempt to consider several independant algebraic operators at the same time.

On the other hand, it can be shown that the system presented in this article terminates, is sound and is complete as soon as the theory has some simple properties. We plan to write soon an extension of the result presented to take into account several algebraic operators.

3 A Motivating Example

We illustrate that when taking the algebraic properties of XOR into account, new attacks can occur. As an example, we use a variant of the Needham-Schroeder-Lowe Protocol [22], i.e.,

the public-key Needham-Schroeder Procotol with Lowe's fix, where in some place, instead of concatenation XOR is used. Using common notation, the protocol is given as follows:

$$\begin{array}{l} 1.\ A \to B :\ \{N_A, A\}^p_{K_B} \\ 2.\ B \to A :\ \{N_B, \oplus(\{N_A, B\})\}^p_{K_A} \\ 3.\ A \to B :\ \{N_B\}^p_{K_B} \end{array}$$

If XOR is interpreted as free symbol, such as pairing, then according to [22] this protocol is secure. In particular, the intruder is not able to get hold of N_B. However, if the algebraic properties of XOR are taken into account, the following attack is possible, which is a variant of the original attack on the Needham-Schroeder Protocol and which allows the intruder I to obtain N_B. In this attack, two sessions run interleaved where the steps of the second session are marked with $'$. In the first session, A talks to the intruder I, and in the second session I, purporting to be A, talks to B. We emphasize that in this attack I generates new messages by applying the XOR operator and uses that $N_A \oplus B \oplus I \oplus B = N_A \oplus I$.

$$\begin{array}{ll} 1.\ A \to I & :\ \{N_A, A\}^p_{K_I} \\ 1'.\ I(A) \to B & :\ \{\oplus(\{N_A, B, I\}), A\}^p_{K_B} \\ 2'.\ B \to I(A) & :\ \{N_B, \oplus(\{N_A, B, I, B\})\}^p_{K_A} \\ 2.\ I \to A & :\ \{N_B, \oplus(\{N_A, B, I, B\})\}^p_{K_A} \\ 3.\ A \to I & :\ \{N_B\}^p_{K_I} \\ 3'.\ I(A) \to B & :\ \{N_B\}^p_{K_B} \end{array}$$

4 Terms and Protocols

4.1 Messages and Knowledge

In this paper we assume the same setting as the one considered in [9]. The messages are modelled by terms over a signature containing a denumerable number of free constants and:

$$\mathcal{F} = \left\{ \langle \_, \_ \rangle, \{\_\}^p_\_, \_^{-1}, \{\_\}^s_\_, \_ \oplus \_, 0 \right\}$$

The $\langle \_, \_ \rangle$ represents the concatenation (pairing) of its two arguments.The constructor $\{\_\}^p_\_$ represents the public key encryption, and for a public key $k \in \mathcal{K}ey$, the term k^{-1} represents the inverse key. The operator $\{\_\}^s_\_$ represents the symmetric key encryption. We call $\mathcal{F}_f$ the set of these operators. Under the hypothesis of non-collision between messages, the operators in $\mathcal{F}_f$ are *free*: For $f, g \in \mathcal{F}_f$, the equality $f(t_1, t_2) = g(t'_1, t'_2)$ holds if, and only if, $f = g$, $t_1 = t'_1$ and $t_2 = t'_2$.

On the other hand, the $\oplus$ operator represents the *Exclusive-Or* operation, and 0 is the constant representing sequences of the 0 bit-value of any length. Given two arbitrary messages a and b we consider that this operator has the following properties:

$$\begin{array}{rcll} x \oplus (y \oplus z) & = & (x \oplus y) \oplus z & (A) \\ x \oplus y & = & y \oplus x & (C) \\ x \oplus 0 & = & x & (U) \\ x \oplus x & = & 0 & (N) \end{array}$$

If we orient from left to right the equations (U) and (N), we get a convergent rewrite system modulo AC. Given a term t, we note $\ulcorner t \urcorner$ the normal form of t. For example, we have:

- $\ulcorner \oplus(\{\oplus(\{a, b\}), b, c\}) \urcorner = \oplus(\{a, c\})$
- $\ulcorner \oplus(\{a, \oplus(\{b, c\})\}) \urcorner = \oplus(\{a, b, c\})$

Given a term t, we define the *factors* of t, denoted $\mathrm{Factor}(t)$:

$$\mathrm{Factor}(t) = \begin{cases} \{t_1, \ldots, t_n\} & \text{if } \ulcorner t \urcorner = \oplus(\{t_1, \ldots, t_n\}) \\ \{t\} & \text{otherwise} \end{cases}$$

We note that all the factors of a normalized term $\ulcorner t \urcorner$ have a free root operator.

The *subterms* are defined on normalized terms. Given a term t such that $t = \ulcorner t \urcorner$, the set of its subterms $\mathrm{Sub}(t)$ is defined inductively by:

$$\mathrm{Sub}(t) = \{t\} \cup \begin{cases} \mathrm{Sub}(t_1) \cup \mathrm{Sub}(t_2) & \text{If } t = f(t_1, t_2) \\ & \text{with } f \text{ a free symbol} \\ \cup_{u \in \mathrm{Factor}(t)} \mathrm{Sub}(u) & \text{Otherwise} \end{cases}$$

Given a set of terms E, we also note $\mathrm{Sub}(E)$ the set $\cup_{t \in E} \mathrm{Sub}(t)$.

If the root operator of a *normalized* term t is a free constructor or a constant, we say t is a free term. Otherwise, we say t is a $\oplus$ term.

Last, we call $\mathcal{X}$ the set of variables, and $\mathrm{T}(\mathcal{F},\mathcal{X})$ (resp. $\lceil\mathrm{T}(\mathcal{F},\mathcal{X})\rceil$) the set of terms (resp. normalized terms). *Substitutions*, noted σ, τ, ... are defined as *mappings* from $\mathcal{X}$ to $\lceil\mathrm{T}(\mathcal{F},\mathcal{X})\rceil$. The application of a substitution σ on a term t, denoted $t\sigma$, consists in the term where all variables $x_1,\ldots,x_n$ of t are replaced by the terms $x_1\sigma,\ldots,x_n\sigma$.

4.2 Protocols

In order to decide the security of a protocol w.r.t secrecy or authentication for a fixed number of protocol sessions can be reduced to deciding the security for a single session by guessing an interleaving of the sessions that leads to the security violation [28]. In the same way we can assume that the protocol steps are linearly ordered (otherwise we simply try all possible orderings). Hence we will consider here only a single session of a protocol defined as a sequence of steps.

The following definition is explained below.

Definition 1 *A* protocol rule *is of the form $R \Rightarrow S$ where R and S are terms.*

A protocol *P is a tuple $(\{R_\iota \Rightarrow S_\iota,\ \iota \in \mathcal{I}\},\mathcal{S})$ where $\mathcal{I}$ is an initial segment of the set of natural numbers, $\mathcal{S}$ is a finite set of normalized messages with $0 \in \mathcal{S}$, the* initial intruder knowledge *and $R_\iota \Rightarrow S_\iota$, for every $\iota \in \mathcal{I}$, is a protocol rule such that*

1. *the terms R_ι and S_ι are normalized;*
2. *for all $x \in \mathrm{Var}(S_\iota)$, there exists $\iota' \leq \iota$ such that $x \in \mathrm{Var}(R_{\iota'})$;*
3. *for any subterm $\oplus(\{t_1,\ldots,t_n\})$ of R_ι, there exists $j \in \{1,\ldots,n\}$ such that $\mathrm{Var}(t_i) \subseteq \cup_{\iota'<\iota}\mathrm{Var}(R_{\iota'})$ for every $i \in \{1,\ldots,n\}\setminus\{j\}$. (Note that since R_ι is normalized, the t_i's are free terms.)*

Intuitively for executing a protocol step $R_\iota \Rightarrow S_\iota$ on receiving a (normalized) message m in a protocol run it is first checked whether m and R_ι match, i.e., whether there exists a ground substitution σ such that $m =_E R_\iota\sigma$. If so $\lceil S_\iota\sigma\rceil$ is returned as output. We always assume that the messages exchanged between principals (and the intruder) are normalized — therefore, m is assumed to be normalized and the output of the above rule is not $S_\iota\sigma$ but $\lceil S_\iota\sigma\rceil$. This is because principals and the intruder cannot distinguish between equivalent terms and therefore they may only work on normalized terms (representing the corresponding equivalence class of terms). Finally we note that since the different protocol rules may share variables, some of the variables in R_ι and S_ι may be already bounded by substitutions obtained from previous applications of protocol rules. We are not actually interested in a normal execution of a protocol but rather in attacks on a protocol. This is the reason why the definition of a protocol contains the initial intruder knowledge.

Condition 1. , in the above definition is not a restriction since the transformation performed by a protocol rule and its normalized variant coincide. Condition 2. guarantees

that when an output is produced with S_ι all variables in S_ι are already "bounded". Otherwise, the output of a protocol rule would be arbitrary, since unbounded variables could be mapped to any message. Condition 3. guarantees that the bounding of variables is deterministic. For example if the protocol rule $\oplus(\{x, y\}) \Rightarrow \langle x, y\rangle$ is the first one and thus, x and y are not bounded, then this rule violates Condition 3: On receiving $\oplus(\{a, b, c\})$, for instance, different substitutions are possible, including $\{x \mapsto \oplus(\{a, b\}), y \mapsto c\}$, $\{x \mapsto \oplus(\{b, d\}), y \mapsto \oplus(\{a, c, d\})\}$, etc. In other words, a principal must guess a substitution. We have shown in [8] that every deterministic protocol can be transformed in a protocol satisfying Condition 3.

The protocol informally described in Section 3 can formally be stated as follows: Agent a plays role A and agent b role B; We define $\mathcal{I} = \{1, 2, 3, 4\}$; The initial knowledge of the intruder is $\mathcal{S} = \{0, \mathsf{a}, b, I, ki, ki^{-1}, ka, kb\}$, and the protocol rules are:

$$\begin{array}{lccc}
1: & 0 & \Rightarrow & \{\langle na, \mathsf{a}\rangle\}^p_{ki} \\
2: & \{\langle x_{na}, \mathsf{a}\rangle\}\, kb & \Rightarrow & \{\langle nb, \oplus(\{x_{na}, b\})\rangle\}^p_{ka} \\
3: & \{\langle x_{nb}, \oplus(\{na, I\})\rangle\}^p_{ka} & \Rightarrow & \{x_{nb}\}^p_{ki} \\
4: & \{nb\}^p_{kb} & \Rightarrow & 0
\end{array}$$

Our aim is to determine, given the initial knowledge of the intruder, the set of substitutions σ such that this ordering of messages corresponds to a possible execution. Before proceeding further, we need to formalize the deduction abilities of the intruder.

5 Threat Model

In this section, we model by hostile environment of execution by an active intruder. The *knowledge* of this intruder as well as its deduction abilities are formalized respectively as a set of terms and as rewrite rules over sets of terms.

5.1 The Intruder

We assume that a protocol is run across a hostile environment where the source of the received messages cannot be established. This environment is modeled by an evil actor, called the *intruder*, trying to reach a state that should be banned by the protocol. If the goal of the protocol is to ensure the secrecy of a data M, the forbidden states are those where the intruder knows M. If the goal of the protocol is to ensure authentication of the participants, the forbidden states are those where a honest participant wrongly assumes that a message M it has received originates from another participant.

An attack on a trace-based property of a protocol can be viewed as a particular interleaving of a finite number of protocol sessions. For instance an attack on secrecy can be modeled by adding a message to the protocol where the intruder has to send the confidential piece of data. If this extended protocol admits a *feasible* execution then the initial protocol can be considered as insecure.

Name	Deduction rule
$\mathrm{L}_{\mathrm{c},\langle \_,\_ \rangle}$	$a, b \rightarrow \langle a, b \rangle$
$\mathrm{L}_{\mathrm{c},\{\_\}^{\mathrm{s}}_{\_}}$	$a, b \rightarrow \{a\}^{s}_{b}$
$\mathrm{L}_{\mathrm{c},\{\_\}^{\mathrm{p}}_{\_}}$	$a, b \rightarrow \{a\}^{p}_{b}$
$\mathrm{L}_{\mathrm{c},\oplus}$	$a_1, \ldots, a_n \rightarrow \ulcorner \oplus(\{a_1, \ldots, a_n\}) \urcorner$

Table 1: Composition rules

An execution is feasible if every message received by the honest agents can be derived by the intruder from his initial knowledge and intercepted messages. Hence building an insecure protocol execution reduces to a system of *constraints* to be solved in a particular term algebra. The variables to be solved correspond to the part of the protocol messages that are not read by the honest agents (e.g. because they are encrypted with an unknown key).

The security of the protocol is assessed versus an intruder as strong as possible. We assume it can *divert* all messages sent by honest participants and add their content to its knowledge, it can *send* messages its knowledge permits to deduce under the identity of other actors, and may perform deductions over its knowledge to this end.

5.2 Intruder Deduction Rules

The *knowledge* of the intruder is represented by a set of normalized messages, *i.e.* is a subset of $\ulcorner \mathrm{T}(\mathcal{F}, \mathcal{X}) \urcorner$. The deductions that the intruder can perform from its knowledge are modeled by rewrite rules $l \rightarrow r$ (read: From l deduce r) where l is a set of messages (a subset of $\mathrm{T}(\mathcal{F}, \mathcal{X})$) and r is a message (a term). The right-hand side of a rule is its *result*. We shall only consider rules $l \rightarrow r$ where both l and r are in normal form (*i.e.* $\ulcorner l \urcorner = l$ and $\ulcorner r \urcorner = r$). In order to have lighter notation, and under this hypothesis, we omit the normalization function $\ulcorner \_ \urcorner$ when not necessary. The available deduction rules are split into two disjoint sets: Composition rules and decomposition rules.

Let $l \rightarrow r$ be a deduction rule. It is a decomposition rule iff there exists $t \in l$ such that r is a strict maximal subterm of t. Otherwise, it is a composition rule. We always assume $r \notin l$, since such rules do not permit to deduce new terms.

In Tables 5.2 and 2 we give the rules considered in this paper. The rule:

$$a_1, \ldots, a_n \rightarrow \ulcorner \oplus(a_1, \ldots, a_n) \urcorner$$

is a composition rule if the right-hand side is a $\oplus$-term, and a decomposition rule if its head operator is a constant or a free constructor.

We note:

- $\mathrm{L}_{\mathrm{c,f}} = \mathrm{L}_{\mathrm{c},\langle \_,\_ \rangle} \cup \mathrm{L}_{\mathrm{c},\{\_\}^{\mathrm{s}}_{\_}} \cup \mathrm{L}_{\mathrm{c},\{\_\}^{\mathrm{p}}_{\_}}$

Name	Deduction rule	Decomposed term	Condition
$\mathrm{L}_{\mathrm{d},\langle \_,\_\rangle^1}$	$\langle a,b\rangle \to a$	$\langle a,b\rangle$	$\emptyset$
$\mathrm{L}_{\mathrm{d},\langle \_,\_\rangle^2}$	$\langle a,b\rangle \to b$	$\langle a,b\rangle$	$\emptyset$
$\mathrm{L}_{\mathrm{d},\{\_\}^{\mathrm{s}}_{\_}}$	$\{a\}^s_b, b \to a$	$\{a\}^s_b$	b
$\mathrm{L}_{\mathrm{d},\{\_\}^{\mathrm{p}}_{\_}}$	$\{a\}^p_b, b^{-1} \to a$	$\{a\}^p_b$	b^{-1}
$\mathrm{L}_{\mathrm{d},\oplus}$	$a_1,\ldots,a_n \to \ulcorner\oplus(\{a_1,\ldots,a_n\})\urcorner$	a_1	$\emptyset$

Table 2: Decomposition rules

- $\mathrm{L}_{\mathrm{d,f}} = \mathrm{L}_{\mathrm{d},\langle \_,\_\rangle^1} \cup \mathrm{L}_{\mathrm{d},\langle \_,\_\rangle^2} \cup \mathrm{L}_{\mathrm{d},\{\_\}^{\mathrm{s}}_{\_}} \cup \mathrm{L}_{\mathrm{d},\{\_\}^{\mathrm{p}}_{\_}}$
- $\mathrm{L}_{\mathrm{c}} = \mathrm{L}_{\mathrm{c,f}} \cup \mathrm{L}_{\mathrm{c},\oplus}, \quad \mathrm{L}_{\mathrm{d}} = \mathrm{L}_{\mathrm{d,f}} \cup \mathrm{L}_{\mathrm{d},\oplus}$
- $\mathrm{L}_{\mathrm{f}} = \mathrm{L}_{\mathrm{c,f}} \cup \mathrm{L}_{\mathrm{d,f}}, \quad \mathrm{L}_{\oplus} = \mathrm{L}_{\mathrm{c},\oplus} \cup \mathrm{L}_{\mathrm{d},\oplus}$
- $\mathrm{L} = \mathrm{L}_{\mathrm{f}} \cup \mathrm{L}_{\oplus} = \mathrm{L}_{\mathrm{c}} \cup \mathrm{L}_{\mathrm{d}}$

Transition relation and derivations. Let E and F be two normalized sets of terms, and L a subset of L. We write $E \to_L F$ if there exists a rule $l \to r$ in L such that $l \subseteq E$ and $F = \{r\} \cup E$. We denote by $\to^*$ the reflexive and transitive closure of $\to$. If $L = \{l \to r\}$, we simply note $E \to_{l\to r} F$. Without loss of generality, we also always assume that if $E \to_{l\to r} F$, then $r \notin E$. Under this assumption, a sequence of transitions $E_1 \to_L \ldots \to_L E_n$ is called a *derivation* on L. We say this derivation starts from E_1 and has goal $E_n \setminus E_{n-1}$. A derivation $D : E \to^*_L F$ starting from E and of goal t is defined to be *well-formed* if $F \subseteq \mathrm{Sub}(E \cup \{t\})$.

We have proved in [9] the following useful result:

Proposition 1 *(Existence of well-formed derivations) Let E be a normalized set of terms and t be a term in normal form. There exists a well-formed formed derivation on* L *starting from E of goal t if, and only if, there exists a derivation on* L *starting from E of goal t.*

5.3 Set of Deducible Messages and Properties

Let R be any system of rewrite rules over sets of terms.

Definition 2 *We note $\overline{E}^R$ the set of messages deducible from E using the rewrite system R:*

$$\overline{E}^R = \{t \,|\, \exists E', E \to^*_R E' \text{ and } t \in E'\}$$

In the case of intruder deduction system L, we simply note $\overline{E}$ the set $\overline{E}^{\mathrm{L}}$. We have given the proofs of the two following propositions as they are both easy and very important. Indeed, Proposition 3 permits to link a set of messages with all the possible instantiations of the variables.

Proposition 2 *Suppose $F \subseteq \overline{E}$. Then $\overline{E} = \overline{F \cup E}$*

PROOF. This proposition is based on the fact that the $\overline{=}$ operator is idempotent ($\overline{\overline{E}} = \overline{E}$) and growing for $\subseteq$ ($E \subseteq F$ implies $\overline{E} \subseteq \overline{F}$). The announced equality follows from these two properties by a double inclusion argument.

Proposition 3 *Let E and F be two sets of terms. We have $\overline{E} = \overline{F}$ iff for all substitutions σ, we have $\overline{E\sigma} = \overline{F\sigma}$*

PROOF. The right to left direction is trivial: Consider the Identity substitution. To prove the left to right direction, we note that for all sets of terms E, if $E \rightarrow_{\mathrm{L}}^{*} E'$, then $\overline{E} = \overline{E'}$. The equality $\overline{E} = \overline{F}$ implies there exists a set of terms G such that $E \rightarrow^{*} G$ and $F \rightarrow^{*} G$. Let σ be a substitution. One can check that $l \rightarrow r \in \mathrm{L}$ implies $\ulcorner l\sigma \urcorner \rightarrow \ulcorner r\sigma \urcorner \in \mathrm{L}$. Thus, we have $\ulcorner E\sigma \urcorner \rightarrow^{*} \ulcorner G\sigma \urcorner$ and $\ulcorner F\sigma \urcorner \rightarrow^{*} \ulcorner G\sigma \urcorner$. By construction, we have:

$$\begin{cases} \ulcorner G\sigma \urcorner \subseteq \overline{\ulcorner E\sigma \urcorner} \\ \ulcorner G\sigma \urcorner \subseteq \overline{\ulcorner F\sigma \urcorner} \end{cases}$$

Thus, Proposition 2 permits to conclude.

6 Simultaneous Construction Problems

We now introduce constraints that an intruder has to solve in order to build a protocol execution leading to an attack.

Definition 3 *A* Construction Problem *is a pair (E, t) noted $E \rhd t$ with E a finite subset of $\ulcorner \mathrm{T}(\mathcal{F}, \mathcal{X}) \urcorner$ and $t \in \ulcorner \mathrm{T}(\mathcal{F}, \mathcal{X}) \urcorner$. A substitution σ satisfies $E \rhd t$ iff $\ulcorner t\sigma \urcorner \in \overline{\ulcorner E\sigma \urcorner}$. In this case, we note:*

$$\sigma \models E \rhd t$$

Definition 4 *A* Simultaneous Construction Problem (SCP) *is a finite sequence of construction problems $(E_i \rhd t_i)_{i \in \{1,\ldots,n\}}$ such that:*

1. *for all $i \in \{1, \ldots, n\}$ and for all $x \in \mathrm{Var}(E_i)$, there exists $j < i$ such that $x \in \mathrm{Var}(t_j)$;*
2. *for all $i, j \in \{1, \ldots, n\}$ with $i < j$, there exists $F_j \subseteq E_j$ such that $\overline{F_j} = \overline{E_i}$*
3. *for all $i \in \{1, \ldots, n\}$, if $\oplus(\{u_1, \ldots, u_l\}) \in \mathrm{Sub}(t_i)$, then there exists at most one j such that $\mathrm{Var}(u_j) \not\subseteq \mathrm{Var}(t_1, \ldots, t_{i-1})$.*

Moreover, we assume that if $n \geq 1$, then $E_1 \neq \emptyset$. In the case of protocol analysis, this can be ensured *e.g.* by stating that the intruder always knows her name. The last condition is true by the Condition 3. on protocols when a SCP is built by an execution order over a protocol P. It will be easy to see that all transformations on SCP's that will be defined in the following preserve this property.

Definition 5 *(Satisfiability of SCP) Let L be a set of deduction rules over sets of terms. A SCP $\mathcal{C}$ is σ-satisfiable for L if:*

$$\text{for all } E \rhd t \text{ in } \mathcal{C}, \quad \sigma \models E \rhd_L t$$

We note $\mathsf{Sol}(\mathcal{C})$ the set of substitutions σ such that $\mathcal{C}$ is σ-satisfiable.

The next definition will allow us to obtain a generic description of the set of solutions of an SCP. Since attacks in our setting are substitutions let us define the set of prefixes of a set of substitutions.

Definition 6 *(Prefix set) A set of substitutions Θ is a* prefix *of a set of substitutions Σ if:*

$$\begin{cases} \forall \sigma \in \Sigma \ \exists \tau \ \exists \rho \in \Theta, \sigma = \rho\tau \\ \forall \tau \in \Theta \ \exists \rho \ \tau\rho \in \Sigma \end{cases}$$

The first condition ensures that all substitutions in the set Σ are instances of a substitution in the prefix set. The second condition ensures that each substitution in the prefix set covers a non-empty subset of Σ. Thus, if Θ is a prefix set of Σ, the set Θ is empty if, and only if, Σ is empty. Note that the notion of prefix set is weaker than the notion of most general unifier (summarized as: All the instances are solutions). Suppose Θ is the prefix set of a set Σ, and let $\tau \in \Theta$. Then there might exist a substitution ρ such that $\tau\rho$ is *not* in Σ. For example, the set $\{\mathrm{Id}\}$ is a prefix set of any non-empty set of substitutions Σ but unless Σ is the set of all substitutions, not all instances of Id are in Σ.

Connection with Protocols. Let $P = (\{R_\iota \Rightarrow S_\iota, \iota \in \mathcal{I}\}, \mathcal{S})$ be a protocol with $\mathcal{I} = \{1, \ldots, n\}$. A sequence of ground terms $m_1, \ldots, m_n$ forms a valid trace of the protocol if there exists a substitution σ such that $m_i = \ulcorner S_i\sigma \urcorner$ for all $i \in \mathcal{I}$ and if the intruder was able at every stage to deduce $\ulcorner R_i\sigma \urcorner$. This condition can be formalized as follows. Let F_i be the knowledge of the intruder after she has diverted the i-th message. One has:

$$\begin{cases} F_0 = \mathcal{S} \\ F_i = F_{i-1} \cup \left\{\ulcorner S_i\sigma \urcorner\right\} \end{cases}$$

The intruder can deduce, at every stage, all m_i if, for all $i \in \mathcal{I}$, one has:

$$\ulcorner R_i\sigma \urcorner \in \overline{F_{i-1}}$$

Conversely, if these facts hold for σ, then $\ulcorner S_1\sigma \urcorner, \ldots, \ulcorner S_n\sigma \urcorner$ is a valid trace of the protocol.

Let us now define:

$$\begin{cases} E_0 = \mathcal{S} \\ E_i = E_{i-1} \cup \{S_i\} \end{cases}$$

From what precedes, all the valid traces of the protocol P are given by the solutions of the SCP:

$$\mathcal{C} = E_0 \rhd R_1, \ldots, E_{n-1} \rhd R_n$$

However, the set $\mathsf{Sol}(\mathcal{C})$ may be infinite. In order to decide whether a protocol has a secrecy attack, it is sufficient to be able to decide whether the SCP $\mathcal{C}$ associated to a protocol is satisfiable. But other trace-based properties need a finer result to be decided. For example, one needs to know the possible values of the protocol variables in order to decide if a given protocol has an authentication flaw. Thus, our aim is not to decide the emptyness of the set $\mathcal{C}$, but to find a finite and symbolic representation of it. As a consequence, we will solve the following more general problem:

- PREFIX($\mathcal{C}$): find a finite prefix set Θ of $\mathsf{Sol}(\mathcal{C})$;

6.1 Solved form

We first introduce the notion of SCP in *solved form*. This notion is a generalization of the one considered in [24] for the Dolev-Yao intruder in the free theory. Given a set of variables $\mathcal{X}$ let us note $\mathrm{FV}(t) = \mathrm{Factor}(t) \cap \mathcal{X}$.

Definition 7 *(Solved form) Let $\mathcal{C} = (E_i \rhd t_i)_{i\in\{1,\ldots,n\}}$ be a SCP, and let $\mathcal{X}$ be the set of variables appearing in $\mathcal{C}$. We say $\mathcal{C}$ is in* solved form *if:*

- $\forall i \in \{1,\ldots,n\}\,,\ \mathrm{FV}(t_i) = \{x_i\}$ *and* $i \neq j$ *implies* $x_i \neq x_j$*;*
- $\mathcal{X} = \{x_1,\ldots,x_n\}$.

This notion is crucial because of the following proposition.

Proposition 4 *Every SCP in solved form is satisfiable.*

PROOF. Let $\mathcal{C} = (E_i \rhd t_i)_{i\in\{1,\ldots,n\}}$ be a SCP in solved form, and let $\mathcal{X} = \{x_1,\ldots,x_n\}$. The conditions on the solved form imply that there exists a bijection between the construction equations $E_i \rhd t_i$ and the variables. Hence up to re-indexing if necessary, one has either $t_i = x_i$ or $t_i = \oplus(\{x_i, t_{i,1},\ldots,t_{i,n_i}\})$. The third property of SCPs then implies that for all $i \in \{1,\ldots,n\}$, one has $\mathrm{Var}(t_{i,1},\ldots,t_{i,n_i}) \subseteq \mathrm{Var}(t_1,\ldots,t_{i-1})$. For $i \in \{1,\ldots,n\}$, let $u_i = 0$ if $t_i = x_i$ and $u_i = \oplus(\{t_{i,1},\ldots,t_{i,n_i}\})$ otherwise. The unification problem $\mathcal{U}$:

$$\forall i \in \{1,\ldots,n\}\,, \quad x_i \stackrel{?}{=} u_i$$

is then in solved form: One obtains a solution σ by replacing, for i from 1 to n, all the variables y in u_i by $y\sigma$ which is already computed. By definition of σ as a solution of $\mathcal{U}$, we have:

$$\ulcorner \mathcal{C}\sigma \urcorner = \ulcorner E_1\sigma \urcorner \rhd 0, \ldots, \ulcorner E_n\sigma \urcorner \rhd 0$$

which is trivially satisfiable. Hence $\mathcal{C}$ has at least one solution σ.

7 Normalization of SCP's

We introduce a *normalization* procedure for transforming an SCP into a simpler equivalent one. In the remaining part of this subsection, we give some normalization rules on SCP's. In this paper we only give normalization rules to prove the completeness of $\mathcal{L}$. For a faster constraint-solving algorithm one may add more normalization rules as the ones given in [8] for free operators such as encryption and pairing.

Proposition 5 *Let*

$$\mathcal{C} = \mathcal{C}_\alpha, E \cup \{u_x\} \rhd t, \mathcal{C}_\beta$$

be a SCP such that there exists $E_x \rhd t_x$ *in* $\mathcal{C}_\alpha$ *with* $\mathrm{FV}(t_x) = \mathrm{FV}(u_x) = \{x\}$ *and* $x \notin \mathrm{Var}(E_x)$. *Then* $\mathcal{C}$ *is equivalent to:*

$$\mathcal{C}' = \mathcal{C}_\alpha, E \cup \left\{\ulcorner\oplus(\{u_x, t_x\})\urcorner\right\} \rhd t, \mathcal{C}_\beta$$

We note $\mathcal{C} \Rightarrow_{\oplus,l} \mathcal{C}'$.

Proof. It is sufficient to prove that, for all substitution σ such that $\sigma \models E_x \rhd t_x$, then $\sigma \models E \cup \{u_x\} \rhd t$ if, and only if, $\sigma \models E \cup \left\{\ulcorner\oplus(\{u_x, t_x\})\urcorner\right\} \rhd t$. Let us prove the *only if* direction, and thus suppose:

$$\begin{cases} \sigma \models E_x \rhd t_x \\ \sigma \models E \cup \{u_x\} \rhd t \end{cases}$$

By Condition 1. on SCPs, there exists $F \subseteq E \cup \{u_x\}$ such that $\overline{F} = \overline{E_x}$. Since $x \notin \mathrm{Var}()(E_x)$, we have $x \notin \mathrm{Var}()(F)$, and thus $F \subseteq E$. The equality $\overline{F} = \overline{E_x}$ and Proposition 3 imply:

$$\ulcorner t_x\sigma\urcorner \in \overline{\ulcorner F\sigma\urcorner} \subseteq \overline{\ulcorner E\sigma\urcorner}$$

Thus, it is clear that:

$$\overline{\ulcorner E\sigma\urcorner \cup \left\{\ulcorner u_x\sigma\urcorner\right\}} = \overline{\ulcorner E\sigma\urcorner \cup \left\{\ulcorner\oplus(\{t_x, u_x\})\sigma\urcorner\right\}}$$

and therefore $\ulcorner t\sigma\urcorner \in \overline{\ulcorner E\sigma\urcorner \cup \left\{\ulcorner u_x\sigma\urcorner\right\}}$ implies $\ulcorner t\sigma\urcorner \in \overline{\ulcorner E\sigma\urcorner \cup \left\{\ulcorner\oplus(\{t_x, u_x\})\sigma\urcorner\right\}}$. In other words we have:

$$\sigma \models E \cup \left\{\ulcorner\oplus(\{u_x, t_x\})\urcorner\right\} \rhd t$$

Proposition 6 *Let*

$$\mathcal{C} = \mathcal{C}_\alpha, E \rhd \{u_x\}, \mathcal{C}_\beta$$

be a SCP such that there exists $E_x \rhd t_x$ *in* $\mathcal{C}_\alpha$ *with* $\mathrm{FV}(t_x) = \mathrm{FV}(u_x) = \{x\}$. *Then* $\mathcal{C}$ *is equivalent to:*

$$\mathcal{C}' = \mathcal{C}_\alpha, E \rhd \left\{\ulcorner\oplus(\{u_x, t_x\})\urcorner\right\}, \mathcal{C}_\beta$$

We note $\mathcal{C} \Rightarrow_{\oplus,r} \mathcal{C}'$.

PROOF. By condition 2. on SCP, there exists a subset F of E such that $\overline{E_x} = \overline{F} \subseteq \overline{E}$. Given a substitution σ, we have $\sigma \in \mathsf{Sol}(\mathcal{C})$ implies:

$$\begin{cases} \ulcorner t_x\sigma \urcorner \in \overline{\ulcorner E_x\sigma \urcorner} \\ \ulcorner u_x\sigma \urcorner \in \overline{\ulcorner E\sigma \urcorner} \end{cases}$$

By Proposition 3, we also have:

$$\ulcorner t_x\sigma \urcorner \in \overline{\ulcorner F\sigma \urcorner}$$

and thus, since $\overline{F} \subseteq \overline{E}$ and again by Proposition 3:

$$\ulcorner t_x\sigma \urcorner \in \overline{\ulcorner E_x\sigma \urcorner}$$

Hence, applying the rule:

$$\ulcorner t_x\sigma \urcorner, \ulcorner u_x\sigma \urcorner \rightarrow \ulcorner \oplus(\{\ulcorner t_x\sigma \urcorner, \ulcorner u_x\sigma \urcorner\}) \urcorner$$

one obtains that $\ulcorner \oplus(\{\ulcorner t_x\sigma \urcorner, \ulcorner u_x\sigma \urcorner\}) \urcorner$ is also in $\overline{\ulcorner E_x\sigma \urcorner}$. But we have the equality:

$$\ulcorner \oplus(\{\ulcorner t_x\sigma \urcorner, \ulcorner u_x\sigma \urcorner\}) \urcorner = \ulcorner \oplus(\{u_x, t_x\})\sigma \urcorner$$

and thus:

$$\sigma \models E \rhd \ulcorner \oplus(\{u_x, t_x\}) \urcorner$$

Thus, and given the notations of the proposition, we have:

$$\sigma \models \mathcal{C}'$$

Conversely, if $\sigma \models \mathcal{C}'$, and by a similar proof, we have $\sigma \models \mathcal{C}$.

Note that in Propositions 5 and 6, by definition of the normalization function $\ulcorner \_ \urcorner$, we have $\mathrm{FV}(\ulcorner \oplus(\{u_x, t_x\}) \urcorner) = \emptyset$.

We write $\mathcal{C} \Rightarrow \mathcal{C}'$ if either $\mathcal{C} \Rightarrow_{\oplus,r} \mathcal{C}'$ or $\mathcal{C} \Rightarrow_{\oplus,l} \mathcal{C}'$. A SCP $\mathcal{C}$ is in normal form if there does not exist $\mathcal{C}'$ such that $\mathcal{C} \Rightarrow \mathcal{C}'$ and if all its subterms are in normal form for $\ulcorner \_ \urcorner$.

8 A System for Solving SCP

We transform the SCP using rules of a system $\mathcal{L}$. These rules can be partitionned in two subsets: those applying to terms whose root symbol, for a solution σ, is in the free theory, called $\mathcal{L}_{\mathrm{f}}$, and those applying to xor terms, called $\mathcal{L}_{\oplus}$. We now define these systems.

The notation $\sigma \in \mathrm{mgu}\,(u, t)$ means that the substitution σ is a most general solution of the unification problem $u \stackrel{?}{=} t$ over the theory $E = E_1 \cup E_2$, with E_1 the free theory over standard Dolev-Yao operations (concatenation $-\langle \_, \_ \rangle -$, symmetric key encryption $-\{\_\}^s_\_ -$ and public key encryption $-\{\_\}^p_\_ -$), and E_2 is the ACUN theory of the $\oplus$ operator and 0. Note that the set of most general unifiers is either empty or finite up to renaming.

8.1 The Rules in $\mathcal{L}_{\mathrm{f}}$

In Figure 1 the symbol f is any free operator in $\{\langle \_, \_ \rangle, \{\_\}^s_\_, \{\_\}^p_\_\}$. Intuitively, the Comp rule is the counterpart for the SCPs of the rules of Table 5.2 that are associated with free constructors. Symmetrically, the Dec rule is the counterpart of the rules of Table 2 associated with free constructors.

The Unif rule plays a different role. It is applied when the term to be built is in the knowledge of the intruder once a substitution is applied.

$$\textsf{Comp}: \quad \frac{\mathcal{C}_\alpha, E \rhd t, \mathcal{C}_\beta}{(\mathcal{C}_\alpha, E \rhd x_1, E \rhd x_2, \mathcal{C}_\beta)\sigma} \ \sigma \in \mathrm{mgu}\,(t, f(x_1, x_2))$$

$$\textsf{Dec}: \quad \frac{\mathcal{C}_\alpha, E \rhd t, \mathcal{C}_\beta}{(\mathcal{C}_\alpha, E \rhd \mathrm{Cond}(f(x_1, x_2)), E \cup \mathrm{Res}(f(x_1, x_2)) \rhd t, \mathcal{C}_\alpha)\sigma} \ \begin{array}{c} u \in E, \\ \sigma \in \mathrm{mgu}\,(u, f(x_1, x_2)) \end{array}$$

$$\textsf{Unif}: \quad \frac{\mathcal{C}_\alpha, E \rhd t, \mathcal{C}_\beta}{(\mathcal{C}_\alpha, \mathcal{C}_\beta)\sigma} \ u \in E, \ \sigma \in \mathrm{mgu}\,(u, t)$$

Figure 1: System $\mathcal{L}_{\mathrm{f}}$ of transformation rules.

8.2 The Rules in $\mathcal{L}_{\oplus}$

The rules in $\mathcal{L}_{\oplus}$ are used to treat specifically the case of the xor operator. They are given in Figure 2.

The $\textsf{Comp}_{\oplus}$ (resp. $\textsf{Dec}_{\oplus}$) rule has no real counterpart in Tables 5.2 and 2. On the other hand, the $\textsf{Apply}_{\oplus}$ rule corresponds to the application of a deduction rule:

$$a_1, \ldots, a_n \rightarrow \ulcorner \oplus(\{a_1, \ldots, a_n\}) \urcorner$$

regardless of whether this is a composition or decomposition rule.

We note $\mathcal{C} \rightarrow_l \mathcal{C}'$ if l is an applicable rule of $\mathcal{L}$.

8.3 Restrictions on the Application of Rules in $\mathcal{L}$

Consider a rule of $\mathcal{L}$:

$$\frac{\mathcal{C} = \mathcal{C}_\alpha, Eq, \mathcal{C}_\beta}{\mathcal{C}'}$$

$$\mathsf{Comp}_{\oplus}: \quad \frac{\mathcal{C}_\alpha, E \rhd \oplus(\{t_1,\ldots,t_n\}), \mathcal{C}_\beta}{\mathcal{C}_\alpha, E \rhd t_1, E \rhd \oplus(\{t_2,\ldots,t_n\}), \mathcal{C}_\beta}$$

$$\mathsf{Dec}_{\oplus}: \quad \frac{\mathcal{C}_\alpha, E \rhd t, \mathcal{C}_\beta}{\mathcal{C}_\alpha, E \rhd \oplus(\{t_2,\ldots,t_n\}), E \cup \{t_1\} \rhd t, \mathcal{C}_\beta} \; \oplus(\{t_1,\ldots,t_n\}) \in E$$

$$\mathsf{Apply}_{\oplus}: \quad \frac{\mathcal{C}_\alpha, E \rhd t, \mathcal{C}_\beta}{(\mathcal{C}_\alpha, \mathcal{C}_\beta)\sigma} \; U \subseteq E,\ \sigma \in \mathrm{mgu}\left(\bigoplus_{u\in U} u, t\right)$$

Figure 2: System $\mathcal{L}_\oplus$ of transformation rules.

Note that in the above notation, $\mathcal{C}_\alpha$ is a SCP, but not $\mathcal{C}_\beta$. We do several hypotheses on $\mathcal{C}$ and $\mathcal{C}$' for such a rule to be applicable.

First, we assume the SCP $\mathcal{C}$ is normalized. This means that, when applying rules, one should start from a SCP in normal form, and each SCP deduced must be normalized before being employed in further deductions.

Second, we impose in the above rule that $\mathcal{C}_\alpha$ has to be empty or in solved form. Moreover, we impose that if a rule applies on a term u or t, then:

$$\mathrm{FV}(u) = \mathrm{FV}(t) = \emptyset$$

Since the SCP is normalized, this is always the case for a term on the left-hand side of a construction problem. If the rule applies on the right-hand side t of $E \rhd t$, this restriction implies this problem is not in solved form.

In order to ensure termination, we also forbid the application of the Dec rule twice on the same term. This may be formalized using *e.g.* tagging of terms.

9 Example

We now illustrate our procedure on the attack example of Section 3. Let us note $E_0 = \left\{a, b, ka, kb, I, ki, ki^{-1}\right\}$ the initial knowledge of the intruder and:

- $E_1 = E_0 \cup \{\{\langle na, a\rangle\}^p_{ki}\}$
- $E_2 = E_1 \cup \{\{\langle nb, \oplus(\{x_{na}, b\})\rangle\}^p_{ka}\}$
- $E_3 = E_2 \cup \{\{x_{nb}\}^p_{ki}\}$

Given the protocol description in Section 4, the SCP corresponding to the search of an attack is:

$$E_0 \rhd 0, E_1 \rhd \{\langle x_{na}, a\rangle\}^p_{kb}, \\ E_2 \rhd \{\langle x_{nb}, \oplus(\{na, I\})\rangle\}^p_{ka}, E_3 \rhd \{nb\}^p_{kb}$$

The first construction problem can be eliminated by the Unif rule. In order to simplify notations, let $\mathcal{C}_\beta = E_2 \rhd \{\langle x_{nb}, \oplus(\{na, I\})\rangle\}^p_{ka}, E_3 \rhd \{nb\}^p_{kb}$.The intruder immediately decompose the message sent by a.

$$\dfrac{\dfrac{\dfrac{E_1 \rhd \{\langle x_{na}, a\rangle\}^p_{kb}, \mathcal{C}_\beta}{E_1 \rhd ki^{-1}, E_1 \cup \{\langle na, a\rangle\} \rhd \{\langle x_{na}, a\rangle\}^p_{kb}, \mathcal{C}_\beta}\ \mathsf{Dec}}{E_1 \cup \{\langle na, a\rangle\} \rhd \{\langle x_{na}, a\rangle\}^p_{kb}, \mathcal{C}_\beta}\ \mathsf{Unif}}{E_1 \cup \{na, a, \langle na, a\rangle\} \rhd \{\langle x_{na}, a\rangle\}^p_{kb}, \mathcal{C}_\beta}\ \mathsf{Dec}$$

From now on, we note $E'_1 = E_1 \cup \{na, a, \langle na, a\rangle\}$. We apply twice the Comp rule on the first remaining problem:

$$\dfrac{\dfrac{E'_1 \rhd \{\langle x_{na}, a\rangle\}^p_{kb}, \mathcal{C}_\beta}{E'_1 \rhd \langle x_{na}, a\rangle, E'_1 \rhd kb, \mathcal{C}_\beta}\ \mathsf{Comp}}{E'_1 \rhd x_{na}, E'_1 \rhd a, E'_1 \rhd kb, \mathcal{C}_\beta}\ \mathsf{Comp}$$

The Unif rule again permits to eliminate the second and third construction problem since a and kb are in E_0. Note that up to this point, all the rules applied preserve the satisfiability, and thus could be implemented as normalization rules. The next step is non-deterministic, since the intruder does not know ka^{-1}.

Let σ be the most general unifier of the two terms

The two terms $\{\langle x_{nb}, \oplus(\{na, I\})\rangle\}^p_{ka}$ and $\{\langle nb, \oplus(\{x_{na}, b\})\rangle\}^p_{ka}$ have only one most general unifier, the substitution σ such that:

$$\sigma \ : \ \left\{ \begin{array}{lcl} x_{nb} & \mapsto & nb \\ x_{na} & \mapsto & \oplus(\{na, b, I\}) \end{array} \right.$$

Let us note $E'_3 = E_3\sigma$. We have:

$$E'_3 = E_1 \cup \{\{\langle nb, \oplus(\{na, I\})\rangle\}^p_{ka}, \{nb\}^p_{ki}\}$$

The rule inference is:

$$\dfrac{E'_1 \rhd x_{na}, E_2 \rhd \{\langle x_{nb}, \oplus(\{na, I\})\rangle\}^p_{ka}, E_3 \rhd \{nb\}^p_{kb}}{E'_1 \rhd \oplus(\{na, b, I\}), E'_3 \rhd \{nb\}^p_{kb}}\ \mathsf{Unif}$$

The $\mathsf{Apply}_\oplus$ rule can be applied (with the identity substitution) to remove the first constraint:

$$\dfrac{b, I, na, \ldots \rhd \oplus(\{na, b, I\}), E_3 \rhd \{nb\}^p_{kb}}{E'_3 \rhd \{nb\}^p_{kb}}\ \mathsf{Apply}_\oplus$$

Before proceeding further, the intruder decomposes the message $\{nb\}^p_{ki}$:

$$\dfrac{\dfrac{E'_3 \rhd \{nb\}^p_{kb}}{E'_3 \rhd ki^{-1}, E'_3 \cup \{nb\} \rhd \{nb\}^p_{kb}}\ \mathsf{Dec}}{E'_3 \cup \{nb\} \rhd \{nb\}^p_{kb}}\ \mathsf{Unif}$$

Finally, this last constraint is eliminated through the application of the rule Comp followed by two applications of the rule Unif with the identity substitution.

$$\dfrac{\dfrac{\dfrac{E'_3 \cup \{nb\} \rhd \{nb\}^p_{kb}}{E'_3 \cup \{nb\} \rhd nb, E'_3 \cup \{nb\} \rhd kb}\ \mathsf{Comp}}{E'_3 \cup \{nb\} \rhd nb}\ \mathsf{Unif}}{\emptyset}\ \mathsf{Unif}$$

In this case, the substitution found was:

$$\sigma \ : \ \left\{ \begin{array}{lcl} x_{nb} & \mapsto & nb \\ x_{na} & \mapsto & \oplus(\{na, b, I\}) \end{array} \right.$$

Note that in this case, the substitution is ground. As a consequence, and by definition of solved forms, this implies that the final SCP produced is empty. Unless other sequences of transformations leading to different substitutions are found, this means that there is only one feasible execution of this protocol.

10 Termination

10.1 Ordering on SCP

In the sequel we compare SCP in order, first, to simplify them and second to prove termination of the system $\mathcal{L}$ of SCP transformations.

First, let us recall that if $\prec_1$ and $\prec_2$ are two well-founded orderings on sets S_1 and S_2 respectively, then:

- The lexicographic ordering on $S_1 \times S_2$ is also well-founded;
- The extension of $\prec_1$ to multisets of elements of S_1 (functions from S_1 to $\mathbb{N}$ of finite support) is also a well-founded ordering.

Given a term t, we define the *scale* of t the pair $(|\mathrm{FV}(t)|, \mathrm{Factor}(t))$. Let us order the scales by $(n,t) \prec_{\mathrm{scale}} (n',t')$ if $n < n'$ or $n = n'$ and the multiset $\mathrm{Factor}(t)$ is smaller than $\mathrm{Factor}(t')$ for the extension of the subterm relation to multisets. Since the order on $\mathbb{N}$ and the subterm ordering are well-founded, so is the ordering on the scales. As a consequence, this ordering can be extended to a well-founded ordering on multisets of scales.

We now give an order on SCPs to prove that our transformation system, as well as the normalization rules, terminates. Let $\mathcal{C}$ be a SCP. We associate to $\mathcal{C}$ a quadruple $\Pi(\mathcal{C}) = (v, \mathcal{M}^d, \mathcal{M}^c, e)$ with:

- v the number of variables in $\mathcal{C}$;
- $\mathcal{M}^d$ the multiset of scales of terms on which it is possible to apply the rule dec ;
- $\mathcal{M}^c$ the multiset of scales of right-hand sides of construction problems in $\mathcal{C}$;
- e the number of construction problems $\mathcal{C}$.

Let $\prec_{\text{scp}}$ be the lexicographic ordering on these quadruples. As a lexicographic combination of well-founded orderings, $\prec_{\text{scp}}$ is well-founded.

We also note that $\Pi(\ulcorner\mathcal{C}\urcorner) \preceq_{\text{scp}} \Pi(\mathcal{C})$. From now on, we assume all the terms of $\mathcal{C}$ are in normal form. Note that if $\mathcal{C} \Rightarrow \mathcal{C}'$, then $\mathcal{C}' \prec \mathcal{C}$. Since $\prec$ is noetherian, the normalization process terminates.

10.2 Termination

We recall that for any SCP $\mathcal{C}$, if $\mathcal{C} \Rightarrow^* \mathcal{C}'$ and $\mathcal{C}'$ is normalized, then $\mathcal{C}' \preceq_{\text{scp}} \mathcal{C}$. Thus, to prove termination of our transformation system, it suffices to prove that for any rule l of $\mathcal{L}$, if $\mathcal{C} \rightarrow_l \mathcal{C}'$, then $\mathcal{C}' \prec_{\text{scp}} \mathcal{C}$. The following theorem is essential to this end, and is proven in the appendix.

Theorem 1 *Let $s \stackrel{?}{=} t$ be a unification problem modulo $E_1 \cup E_2$, and let σ be a most general solution of this problem. Then either $\sigma = \mathrm{Id}$ or:*

$$|\mathrm{Var}(s\sigma) \cup \mathrm{Var}(t\sigma)| < |\mathrm{Var}(s) \cup \mathrm{Var}(t)|$$

Proposition 7 *Let $\mathcal{C}$ be a SCP, l be a rule in $\mathcal{L}$, and suppose $\mathcal{C} \rightarrow_l \mathcal{C}'$. Then $\mathcal{C}' \prec_{\text{scp}} \mathcal{C}$.*

Proof. Let l be a rule of $\mathcal{L}$. First, consider the case where the substitution σ applied is not the identity. In the case of the rules Unif and $\mathsf{Apply}_{\oplus}$, it is evident by Theorem 1 that the number of variables in the resulting SCP is strictly decreasing. In the case of the Comp rule, if t si not a $\oplus$-term, then necessarily $t = f(t_1, t_2)$, and the substitution σ is the identity on the variables of the SCP. Otherwise, since the SCP is in normal form, we have necessarily $\mathrm{FV}(t) = \emptyset$. Thus, there exists $n > 1$ free terms $t_1, \ldots, t_n$ such that $t = \oplus(\{t_1, \ldots, t_n\})$, and thus there exists $i \in \{1, \ldots, n\}$ such that $\ulcorner t\sigma \urcorner = \ulcorner t_i\sigma \urcorner$. Without loss of generality, suppose that $\ulcorner t\sigma \urcorner = \ulcorner t_1\sigma \urcorner$. The substitution σ applied is thus the solution of the equations $t_1 \stackrel{?}{=} f(x_1, x_2)$ and $\oplus(\{t_2, \ldots, t_n\}) \stackrel{?}{=} 0$. Since neither x_1 nor x_2 are variables of $\mathcal{C}$, the solution of the first equation is the identity on $\mathrm{Var}(\mathcal{C})$, whereas the second one contains only variables of $\mathcal{C}$. Since t is in normal form, the identity cannot be a solution of the second equation. Thus, by Theorem 1, there are strictly less variables in $\mathcal{C}'$ than in $\mathcal{C}$. The case of the Dec rule is similar.

Second, if the substitution applied is the identity, one proves that the rule is decreasing for the order $\prec_{\text{scp}}$. This is in particular the case for the $\mathsf{Dec}_{\oplus}$ and the $\mathsf{Comp}_{\oplus}$ rules.

As a corollary of Proposition 7, the well-foundedness of the order $\prec_{\text{scp}}$ implies the following theorem.

Theorem 2 *(Termination of $\mathcal{L}$) Starting from a SCP $\mathcal{C}$, the strategy of application of rules of $\mathcal{L}$ terminates.*

11 Completeness of $\mathcal{L}$

We prove in this section the completeness of the transformation system $\mathcal{L}$ to solve the problems PREFIX($\mathcal{C}$). Before proving Theorem 3 in Subsection 11.3, we first investigate two special cases in Subsections 11.1 and 11.2.

11.1 Case of a SCP solvable using only composition rules

In this subsection, we consider the case of a SCP $\mathcal{C} = \mathcal{C}_\alpha, E \rhd t$ in normal form where $\mathcal{C}_\alpha$ is in solved form, and there exists a substitution σ such that $\mathcal{C}$ is σ-satisfiable for $\mathrm{L_c}$.

Proposition 8 *Either $\mathcal{C}$ is in solved form or there exists $\mathcal{C}'$ satisfiable such that $\mathcal{C} \to_l^* \mathcal{C}'$. Moreover, if σ_τ is the substitution applied on $\mathcal{C}$ to yield $\mathcal{C}'$, there exists a substitution τ such that $\sigma_\tau\tau = \sigma$ and $\mathcal{C}'$ is τ-satisfiable for* $\mathrm{L_c}$.

PROOF. Suppose $\mathcal{C}$ is not in solved form. By definition, there exists a well-formed derivation on $\mathrm{L_c}$ starting from $\ulcorner E\sigma \urcorner$ of goal $\ulcorner t\sigma \urcorner$.

If this derivation is of length 0, i.e. $\ulcorner t\sigma \urcorner \in \ulcorner E\sigma \urcorner$, there exists $u \in E$ such that $\ulcorner u\sigma \urcorner = \ulcorner t\sigma \urcorner$. Hence, it suffice to apply the rule *Unif*. Let $\sigma_\tau \in \mathrm{mgu}(u, t)$. By definition of a mgu, there exists τ such that $\sigma_\tau\tau = \sigma$, and thus $\mathcal{C}\sigma_\tau$ is τ satisfiable.

Else, consider the last rule $F \to \ulcorner t\sigma \urcorner$ of this derivation.

If it is in $\mathrm{L_{c,f}}$, then we can apply the rule Comp. Suppose $\ulcorner t\sigma \urcorner = f(t_1, t_2)$. Then there is a substitution σ_τ such that σ_τ is in $\mathrm{mgu}(t, f(x_1, x_2))$ and is a more general than σ. Thus, there exists τ such that $\sigma_\tau\tau = \sigma$. Thus $\mathcal{C}_\alpha\sigma_\tau$ is τ-satisfiable. Since the last rule is a composition rule $t_1, t_2 \to f(t_1, t_2)$, then $t_1, t_2 \in \overline{E\sigma}^c$, and thus the system $(\mathcal{C}_\alpha, E \rhd x_1, E \rhd x_2)\sigma_\tau$ is τ satisfiable.

Else, $\ulcorner t\sigma \urcorner$ is a $\oplus$-term and the rule is in $\mathrm{L_{c,\oplus}}$. We partition the left-hand side F into two sets F_1 and F_2, with $F_1 \subseteq \ulcorner E\sigma \urcorner$ and $F_2 \cap \ulcorner E\sigma \urcorner = \emptyset$, and let $F_2 = \{a_1, \ldots, a_n\}$.

First, we need to prove the following claim on the rules $\mathrm{L_\oplus}$.

Claim. If $E \to_{\mathrm{L_\oplus}} E, t \to_{\mathrm{L_\oplus}} E, t, u$, then $E \to_{\mathrm{L_\oplus}} u$

> PROOF. Suppose $E \to_{F_1 \to t} E, t \to_{F_2 \to u} E, t, u$. The result is trivial if $t \notin F_2$. Else, and noting Δ the symmetric difference of sets, the rule $(F_1 \Delta (F_2 \setminus \{t\})) \to u$ is a $\mathrm{L_\oplus}$ rule whose left-hand side is included in E.

Claim. One can assume w.l.o.g. that $F_2 = \emptyset$.

PROOF. By the first claim, we can assume no a_i is the result of a $\mathrm{L}_\oplus$ rule. Thus, and since the derivation contains only composition rules all the a_i are terms with a free head operator. The definition of $\mathrm{L}_\oplus$ rules, together with the definition of the normalization function $\ulcorner\_\urcorner$, implies that each a_i is a maximal subterm of a term $u = \oplus(\{u_1, \ldots, u_n\})\sigma$, with either $u = t$ or u in E.
Since the system is normalised by the rule $\Rightarrow$ defined in Section 7, no u_j is a variable, and thus there exists j such that $\ulcorner u_j\sigma\urcorner = a_i$. In the first case, one can apply the rule $\mathsf{Comp}_\oplus$, and in the second case, one can apply the rule $\mathsf{Dec}_\oplus$ to remove a_i from F. After application of these rules, one can assume $F_2 = \emptyset$.
Note that all a_i are built using only composition rules, and thus the SCP obtained is still a SCP σ-satisfiable for $\mathrm{L_c}$.

By the claim, we assume that $F \subseteq \ulcorner E\sigma\urcorner$. Let U be a subset of E such that $\ulcorner U\sigma\urcorner = F$ and $\Sigma_U = \mathrm{mgu}\,(\oplus_{u\in U}u, t)$. Since one has $\ulcorner(\oplus_{u\in U}u)\sigma\urcorner = \ulcorner t\sigma\urcorner$, there exists one substitution σ_τ in Σ_U more general than σ, and let τ be such that $\sigma_\tau\tau = \sigma$. Thus, it is possible to apply the rule $\mathsf{Apply}_\oplus$ with U and σ_τ chosen such that the resulting system $\mathcal{C}$' is τ satisfiable and $\sigma_\tau\tau = \sigma$.

It is now possible to apply the rule $\mathsf{Apply}_\oplus$ with U chosen such that $\ulcorner U\sigma\urcorner = F$.

Proposition 9 *If $\sigma \models_{\mathrm{L_c}} \mathcal{C}_\alpha, E \rhd t$, and $\mathcal{C}_\alpha$ is in solved form, then there exists $\mathcal{C}'$ such that $\mathcal{C} \to^* \mathcal{C}'$ and $\mathcal{C}'$ is in solved form. Moreover, if σ_τ is the substitution applied on $\mathcal{C}$ to yield $\mathcal{C}'$, there exists a substitution τ such that $\sigma_\tau\tau = \sigma$ and $\mathcal{C}'$ is τ-satisfiable for* $\mathrm{L_c}$.

PROOF. Let us apply Proposition 8 iteratively starting from $\mathcal{C}$. This yields a sequence $\mathcal{C}_1, \ldots, \mathcal{C}_n, \ldots$, of SCP such that, for all i, $\mathcal{C} \to^* \mathcal{C}_i$ and $\mathcal{C}_i$ is satisfiable. Since the system $\mathcal{L}$ terminates this sequence is finite. By Proposition 8 the last SCP $\mathcal{C}'$ in the sequence is in solved form. The remaining properties are direct consequences of Proposition 8.

11.2 Case of a SCP in *almost* Decomposed Form

We say a SCP $E_1 \rhd t_1, \ldots, E_n \rhd t_n$ is in a *almost decomposed form for* σ if the SCP $E_1 \rhd t_1, \ldots, E_{n-1} \rhd t_{n-1}$ is σ-satisfiable *for* $\mathrm{L_c}$ and the construction equation $E_n \rhd t_n$ is σ-satisfiable for L.

In this subsection, we consider the case of a SCP $\mathcal{C} = \mathcal{C}_\alpha, E \rhd t$ in normal form where $\mathcal{C}_\alpha$ is in solved form, and there exists a substitution σ such that $\mathcal{C}$ is almost decomposed for σ.

Proposition 10 *With the above notations, either $\mathcal{C}$ is σ-satisfiable for* $\mathrm{L_c}$*, or there exists $\mathcal{C}'$ and two substitutions σ_τ and τ such that $\mathcal{C} \to^* \mathcal{C}'$ and σ_τ is the substitution applied, $\sigma_\tau\tau = \sigma$ and $\mathcal{C}'$ is almost decomposed for τ.*

PROOF. Suppose $\mathcal{C}$ is not σ-satisfiable for $\mathrm{L_c}$, and let D be a well-formed derivation of minimal length

$$\ulcorner E\sigma\urcorner = E_0 \to_{\mathrm{L}} E_1 \to_{\mathrm{L}} \ldots \to_{\mathrm{L}} E_n$$

starting from $\ulcorner E\sigma\urcorner$ and of goal $\ulcorner t\sigma\urcorner$.

By hypothesis, there exists a minimal index i such that $E_i \rightarrow_l E_{i+1}$ is in D and $l \in \mathrm{L_d}$. Let d_i be the term decomposed by this rule.

Since the derivation D is well-formed and of minimal length, a straightforward case analysis on the composition rules shows that d_i is not the result of a composition rule and thus $d_i \in \ulcorner E\sigma \urcorner$.

Let $u \in E$ be a term such that $\ulcorner u\sigma \urcorner = d_i$. If $l \in \mathrm{L_{d,\oplus}}$, we can apply $\mathsf{Dec}_\oplus$, else we apply Dec. The equality $\ulcorner u\sigma \urcorner = d_i$ ensures the existence of a mgu σ_τ and a substitution τ such that $\sigma_\tau\tau = \sigma$. The choice of the first decomposition rule ensures that the resulting system is of the shape $\mathcal{C}'_\alpha, E' \rhd t'$ with $\mathcal{C}'_\alpha$ a SCP that is τ-satisfiable for $\mathrm{L_c}$, and thus $\mathcal{C}'$ is almost decomposed for τ.

Proposition 11 *If $\sigma \models_{\mathrm{L_c}} \mathcal{C}_\alpha, E \rhd t$, and $\mathcal{C}_\alpha$ is in solved form, then there exists $\mathcal{C}'$ such that $\mathcal{C} \rightarrow^* \mathcal{C}'$ and $\mathcal{C}'$ is in solved form. Moreover, if σ_τ is the substitution applied on $\mathcal{C}$ to yield $\mathcal{C}'$, there exists a substitution τ such that $\sigma_\tau\tau = \sigma$ and $\mathcal{C}'$ is τ-satisfiable.*

Proof. Similar to the proof of Proposition 9.

In other words, if a SCP $\mathcal{C}$ is almost decomposed for a substitution σ, there exists a sequence of transformations from $\mathcal{C}$ to a SCP $\mathcal{C}$' in solved form and such that the substitution σ_τ produced by this sequence is more general than σ.

11.3 General Case

We now consider a generic SCP $\mathcal{C}$ in normal form, and a substitution σ such that $\sigma \models \mathcal{C}$.

Theorem 3 *There exists $\mathcal{C}'$ in solved form such that $\mathcal{C} \rightarrow^* \mathcal{C}'$. Moreover, if σ_τ is the substitution applied on $\mathcal{C}$ to yield $\mathcal{C}'$ then there exists a substitution τ such that $\sigma_\tau\tau = \sigma$ and $\mathcal{C}'$ is τ-satisfiable for $\mathrm{L_c}$.*

Proof. By contradiction, suppose $\mathcal{C}$ is a counter-example SCP with a minimal number of construction problems. Under this hypothesis, $\mathcal{C}$ has at least one construction problem, and thus let $\mathcal{C} = \mathcal{C}_\alpha, E \rhd t$.

We note that $\mathcal{C}_\alpha$ is itself a σ-satisfiable SCP. By minimality, there exists a SCP $\mathcal{C}'_\alpha$, and two substitutions σ_τ and τ such that $\mathcal{C}_\alpha \rightarrow^* \mathcal{C}'_\alpha$, the substitution applied from $\mathcal{C}_\alpha$ to $\mathcal{C}'_\alpha$ is σ_τ, $\mathcal{C}'_\alpha$ is τ-satisfiable for $\mathrm{L_c}$ and $\sigma_\tau\tau = \sigma$. By the last equality, $\mathcal{C}\sigma_\tau$ is also τ-satisfiable. Thus, we have a sequence of transformations:

$$\mathcal{C}_\alpha, E \rhd t \rightarrow^* \mathcal{C}'' = \mathcal{C}'_\alpha, \ulcorner E\sigma_\tau \urcorner \rhd \ulcorner t\sigma_\tau \urcorner$$

This SCP is almost decomposed for τ. Thus, Proposition 11 permits one to derive a contradiction.

12 Decidability of Prefix($\mathcal{C}$)

The results obtained permit to decide the problem Prefix($\mathcal{C}$). Starting from $\mathcal{C}$, rules of $\mathcal{L}$ are applied to generate as many SCPs as possible. For each generated SCP $\mathcal{C}$', let $\sigma_{\mathcal{C}'}$ be the substitution applied along the deductions starting from $\mathcal{C}$ and ending in $\mathcal{C}$'. Finally, let us define:

$$\Pi_{\mathcal{C}} = \{\sigma_{\mathcal{C}'} \mid \mathcal{C}' \text{ in solved form}\}$$

We have:

1. Termination: By Theorem 2, the procedure terminates, and thus $\Pi_{\mathcal{C}}$ is finite;
2. Correctness: Every $\sigma \in \Pi_{\mathcal{C}}$ is a prefix of a substitution in $\mathsf{Sol}(\mathcal{C})$;
3. Completeness: By Theorem 3, for every $\tau \in \mathsf{Sol}(\mathcal{C})$, there exists $\sigma \in \Pi_{\mathcal{C}}$ such that σ is a prefix of τ.

Thus, $\Pi_{\mathcal{C}}$ is a prefix set of $\mathsf{Sol}(\mathcal{C})$ and we can state the following theorem:

Theorem 4 *The problem* Prefix($\mathcal{C}$) *is decidable.*

13 Conclusion

For sake of clarity we have not considered the issue of authentication flaws detection. This goal can be modelled by disequations between parts of messages sent and received. These disequations state that either the sender is not the one that is expected or the value received is different from the sent one. Note that our decidability result of [9] does not work with disequations and therefore cannot be easily extended to authentication properties. However the procedure we have presented in this paper computes a finite prefix set of all substitutions satisfying an execution order π. Applying the substitutions in the prefix set on the inequalities it should be easy to deduce whether some of them are satisfiable and thus if there exists an authentication flaw.

As future works we plan to finish the implementation of the protocol analysis procedure and tune it by exploiting possible optimizations of the unification algorithms as they were proposed by [3]. We shall also investigate other theories (e.g. abelian groups) to which the approach applies trying to find a general criteria.

References

[1] R. Amadio and D. Lugiez. On the reachability problem in cryptographic protocols. In *Proceedings of CONCUR'00*, volume 1877 of *Lecture Notes in Computer Science*, 2000.

[2] A. Armando and L. Compagna. Automatic SAT-Compilation of Protocol Insecurity Problems via Reduction to Planning. In *Foundation of Computer Security & Verification Workshops*, Copenhagen, Denmark, July 25-26 2002.

[3] F. Baader and K. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. In D. Kapur, editor, *Proceedings of the 11th International Conference on Automated Deduction*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 50–65. Springer-Verlag, 1992.

[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. Available at `http://www.avispa-project.org`.

[5] M. Boreale. Symbolic trace analysis of cryptographic protocols. In *Proceedings of the 28th International Conference on Automata, Language and Programming: ICALP'01*, LNCS 2076, pages 667–681. Springer-Verlag, Berlin, 2001.

[6] M. Boreale and M. G; Buscemi. Symbolic analysis of crypto-protocols based on modular exponentiation. In *Proceedings of the Mathematical Foundations of Computer Science 2003, 28th International Symposium (MFCS 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. *Résolution de problèmes d'accessibilité pour la compilation et la validation de protocoles cryptographiques*. PhD thesis, LORIA, Université Henri Poincaré Nancy I, Vandoeuvre-les-Nancy, France, December 2003.

[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. Long version available as Technical Report RR-4697, INRIA, France.

[10] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the Security of Protocols with Diffie-Hellman Exponentiation and Products in Exponents. In *Proceedings of the Foundations of Software Technology and Theoretical Computer Science, FSTTCS'03*, Lecture Notes in Computer Science. Springer, December 2003. Long version available as Christian-Albrecht Universität IFI-Report 0305, Kiel (Germany).

[11] 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. Long version available as Technical Report A01-R-140, LORIA, Nancy (France).

[12] Y. Chevalier and L. Vigneron. Automated Unbounded Verification of Security Protocols. In E. Brinksma and K. Guldstrand Larsen, editors, *14th International Conference on Computer Aided Verification, CAV'2002*, volume 2404 of *Lecture Notes in Computer Science*, pages 324–337, Copenhagen (Denmark), July 2002. Springer.

[13] 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.

[14] 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.

[15] Grit Denker, Jonathan Millen, and Harald Rueß. The CAPSL integrated protocol environment. Technical report, SRI International, October 2000.

[16] N. Dershowitz and J-P. Jouannaud. Rewrite systems. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B)*, pages 243–320. 1990.

[17] M. Fiore and M. Abadi. Computing symbolic models for verifying cryptographic protocols. In *Proc.14th IEEE Computer Security Foundations Workshop*, Cape Breton, Nova Scotia, June 2001.

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

[19] Pierre Ganty Giorgio Delzanno. Automatic verification of time sensitive cryptographic protocols. In *TACAS*, LNCS, pages 342–356. Springer-Verlag, 2004.

[20] Jean Goubault-Larrecq. A method for automatic cryptographic protocol verification. In Dominique Méry Beverly Sanders, editor, *Fifth International Workshop on Formal Methods for Parallel Programming: Theory and Applications (FMPPTA 2000)*, number 1800 in Lecture Notes in Computer Science. Springer-Verlag, 2000. `http://www.dyade.fr/fr/actions/vip/jgl/cpv.ps.gz`.

[21] Y. Lakhnech L. Bozga and M. Perin. Hermes: An automatic tool for the verification of secrecy in security protocols. In *Proceedings of the Computer-Aided Verification Conference CAV'03*, volume 2725 of *Lecture Notes in Computer Science*, pages 219–222. Springer-Verlag, 2003.

[22] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In Margaria and Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166, 1996.

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

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

[25] J.C. Mitchell, V. Shmatikov, and U. Stern. Finite-state analysis of ssl 3.0. In *Seventh USENIX Security Symposium*, pages 201–216, 1998.

[26] L. C. Paulson. Mechanized proofs for a recursive authentication protocol. In *10th Computer Security Foundations Workshop*, pages 84–95. IEEE Computer Society Press, 1997.

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

[28] 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.

[29] Peter Ryan, Steve Schneider, Michael Goldsmith, Gavin Lowe, and Bill Roscoe. *The Modelling and Analysis of Security Protocols.* Addison Wesley, 2000.

[30] P.Y.A. Ryan and S.A. Schneider. An attack on a recursive authentication protocol. *Information Processing Letters*, 65, 1998.

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

Appendix: Solving Unification Problems

We consider, in this section the case of unification modulo a theory $E = E_1 \cup E_2$ where E_1 is the free theory over a non-empty signature Σ_1, and E_2 is the theory over the signature $\Sigma_2 = \{0, \oplus\}$ presented by the following axioms :

$$\begin{array}{rcll} x \oplus (y \oplus z) & = & (x \oplus y) \oplus z & (A) \\ x \oplus y & = & y \oplus x & (C) \\ x \oplus 0 & = & x & (U) \\ x \oplus x & = & 0 & (N) \end{array}$$

The combination result in [3] implies that unification modulo E is decidable and, in fact, unifiability in NP (see also [18] for a specific treatment of the ACUN theory). Since both theories E_1 and E_2 are unitary, the resulting theory E is finitary, the different possible unifiers stemming from the combination phase. Beside these well-known results, and in order to prove the termination of our constraints-solving algorithm, we need a finer result on the number of variables before and after the application of a most general unifier modulo E.

13.1 Presentation of the combination algorithm

We briefly present, in this section, the combination algorithm given in [3]. This six-steps algorithm consists in the transformation of a unification problem $s \stackrel{?}{=} t$ modulo $E_1 \cup E_2$ into a set $\mathcal{S}$ of equations modulo E_1 or E_2 by guessing some properties of a most general unifier σ (if it exists). We suppose the theories E_1 and E_2 are consistant. We apply *unfailing completion* procedure (see [16] for definition and properties) to obtain a (possibly infinite) confluent rewrite system R that terminates on ground terms over the signature $\Sigma_1 \cup \Sigma_2$. Given a term t, we note $(t)_{\downarrow R}$ its normal form for R. In the sequel, we always assume substitution are in normal form: For all variable x and for all substitution σ, we have $x\sigma = (x\sigma)_{\downarrow R}$.

We denote by $\mathcal{S}_i$ the set of equations after step i, and define $\mathcal{S}_0 = \left\{ s \stackrel{?}{=} t \right\}$. We note $\mathcal{S}_{5,j}$ the subset of equations in $\mathcal{S}_5$ that will be solved in theory $j \in \{1, 2\}$. The aim of the algorithm is to build two *pure* systems of equations, that is one over Σ_1 modulo the theory E_1, and one over Σ_2 modulo the theory E_2. In the process of purification of equations, variables are introduced to represent maximal *alien* subterms of a term (steps 1. and 2.).

1. **terms purification:** For simplification of the presentation, one replaces in a bottom-up way each subterm $r = r'[t_1, \ldots, t_n]$, where the t_i are alien subterms of r, of the initial unification problem by the term $r'[x_1, \ldots, x_n]$ and adds problems $x_i \stackrel{?}{=} t_i$ to $\mathcal{S}_0$, for $i \in \{1, \ldots, n\}$;

2. **equations purification:** Adding variables and problems to $\mathcal{S}_1$ if necessary, one ensures that all equations in $\mathcal{S}_2$ are pure;

3. **variables identification:** One chooses an equivalence relation over variables of $\mathcal{S}_2$, and replaces each variable by the representant of its class;

4. **choice of theory:** One chooses, for each class of variables, a theory 1. or 2. that also will be the signature where the head operator of $x\sigma$ will be defined after application of a mgu σ;

5. **choice of a linear ordering:** One chooses a linear ordering $\prec$ over the classes of variables such that $x \prec y$ implies $y\sigma$ is not a subterm of $x\sigma$;

6. **resolution:** Each pure equation is solved in its theory E_i. A variable in the other theory is considered to be a constant. A unifier σ is built from the solutions σ_1 and σ_2 of the two systems $\mathcal{S}_{5,1}$ and $\mathcal{S}_{5,2}$.

Each equation introduced in steps 1. and 2. is indexed by the subterm of s (or t) to which the variable corresponds. This defines a partial ordering $\prec_l$ over equations introduced during steps 1. and 2.

The choice of theory of a class of variables is important: If a variable is chosen in E_j, for $j \in \{1, 2\}$, it will be considered as a variable in $\mathcal{S}_{5,j}$, and as a free constant in $\mathcal{S}_{5,j'}$ $(j' \neq j)$.

Example. Suppose E_1 is the free theory $f(x, y, z) \equiv_{E_1} f(x, y, z)$ over the signature $\Sigma_1 = \{f\}$, E_2 is the ACUN theory, and we want to solve the problem:

$$y \stackrel{?}{=}_{E_1 \cup E_2} f(x_1 \oplus y, x_2 \oplus y, x_1 \oplus x_2)$$

The system $\mathcal{S}_1$ built after step 1. is, with the indices:

$$\left\{ \begin{array}{lcll} y & \stackrel{?}{=} & f(z_1, z_2, z_3) & \\ z_1 & \stackrel{?}{=} & x_1 \oplus y & (x_1 \oplus y) \\ z_2 & \stackrel{?}{=} & x_2 \oplus y & (x_2 \oplus y) \\ z_3 & \stackrel{?}{=} & x_1 \oplus x_2 & (x_1 \oplus x_2) \end{array} \right.$$

The equations are already pure, so we skip step2. We do not identify any variables at step 3. At step 4., we choose y, z_1, z_2 in the free theory and x_1, x_2, z_3 in the ACUN theory. At step 5, we choose the linear ordering:

$$z_1 \prec z_2 \prec z_3 \prec y \prec x_1 \prec x_2$$

At step 5. we have to solve the systems:

$$\mathcal{S}_{5,1} = \left\{ \begin{array}{lcl} y & \stackrel{?}{=} & f(z_1, z_2, z_3) \end{array} \right.$$

$$\mathcal{S}_{5,2} = \left\{ \begin{array}{lcl} z_1 & \stackrel{?}{=} & x_1 \oplus y \\ z_2 & \stackrel{?}{=} & x_2 \oplus y \\ z_3 & \stackrel{?}{=} & x_1 \oplus x_2 \end{array} \right.$$

with y, z_1, z_2 (resp. x_1, x_2, z_3) considered as variables in $\mathcal{S}_{5,1}$ (resp. $\mathcal{S}_{5,2}$) and as constants in $\mathcal{S}_{5,2}$ (resp. $\mathcal{S}_{5,1}$). The most general unifier of $\mathcal{S}_{5,i}$, for $i \in \{1,2\}$ is σ_i, with:

$$\begin{cases} \sigma_1 = & \{y \mapsto f(z_1, z_2, z_3)\} \\ \sigma_2 = & \{x_1 \mapsto z_1 \oplus y, x_2 \mapsto z_2 \oplus y, \\ & z_3 \mapsto z_1 \oplus z_2\} \end{cases}$$

A most general unifier of:

$$y \stackrel{?}{=}_{E_1 \cup E_2} f(x_1 \oplus y, x_2 \oplus y, x_1 \oplus x_2)$$

is then built inductively over $\prec$:

1. $z_1\sigma = z_1$
2. $z_2\sigma = z_2$
3. $z_3\sigma = z_1 \oplus z_2$
4. $y\sigma = f(z_1, z_2, z_1 \oplus z_2)$
5. $x_1\sigma = z_1 \oplus f(z_1, z_2, z_1 \oplus z_2)$
6. $x_2\sigma = z_2 \oplus f(z_1, z_2, z_1 \oplus z_2)$

We note that, in this example, one has

$$|\mathrm{Var}(s\sigma) \cup \mathrm{Var}(t\sigma)| = 2$$
$$|\mathrm{Var}(s) \cup \mathrm{Var}(t)| = 3$$

The aim of this section is to prove that, for all equations $s \stackrel{?}{=}_E t$ over a combination of the free theory on a non-empty signature Σ_1 and of the ACUN theory, and for all unifier σ found using this procedure, the inequality:

$$|\mathrm{Var}(s\sigma) \cup \mathrm{Var}(t\sigma)| < |\mathrm{Var}(s) \cup \mathrm{Var}(t)|$$

holds as soon as σ is not the identity substitution. This is already the case for the free theory and for the ACUN theory. The main difficulty is to deal with variables introduced at steps 1. and 2.

13.2 General results

In this subsection, E_1 and E_2 are two arbitrary theories such that there exists an algorithm for unification with linear constant restriction in E_1 and E_2. In the sequel, we assume the sets of most general unifiers of the unification problem with linear constant restriction $\mathcal{S}_{5,j}$, for $j \in \{1,2\}$, are not empty, and that σ_j is one most general solution of $\mathcal{S}_{5,j}$, for $j \in \{1,2\}$. Finally, let us call $\mathcal{X}_l$ the set of variables introduced at steps 1. and 2., and $\mathcal{X}_0$ the set of

variables in s or t. After step 3., a substitution $\sigma_{Eq} : \mathcal{X}_0 \cup \mathcal{X}_l \mapsto \mathcal{X}_0 \cup \mathcal{X}_l$ is applied mapping each variable to a representant of its equivalence class. Without loss of generality, we assume $x\sigma_{Eq} \in \mathcal{X}_0$ as soon as there exists a variable from $\mathcal{X}_0$ in the equivalence class of x. Based on this convention, the first lemma does not depend on the theories considered, and is a direct consequence of the combination algorithm.

Lemma 1 *For each variable $x \in \mathcal{X}_l \cap \mathrm{Var}(\mathcal{S}_5)$, there exists in $\mathcal{S}_5$ an equation $x \stackrel{?}{=} r$ where $x \notin \mathrm{Var}(r)$.*

Proof. Let x be such a variable, and consider the set $\Omega_x \subseteq \mathcal{S}_3$ of equations:

$$x \stackrel{?}{=} r$$

The set Ω_x is not empty, since at least one such equation was introduced at the same time as the represent of the class x. Among the equations in Ω_x, suppose $x \stackrel{?}{=} r$ is one of minimal index for $\prec_l$.

Claim. $x \notin \mathrm{Var}(r)$.

Proof. By contradiction, suppose $x \in \mathrm{Var}(r)$. This implies there exists in $\mathcal{S}_2$ a unification problem $y_1 \stackrel{?}{=} r'$ and a variable y_2 such that y_1, y_2 are in the equivalence class of x, and $r = r'\sigma_{Eq}$. By choice of representant in σ_{Eq}, $x \notin \mathcal{X}_0$ implies $y_2 \notin \mathcal{X}_0$. Thus, there exists in $\mathcal{S}_2$ an equation $y_2 \stackrel{?}{=} r_2$. By construction, this equation is of smaller index (for $\prec_l$) than $y_1 \stackrel{?}{=} r'$, and thus there exists in Ω_x a problem $x \stackrel{?}{=} r_2\sigma_{Eq}$ of smaller index than $x \notin \mathrm{Var}(r)$. This contradicts the minimality of $x \stackrel{?}{=} r$.

Note that the lemma implies that $x \neq r$. In the sequel, we transform a system of equations in order to remove variables introduced at steps 1. and 2. To this end, we note $\Pi = (\mathcal{P}_1, \mathcal{P}_2, \delta, \prec)$ a combination problem:

- $\mathcal{P}_i$, for $i \in \{1, 2\}$, is a set of pure i-unification problems;
- We note $\mathrm{Const}(\Pi) = \mathrm{Const}(\mathcal{P}_1) \cup \mathrm{Const}(\mathcal{P}_2)$ and $\mathrm{Var}(\Pi) = \mathrm{Var}_1(\mathcal{P}_1) \cup \mathrm{Var}_2(\mathcal{P}_2)$;
- δ is a set of couples (x, t) where $x \in \mathrm{Const}(\Pi) \setminus \mathrm{Var}(\Pi)$;
- $\prec$ is a linear ordering of variables and free constants of Π.

Note that, given a combination problem Π, we can have $\mathrm{Const}(\Pi) \cap \mathrm{Var}(\Pi) \neq \emptyset$. The set $\mathrm{Const}(\Pi)$ is the subset of variables that are considered as constants either in $\mathcal{P}_1$ or $\mathcal{P}_2$, and $\mathrm{Var}(\Pi)$ is the subset of variables that are treated as variables either in $\mathcal{P}_1$ or $\mathcal{P}_2$. A most general solution σ of the combination problem Π is obtained by the combination of the solutions σ_1 and σ_2 of $\mathcal{P}_1$ and $\mathcal{P}_2$ using the ordering $\prec$ over variables and constants.

Our first aim is to remove part of the variables introduced during step 1. and 2. from the unification problems $\mathcal{P}_1$ and $\mathcal{P}_2$. The problem is that even if a variable x chosen in

theory E_1 is removed (by partial application of a substitution), it may still appear in $\mathcal{P}_2$. We collect in third argument δ the remplacement that can be done on these variables. By the inductive construction of σ over $\prec$, it is clear that if $(x,t) \in \delta$ and $x \notin \mathrm{Var}(t)$, then x will not appear in the expression of the solution σ.

The combination problem produced by the algorithm is initially:

$$\Pi_0 = (\mathcal{S}_{5,1}, \mathcal{S}_{5,2}, \emptyset, \prec)$$

Two sets of combination problems $\mathbb{P}$ and $\mathbb{P}'$ are said to be *equivalent over* $\mathcal{X}$ if $\mathcal{X} \subseteq \mathrm{Var}(\mathbb{P}) \cap \mathrm{Var}(\mathbb{P}')$ and there exists a bijection ψ between the most general unifiers of combination problems in $\mathbb{P}$ and $\mathbb{P}'$ such that, for all most general solution σ of $\mathbb{P}$, and for all $x \in \mathcal{X}$, we have: $x\sigma = x(\psi(\sigma))$.

Proposition 12 *Starting from a combination problem* $\Pi_0 = (\mathcal{S}_{5,1}, \mathcal{S}_{5,2}, \emptyset, \prec)$ *with solutions* $(\sigma_1^1, \sigma_2^1), \ldots, (\sigma_1^m, \sigma_2^m)$ *of* $(\mathcal{S}_{5,1}, \mathcal{S}_{5,2})$*, one can build a set of combination problems* $\left\{\Pi^j = (\mathcal{P}_1^j, \mathcal{P}_2^j, \delta, \prec)\right\}_{j\in\{1,\ldots,m\}}$ *equivalent with* $\{\Pi_0\}$ *over* $\mathcal{X}_0$ *such that* $x \in \mathrm{Var}(\Pi^j)$ *iff either* $x \in \mathcal{X}_0$ *or:*

- $x\sigma_1^j = x\sigma_2^j = x$
- *and there exists a variable* $y \neq x$ *such that* $x \in \mathrm{Var}(y\sigma_{j'}^j)$*, for* $j' \in \{1,2\}$.

Proof. Let σ be a solution of Π_0, $\mathcal{S}'_{5,i}$ be $\mathcal{S}_{5,i}$ in solved form for σ, and let:

$$\Omega_i = \left\{x \in \mathrm{Var}_i(\mathcal{S}'_{5,i}) \cap \mathcal{X}_l \,|\, x \stackrel{?}{=} t_x \in \mathcal{S}'_{5,i} \text{ and } t_x \neq x\right\}$$

Note that the sets Ω_1 and Ω_2 are disjoint, and let:

$$\left\{\begin{array}{lcll} S_j & = & \cup_{x\in\Omega_j}\left\{x \stackrel{?}{=} t_x\right\} & j \in \{1,2\} \\ s_j & = & \cup_{x\in\Omega_j}(x, t_x) & j \in \{1,2\} \end{array}\right.$$

The combination problem $(\mathcal{S}'_{5,1} \setminus S_1, \mathcal{S}'_{5,2} \setminus S_2, s_1 \cup s_2, \prec)$ has one solution equivalent to σ over $\mathcal{X}_0$. The last condition can be ensured by simply removing the equations $x \stackrel{?}{=} x$ over variables $x \in \mathcal{X}_l$ such that x does not appear in the image of any other variable y.

The Proposition 13 implies that if $x \in \mathrm{Var}(\Pi) \cap \mathcal{X}_l$, then x was chosen in the free theory: In $\mathcal{S}_{5,2}$, we can replace every occurrence of a variable in $\mathrm{Var}_2(\mathcal{X}_l)$ by a term t such that $\mathrm{Var}_2(t) \cap \mathcal{X}_l = \emptyset$. This last equality ensures the termination of the remplacement procedure.

Let $x \stackrel{?}{=} t$ be the unification problem given by Lemma 1. If t is a term over Σ_1, x cannot be a 2-term. Thus, t is a 2-term and it is possible to replace x by t in $\mathcal{P}_2$, and add (x,t) to δ. Thus, it is sufficient to handle the case of $x \in \mathcal{X}_l$ and x chosen in the free theory E_1.

13.3 Case of the ACUN theory

We investigate in this section the more specific case where E_1 is the free theory and E_2 is the ACUN theory. Since both E_1 and E_2 are unitary, Π_0 has only one solution σ. Let $\Pi = (\mathcal{P}_1, \mathcal{P}_2, \delta, \prec)$ be the combination problem obtained from $(\mathcal{S}_{5,1}, \mathcal{S}_{5,2}, \emptyset, \prec)$ by application of Proposition 12. We assume that the variables of $\mathcal{P}_2$ are either in $\mathcal{X}_0$ or in $V = \{z_1, \ldots, z_k\}$, and that, for all $i \in \{1, \ldots, k\}$:

- z_i is not equivalent to a variable in $\mathcal{X}_0$;
- $z_i\sigma_1 = z_i\sigma_2 = z_i$ and there exists a variable $y \neq z_i$ such that $z_i \in \mathrm{Var}(y\sigma_j)$, for $j \in \{1, 2\}$.

We note $x_1, \ldots, x_n$ the variables (modulo equivalence) in $\mathcal{X}_0$ chosen in the theory 2, and $y_1, \ldots, y_l$ the variables in $\mathcal{X}_0$ chosen in theory 1.

Finally, for $z_i \in V$, we note $\mathrm{Eq}_{z_i} = z_i \stackrel{?}{=} t_i$ the unification problem given by Proposition 13.

First, let us prove the following lemma that bounds the occurrences of variables in $\mathcal{X}_l$.

Proposition 13 *For each $i \in \{1, \ldots, k\}$, there is an equation $z_i \stackrel{?}{=} t_i$ in $\mathcal{S}_{5,2}$ such that $\mathrm{Var}(t_i) \cap \{z_1, \ldots, z_k\} = \emptyset$.*

Proof. Without loss of generality, consider z_1, let $\mathrm{Eq}_{z_1} = z_1 \stackrel{?}{=} t_1$ the equation given by Lemma 1, and let $z' \in V \cap \mathrm{Var}(t)$. By Lemma 1, we know that $z' \neq z_1$ and, since all the variables in V are different after step 3., we have $z' \neq t_1$. Thus, t_1 is a non-atomic term. Since $z_1\sigma_1 = z_1\sigma_2 = z_1$, t_1 must be a 2-term, and thus Eq_{z_1} is in $\mathcal{S}_{5,2}$. Since $z' \in V$ implies $z' \in \mathcal{X}_l$, let $z' \stackrel{?}{=} t'$ be the equation introduced at the same time as z'. Since z' is a strict subterm of t, it must have been introduced at step 1. Since t is a pure 2-term and z' represents an alien subterm, t' is a 1-term. The fact $z' \in \mathcal{X}_l$ implies that t' is not in $\mathcal{X}_0$, and thus is a non-atomic term. Since E_1 is the free theory, z' must be chosen in theory E_1 and *is instantiated*. This contradicts the fact that $z' \in V$ by point 2., thus proving the proposition.

Corollary 1 *There are at least k independant equations in $\mathcal{S}_{5,2}$.*

Proof. In the ACUN theory, the set of unification problems $\{\mathrm{Eq}_z\}_{z \in V}$ can be viewed as an affine equation system $\{z + t_z = 0\}_{z \in V}$, whose solution is the affine subspace expressed by σ_2. Let $(\alpha_i)_{i \in \{1,\ldots,k\}}$ be a sequence of coefficients such that:

$$\Sigma_{i=1}^{k} \alpha_i z_i = \Sigma_{i=1}^{k} \alpha_i t_i$$

The z_i do not appear on the right-hand side by Proposition 13. Thus, both members of this equation are null. This implies that $\Sigma_{i=1}^{k} \alpha_i \cdot z_i = 0$. Since the z_i are different one from another, the sum is zero if, and only if, all the factors α_i are null. This implies that the pure equations Eq_{z_i} are linearly independant one from another.

The relation between the rank of the system and the number of variables in V given by Corollary 1 permits to bound the number of variables in V.

Proposition 14 *We have:*

$$|\mathrm{Var}_2(\mathcal{X}_0\sigma_2)| + k \leq |\mathcal{X}_0|$$

and if equality holds, then either $k \geq 1$ *or* $\sigma_2 = \mathrm{Id}$.

PROOF. First, let us consider the case $k = 0$. If all equations in $\mathcal{S}_{5,2}$ are trivially satisfiable ($\sigma_2 = \mathrm{Id}$), the results holds. Else, let m be the number of lineary independant equations. The affine solution space (and hence σ_2) is of dimension $n-m$, and thus σ_2 is parameterized by $n - m < n$ variables. Then again, the proposition holds.

Let us now consider the general case $k \geq 1$. Suppose there are p equations in $\mathcal{S}_{5,2}$. Finding the (unique) most general unifier σ_2 amounts to solving an affine system:

$$M \cdot X = N \cdot Y + P \cdot Z$$

where:

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \qquad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_l \end{pmatrix} \qquad Z = \begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix}$$

and M (resp. N, resp. P) is a $p \times n$ (resp. $p \times l$, resp. $p \times k$) matrix over the field $\frac{\mathbf{Z}}{2\mathbf{Z}}$. Using gaussian elimination, there exists two invertible matrices A (of dimension $p \times p$) and B (of dimension $n \times n$) such that:

$$A.M.B = D = \begin{pmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & & 0 & \\ 0 & & & & \ddots \\ & & & & & 0 \end{pmatrix}$$

Let us note $X' = B.X$, $N' = A.N$ and $P' = A.P$. The system above is equivalent to:

$$D.X' = N'.Y + P'.Z$$

Claim. The $p - m$ bottom lines of P' and N' are null.

PROOF. The choices on variables imply that:

- every variables $v, v' \in \{y_1, \ldots, y_l, z_1, \ldots, z_k\}$ are distinct, by choice of equivalence classes over variables at the third step;
- no variable $v \in \{y_1, \ldots, y_l, z_1, \ldots, z_k\}$ is a linear combination of other variables, by the first point and by the fact that, for a solution σ_2 of this system, we have $v\sigma_2 = v$.

It follows that every linear combination:

$$\Sigma_{i=1}^{l} \alpha_i y_i + \Sigma_{j=1}^{k} \beta_j z_j$$

is null *iff* all the α_i, β_j are null. The shape of D and the existence of a solution permit to conclude.

Thus, the matrices M', N' and P' are *at most* of rank m. From A invertible, it follows that the matrices M, N and P are also *at most* of rank m. This implies that there are at most m linearly independant equations in $\mathcal{S}_{5,2}$. Thus, Corollary 1 implies $k \leq m$.

The solution of the homogen system $M.X = 0$ is parameterized by $n-m$ variables, hence there are at most n variables in the expression of a general solution σ of the system. The desired result is obtained after adding the l constants y_j.

Theorem 1 *Let $s \stackrel{?}{=} t$ be a unification problem modulo $E_1 \cup E_2$, and let σ be a most general solution of this problem. Then either $\sigma = \mathrm{Id}$ or:*

$$|\mathrm{Var}(s\sigma) \cup \mathrm{Var}(t\sigma)| < |\mathrm{Var}(s) \cup \mathrm{Var}(t)|$$

PROOF. Let $l = |\mathcal{X}_0 \cap \mathrm{Var}_1(\mathcal{X}_0\sigma_1)|$, $k = |\mathcal{X}_l \cap \mathrm{Var}_1(\mathcal{X}_0\sigma_1)|$, $m = |\mathrm{Var}_2(\mathcal{X}_0\sigma_2)|$ and $n = |\mathrm{Var}_2(\mathcal{X}_0)|$. By Proposition 14, $k + m \leq n$ and if equality holds, then either σ_1 is not the identity or σ_2 is the identity. A property of the free theory is that $l \leq |\mathcal{X}_0| - n$, and if equality holds, then $\sigma_1 = \mathrm{Id}$. Thus, if either σ_1 or σ_2 is not the identity, one among the two above equality is strict, and by summing them, we obtain:

$$k + l + m < |\mathcal{X}_0|$$

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