Computing knowledge in security protocols under convergent equational theories ⋆ (for presentation only)

Ştefan Ciobâcă, Stéphanie Delaune, and Steve Kremer

LSV, ENS Cachan & CNRS & INRIA, France

Abstract. We propose a procedure for the intruder deduction problem and for the static equivalence problem, in the case where cryptographic primitives are modeled by a convergent equational theory.
Our procedure terminates on a wide range of equational theories. In particular, we obtain a new decidability result for a theory of trapdoor commitment that we encountered in the study of e-voting protocols. We also provide a prototype implementation.
This work has been accepted for publication at CADE 22.

1 Introduction

Cryptographic protocols are small distributed programs that use cryptographic primitives such as encryption and digital signatures to communicate securely over a network.

In symbolic approaches to cryptographic protocol analysis, the protocol messages and cryptographic primitives (e.g. encryption) are generally modeled using a term algebra. This term algebra is interpreted modulo an equational theory. Using equational theories provides a convenient and flexible framework for modeling cryptographic primitives [10]. For instance, a simple equational theory for symmetric encryption can be specified by the equation $dec(enc(x, y), y) = x$. This equation models the fact that decryption cancels out encryption when the same key is used.

Traditionally, the knowledge of the attacker is expressed in terms of *deducibility* (e.g. [14, 6]). A message s (intuitively the secret) is said to be deducible from a set of messages φ, if an attacker is able to compute s from φ. To perform this computation, the attacker is allowed, for example, to decrypt deducible messages by deducible keys. The problem of determining if a message is deducible from some set of messages is called the *intruder deduction problem* or simply the *deducibility problem.*

However, deducibility is not always sufficient. Consider for example the case where a protocol participant sends over the network the encryption of one of the constants "yes" or "no" (e.g. the value of a vote). Deducibility is not the right notion of knowledge in this case, since both possible values ("yes" and

⋆ This work has been partly supported by the ANR SeSur AVOTÉ.

"no") are indeed "known" to the attacker. In this case, a more adequate form of knowledge is *indistinguishability* (e.g. [1]): is the attacker able to distinguish between two transcripts of the protocol, one running with the value "yes" and the other one running with the value "no"? Given two sets of messages, the problem of determining if the intruder can distinguish between the two is called the *static equivalence problem* (see for example [2]).

Our contributions. We provide a procedure which is correct, in the sense that if it terminates it gives the right answer, for any convergent equational theory. As deduction and static equivalence are undecidable for this class of equational theories [1], the procedure does not always terminate. However, we show that it does terminate for the class of *subterm convergent* equational theories (already shown decidable in [1]) and several other theories among which the theory of *trapdoor commitment* encountered in our electronic voting case studies [11].

Our second contribution is an efficient prototype implementation of this generic procedure. Our procedure relies on a simple fixed point computation based on a few saturation rules, making it convenient to implement.

Related work. Many decision procedures have been proposed for deducibility (e.g. [6, 3, 12]) under a variety of equational theories modeling encryption, digital signatures, exclusive OR, and homomorphic operators. Several papers are also devoted to the study of static equivalence. Most of these results introduce a new procedure for each particular theory and even in the case of the general decidability criterion given in [1, 9], the algorithm underlying the proof has to be adapted for each particular theory, depending on how the criterion is fulfilled.

The first generic algorithm that has been proposed handles subterm convergent equational theories [1] and covers the classical theories for encryption and signatures. This result is encompassed by the recent work of Baudet *et al.* [5] in which the authors propose a generic procedure that works for any convergent equational theory, but which may fail or not terminate. This procedure has been implemented in the YAPA tool [4] and has been shown to terminate without failure in several cases (e.g. subterm convergent theories and blind signatures). However, due to its simple representation of deducible terms (represented by a finite set of *ground* terms), the procedure fails on several interesting equational theories like the theory of trapdoor commitments. Our representation of deducible terms overcomes this limitation by including terms with variables which can be substituted by any deducible terms.

2 Formal model

2.1 Term algebras

As usual, messages will be modeled using a term algebra. Let $\mathcal{F}$ be a finite set of *function symbols* coming with an arity function $\mathrm{ar} : \mathcal{F} \to \mathbb{N}$. Function symbols of arity 0 are called *constants*. We consider several kind of *atoms* among which an infinite set of *names* $\mathcal{N}$, an infinite set of *variables* $\mathcal{X}$ and a set of parameters $\mathcal{P}$. The set of terms $\mathcal{T}(\mathcal{F},\mathcal{A})$ built over $\mathcal{F}$ and the atoms in $\mathcal{A}$ is defined as

$$\begin{array}{lll} t, t_1, \ldots ::= & & \text{term} \\ & \mid\ a & \text{atom } a \in \mathcal{A} \\ & \mid\ f(t_1, \ldots, t_k) & \text{application of symbol } f \in \mathcal{F}, \mathrm{ar}(f) = k \end{array}$$

A term t is said to be *ground* when $t \in \mathcal{T}(\mathcal{F},\mathcal{N})$. We assume the usual definitions to manipulate terms. We write $\mathrm{fn}(t)$ (resp. $\mathrm{var}(t)$) the set of (free) names (resp. variables) that occur in a term t and $\mathrm{st}(t)$ the set of its (syntactic) subterms. These notations are extended to tuples and sets of terms in the usual way. We denote by $|t|$ the *size* of t defined as the number of symbols that occur in t (variables do not count), and $\#T$ denotes the *cardinality* of the set T.

The set of positions of a term t is written $\mathrm{pos}(t) \subseteq \mathbb{N}^*$. If p is a position of t then $t|_p$ denotes the subterm of t at the position p. The term $t[u]_p$ is obtained from t by replacing the occurrence of $t|_p$ at position p with u. A *context* C is a term with (1 or more) holes and we write $C[t_1, \ldots t_n]$ for the term obtained by replacing these holes with the terms $t_1, \ldots, t_n$. A context is *public* if it only consists of function symbols and holes.

Substitutions are written $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ with $\mathrm{dom}(\sigma) = \{x_1, \ldots, x_n\}$. The application of a substitution σ to a term t is written $t\sigma$. The substitution σ is *grounding for* t if the resulting term $t\sigma$ is ground. We use the same notations for *replacements* of names and parameters by terms.

2.2 Equational theories and rewriting systems

Equality between terms will generally be interpreted modulo an *equational theory*. An equational theory $\mathcal{E}$ is defined by a set of equations $M \sim N$ with $M, N \in \mathcal{T}(\mathcal{F},\mathcal{X})$. Equality modulo $\mathcal{E}$, written $=_\mathcal{E}$, is defined to be the smallest equivalence relation on terms such that $M =_\mathcal{E} N$ for all $M \sim N \in \mathcal{E}$ and which is closed under substitution of terms for variables and application of contexts.

It is often more convenient to manipulate rewriting systems than equational theories. A *rewriting system* $\mathcal{R}$ is a set of rewriting rules $l \rightarrow r$ where $l, r \in \mathcal{T}(\mathcal{F},\mathcal{X})$ and $\mathrm{var}(r) \subseteq \mathrm{var}(l)$. A term t rewrites to t' by $\mathcal{R}$, denoted by $t \rightarrow_\mathcal{R} t'$, if there exists $l \rightarrow r \in \mathcal{R}$, a position $p \in \mathrm{pos}(t)$ and a substitution σ such that $t|_p = l\sigma$ and $t' = t[r\sigma]_p$. We denote by $\rightarrow^+_\mathcal{R}$ the transitive closure of $\rightarrow_\mathcal{R}$, $\rightarrow^*_\mathcal{R}$ its reflexive and transitive closure, and $=_\mathcal{R}$ its reflexive, symmetric and transitive closure.

A rewrite system $\mathcal{R}$ is *convergent* if is *terminating*, i.e. there is no infinite chains $u_1 \rightarrow_\mathcal{R} u_2 \rightarrow_\mathcal{R} \ldots$, and *confluent*, i.e. for every terms u_1, u_2 such that $u_1 =_\mathcal{R} u_2$, there exists u such that $u_1 \rightarrow^*_\mathcal{R} u$ and $u_2 \rightarrow^*_\mathcal{R} u$. A term u is in *$\mathcal{R}$-normal form* if there is no term u' such that $u \rightarrow_\mathcal{R} u'$. If $u \rightarrow^*_\mathcal{R} u'$ and u' is in $\mathcal{R}$-normal form then u' is *a $\mathcal{R}$-normal form of u*. When this reduced form is unique (in particular if $\mathcal{R}$ is convergent), we write $u' = u{\downarrow}_{\mathcal{R}_\mathcal{E}}$.

We are particularly interested in theories $\mathcal{E}$ that can be represented by a convergent rewrite system $\mathcal{R}$, i.e. theories for which there exists a convergent rewrite system $\mathcal{R}$ such that the two relations $=_{\mathcal{R}}$ and $=_{\mathcal{E}}$ coincide. Given an equational theory $\mathcal{E}$ we define the corresponding rewriting system $\mathcal{R}_{\mathcal{E}}$ by orienting all equations in $\mathcal{E}$ from left to right, i.e., $\mathcal{R}_{\mathcal{E}} = \{l \to r \mid l \sim r \in \mathcal{E}\}$. We say that $\mathcal{E}$ is *convergent* if $\mathcal{R}_{\mathcal{E}}$ is convergent.

Example 1. A classical equational theory modelling symmetric encryption is $\mathcal{E}_{enc} = \{dec(enc(x,y),y) \sim x\}$. As a running example we consider a slight extension of this theory modelling *malleable* encryption

$$\mathcal{E}_{mal} = \mathcal{E}_{enc} \cup \{mal(enc(x,y),z) \sim enc(z,y)\}.$$

This malleable encryption scheme allows one to arbitrarily change the plaintext of an encryption. This theory does certainly not model a realistic encryption scheme but it yields a simple example of a theory which illustrates well our procedures. In particular all existing decision procedure we are aware of fail on this example. The rewriting system $\mathcal{R}_{\mathcal{E}_{mal}}$ is convergent.

From now on, we assume given a convergent equational theory $\mathcal{E}$ built over a signature $\mathcal{F}$ and represented by the convergent rewriting system $\mathcal{R}_{\mathcal{E}}$.

2.3 Deducibility and static equivalence

In order to describe the messages observed by an attacker, we consider the following notions of *frame* that comes from the applied-pi calculus [2].

A frame φ is a sequence of messages $u_1, \ldots, u_n$ meaning that the attacker observed each of these message in the given order. Furthermore, we distinguish the names that the attacker knows from those that were freshly generated by others and that are *a priori* unknown by the attacker. Formally, a frame is defined as $\nu\tilde{n}.\sigma$ where $\tilde{n}$ is its set of bound names, denoted by $\mathrm{bn}(\varphi)$, and a replacement $\sigma = \{w_1 \mapsto u_1, \ldots, w_n \mapsto u_n\}$. The parameters $w_1, \ldots, w_n$ enable us to refer to $u_1, \ldots, u_n \in \mathcal{T}(\mathcal{F},\mathcal{N})$. The *domain* $\mathrm{dom}(\varphi)$ of φ is $\{w_1, \ldots, w_n\}$. Given terms M and N such that $\mathrm{fn}(M,N) \cap \tilde{n} = \emptyset$, we sometimes write $(M =_{\mathcal{E}} N)\varphi$ (resp. $M\varphi$) instead of $M\sigma =_{\mathcal{E}} N\sigma$ (resp. $M\sigma$).

Definition 1 (deducibility). *Let φ be a frame. A ground term t is* deducible *in $\mathcal{E}$ from φ, written $\varphi \vdash_{\mathcal{E}} t$, if there exists $M \in \mathcal{T}(\mathcal{F},\mathcal{N} \cup \mathrm{dom}(\varphi))$, called the* recipe*, such that $\mathrm{fn}(M) \cap \mathrm{bn}(\varphi) = \emptyset$ and $M\varphi =_{\mathcal{E}} t$.*

Deducibility does not always suffice for expressing the knowledge of an attacker. For instance deducibility does not allow one to express indistinguishability between two sequences of messages. This is important when defining the confidentiality of a vote or anonymity-like properties. This motivates the following notion of static equivalence introduced in [2].

Definition 2 (static equivalence). *Let φ_1 and φ_2 be two frames such that $\mathrm{bn}(\varphi_1) = \mathrm{bn}(\varphi_2)$. They are* statically equivalent *in $\mathcal{E}$, written $\varphi_1 \approx_{\mathcal{E}} \varphi_2$, if*

– $\mathrm{dom}(\varphi_1) = \mathrm{dom}(\varphi_2)$

– *for all terms* $M, N \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathrm{dom}(\varphi_1))$ *such that* $\mathrm{fn}(M, N) \cap \mathrm{bn}(\varphi_1) = \emptyset$

$$(M =_{\mathcal{E}} N)\varphi_1 \;\Leftrightarrow\; (M =_{\mathcal{E}} N)\varphi_2.$$

Example 2. Consider the two frames described below:

$$\varphi_1 = \nu a, k.\{w_1 \mapsto enc(a, k)\} \quad \text{and} \quad \varphi_2 = \nu a, k.\{w_1 \mapsto enc(b, k)\}.$$

We have that b and $enc(c, k)$ are deducible from φ_2 in $\mathcal{E}_{mal}$ with recipes b and $mal(w_1, c)$ respectively. We have that $\varphi_1 \not\approx_{\mathcal{E}_{mal}} \varphi_2$ since $(w_1 \neq_{\mathcal{E}_{mal}} mal(w_1, b))\varphi_1$ while $(w_1 =_{\mathcal{E}_{mal}} mal(w_1, b))\varphi_2$. Note that $\varphi_1 \approx_{\mathcal{E}_{enc}} \varphi_2$ (in the theory $\mathcal{E}_{enc}$).

3 Procedures for deduction and static equivalence

In this section we describe our procedures for checking deducibility and static equivalence. After some preliminary definitions, we present the main part of our procedure, i.e. a set of saturation rules used to reach a fixed point. Then, we show how to use this saturation procedure to decide deducibility and static equivalence. The complete proofs of correctness can be found in our research report [8].

Since both problems are undecidable for arbitrary convergent equational theories [1], our saturation procedure does not always terminate. In Section 4, we exhibit (classes of) equational theories for which the saturation terminates.

3.1 Preliminary definitions

The main objects that will be manipulated by our procedure are *facts*, which are either *deduction facts* or *equational facts*.

Definition 3 (facts). *A* deduction fact *(resp. an* equational fact*) is an expression denoted* $[U \rhd u \mid \Delta]$ *(resp.* $[U \sim V | \Delta]$*) where* Δ *is a finite set of the form* $\{X_1 \rhd t_1, \ldots, X_n \rhd t_n\}$ *that contains the* side conditions *of the fact. Moreover, we assume that:*

– $u, t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X})$ *with* $\mathrm{var}(u) \subseteq \mathrm{var}(t_1, \ldots, t_n)$*;*
– $U, V \in \mathcal{T}(\mathcal{F}, \mathcal{N} \cup \mathcal{X} \cup \mathcal{P})$ *and* $X_1, \ldots, X_n$ *are distinct variables;*
– $\mathrm{var}(U, V, X_1, \ldots, X_n) \cap \mathrm{var}(u, t_1, \ldots, t_n) = \emptyset$.

A fact is solved *if* $t_i \in \mathcal{X}$ $(1 \leq i \leq k)$*. Otherwise, it is* unsolved*. A deduction fact is* well-formed *if it is unsolved or if* $u \notin \mathcal{X}$*.*

A fact makes a statement about a frame. We read $[U \rhd u \mid \{X_1 \rhd t_1, \ldots, X_n \rhd t_n\}]$ (resp. $[U \sim V \mid \{X_1 \rhd t_1, \ldots, X_n \rhd t_n\}]$) as "$u$ is deducible with recipe U (resp. U is equal to V) if t_i is deducible with recipe X_i (for all $1 \leq i \leq n$)".

For notational convenience we sometimes omit curly braces for the set of side conditions and write $[U \rhd u \mid X_1 \rhd t_1, \ldots, X_n \rhd t_n]$. When $n = 0$ we simply write $[U \rhd u]$ or $[U \sim V]$. We say that two facts are equivalent if they are equal up to bijective renaming of variables. In the following we implicitly suppose that all operations are carried out modulo the equivalence classes. In particular set union will not add equivalent facts and inclusion will test for equivalent facts. Also, we allow *on-the-fly* renaming of variables in facts to avoid variable clashes.

We now introduce the notion of generation of a term t from a set of facts F. Intuitively, t is generated if it can be syntactically "deduced" from F.

Definition 4 (generation). *Let* F *be a finite set of well-formed deduction facts. A term* t *is* generated by F with recipe R, *written* $\mathsf{F} \vdash^R t$, *if*

1. *either* $t = x \in \mathcal{X}$ *and* $R = x$*;*
2. *or there exist a solved fact* $[R_0 \rhd t_0 \mid X_1 \rhd x_1, \ldots, X_n \rhd x_n] \in \mathsf{F}$*, some terms* R_i *for* $1 \leq i \leq n$ *and a substitution* σ *with* $\mathrm{dom}(\sigma) \subseteq \mathrm{var}(t_0)$ *such that* $t = t_0\sigma$*,* $R = R_0[X_1 \mapsto R_1, \ldots, X_k \mapsto R_k]$*, and* $\mathsf{F} \vdash^{R_i} x_i\sigma$ *for every* $1 \leq i \leq n$*.*

A term t *is* generated by F, *written* $\mathsf{F} \vdash t$, *if there exists* R *such that* $\mathsf{F} \vdash^R t$.

From this definition follows a simple recursive algorithm for effectively deciding whether $\mathsf{F} \vdash t$, providing also the recipe. Termination is ensured by the fact that $|x_i\sigma| < |t|$ for every $1 \leq i \leq n$. Note that using memoization we can obtain an algorithm in polynomial time.

Given a finite set of equational facts E and terms M, N, we write $\mathsf{E} \models M \sim N$ if $M \sim N$ is a consequence, in the usual first order theory of equality, of

$$\{U\sigma \sim V\sigma \mid [U \sim V \mid X_1 \rhd x_1, \ldots, X_k \rhd x_k] \in \mathsf{E}\} \text{ where } \sigma = \{X_i \mapsto x_i\}_{1 \leq i \leq k}.$$

Note that it may be the case that $x_i = x_j$ for $i \neq j$ (whereas $X_i \neq X_j$).

3.2 Saturation procedure

We define for each fact f its *canonical form* f which is obtained by first applying rule (1) and then rule (2) defined below. The idea is to ensure that each variable x_i occurs at most once in the side conditions and to get rid of those variables that do not occur in t. Unsolved deduction facts are kept unchanged.

$$(1) \quad \frac{[R \rhd t \mid X_1 \rhd x_1, \ldots, X_k \rhd x_k] \quad \{i, j\} \subseteq \{1, \ldots, n\} \quad j \neq i \text{ and } x_j = x_i}{[R[X_i \mapsto X_j] \rhd t \mid X_1 \rhd x_1, \ldots, X_{i-1} \rhd x_{i-1}, X_{i+1} \rhd x_{i+1}, \ldots, X_k \rhd x_k]}$$

$$(2) \quad \frac{[R \rhd t \mid X_1 \rhd x_1, \ldots, X_k \rhd x_k] \quad x_i \notin \mathrm{var}(t)}{[R \rhd t \mid X_1 \rhd x_1, \ldots, X_{i-1} \rhd x_{i-1}, X_{i+1} \rhd x_{i+1}, \ldots, X_k \rhd x_k]}$$

A *knowledge base* is a tuple (F, E) where F is a finite set of well-formed deduction facts that are in canonical form and E a finite set of equational facts.

Definition 5 (update). *Given a fact* $\mathsf{f} = [R \rhd t \mid X_1 \rhd t_1, \ldots, X_n \rhd t_n]$ *and a knowledge base* (F, E), the update of (F, E) by f, *written* $(\mathsf{F}, \mathsf{E}) \oplus \mathsf{f}$, *is defined as*

$$\begin{cases} (\mathsf{F} \cup \{\mathsf{f}'\}, \mathsf{E}) & \text{if } \mathsf{f} \text{ is solved and } \mathsf{F} \not\vdash t \quad \textsf{useful fact} \\ \quad \text{where } \mathsf{f}' \text{ is the canonical form of } \mathsf{f} & \\ (\mathsf{F}, \mathsf{E} \cup \{[R' \sim R\{X_i \mapsto t_i\}_{1 \leq i \leq n}\}]) & \text{if } \mathsf{f} \text{ is solved and } \mathsf{F} \vdash t \quad \textsf{useless fact} \\ \quad \text{where } \mathsf{F} \vdash^{R'} t & \\ (\mathsf{F} \cup \{\mathsf{f}\}, \mathsf{E}) & \text{if } \mathsf{f} \text{ is not solved} \quad \textsf{unsolved fact} \end{cases}$$

The choice of the recipe R' in the *useless fact* case is defined by the implementation. While this choice does not influence the correctness of the procedure, it might influence its termination as we will see later. Note that, the result of updating a knowledge base by a (possibly not well-formed and/or not canonical) fact is again a knowledge base. Facts that are not well-formed will be captured by the *useless fact* case, which adds an equational fact.

Initialisation. Given a frame $\varphi = \nu\tilde{n}.\{w_1 \mapsto t_1, \ldots, w_n \mapsto t_n\}$, our procedure starts from an *initial knowledge base* associated to φ and defined as follows:

$$\begin{array}{ll} \mathrm{Init}(\varphi) = & (\emptyset, \emptyset) \\ \bigoplus_{1\leq i\leq n} & [w_i \rhd t_i] \\ \bigoplus_{n\in\mathrm{fn}(\varphi)} & [n \rhd n] \\ \bigoplus_{f\in\mathcal{F}} & [f(X_1,\ldots,X_k) \rhd f(x_1,\ldots,x_k) \mid X_1 \rhd x_1, \ldots \rhd X_k \rhd x_k] \end{array}$$

Example 3. Consider the rewriting system $\mathcal{R}_{\mathcal{E}_{mal}}$ and $\varphi_2 = \nu a, k.\{w_1 \mapsto enc(b,k)\}$. The knowledge base $\mathrm{Init}(\varphi_2)$ is made up of the following deduction facts:

$$\begin{array}{ll} [\, w_1 \rhd enc(b,k) \mid \emptyset] \quad (\mathsf{f}_1) & [\, enc(Y_1,Y_2) \rhd enc(y_1,y_2) \mid Y_1 \rhd y_1, Y_2 \rhd y_2] \quad (\mathsf{f}_3) \\ [\, b \rhd b \mid \emptyset] \quad (\mathsf{f}_2) & [\, dec(Y_1,Y_2) \rhd dec(y_1,y_2) \mid Y_1 \rhd y_1, Y_2 \rhd y_2] \quad (\mathsf{f}_4) \\ & [\, mal(Y_1,Y_2) \rhd mal(y_1,y_2) \mid Y_1 \rhd y_1, Y_2 \rhd y_2] \quad (\mathsf{f}_5) \end{array}$$

Saturation. The main part of our procedure consists in saturating the knowledge base $\mathrm{Init}(\varphi)$ by means of the transformation rules described in Figure 1. The rule Narrowing is designed to apply a rewriting step on an existing deduction fact. Intuitively, this rule allows us to get rid of the equational theory and nevertheless ensure that the generation of deducible terms is complete. The rule F-Solving is used to instantiate an unsolved side condition of an existing deduction fact. Unifying and E-Solving add equational facts which remember when different recipes for a same term exist.

Note that this procedure may not terminate and that the fixed point may not be unique. We write $\Longrightarrow^*$ for the reflexive and transitive closure of $\Longrightarrow$.

Example 4. Continuing Example 3, we illustrate the saturation procedure. We can apply the rule Narrowing on fact f_4 and rewrite rule $dec(enc(x,y),y) \to x$, as well as on fact f_5 and rewrite rule $mal(enc(x,y),z) \to enc(z,y)$ adding facts

$$\begin{array}{ll} [dec(Y_1,Y_2) \rhd x \mid Y_1 \rhd enc(x,y), Y_2 \rhd y] & (\mathsf{f}_6) \\ [mal(Y_1,Y_2) \rhd enc(z,y) \mid Y_1 \rhd enc(x,y), Y_2 \rhd z] & (\mathsf{f}_7) \end{array}$$

The facts f_6 and f_7 are not solved and we can apply the rule F-Solving with f_1 adding the facts:

$$[dec(w_1,Y_2) \rhd b \mid Y_2 \rhd k] \ (\mathsf{f}_8) \qquad [mal(w_1,Y_2) \rhd enc(z,k) \mid Y_2 \rhd z] \ (\mathsf{f}_9)$$

Rule Unifying can be used on facts $\mathsf{f}_1/\mathsf{f}_3$, $\mathsf{f}_3/\mathsf{f}_9$ as well as $\mathsf{f}_1/\mathsf{f}_9$ to add equational facts. This third case allows one to obtain $\mathsf{f}_{10} = [w_1 \sim mal(w_1,Y_2) \mid Y_2 \rhd b]$ which can be solved (using E-Solving with f_2) to obtain $\mathsf{f}_{11} = [w_1 \sim mal(w_1,b)]$. Because of lack of space we do not detail the remaining rule applications. When reaching a fixed point the knowledge base contains the solved facts f_9 and f_{11} as well as those in $\mathrm{Init}(\varphi_2)$.

Narrowing

$$\frac{\mathsf{f} = [M \rhd C[t] \mid X_1 \rhd x_1, \ldots, X_k \rhd x_k] \in \mathsf{F},\ l \to r \in \mathcal{R}_{\mathcal{E}} \quad \text{with } t \notin \mathcal{X},\ \sigma = \mathrm{mgu}(l,t) \text{ and } \mathrm{var}(\mathsf{f}) \cap \mathrm{var}(l) = \emptyset.}{(\mathsf{F},\mathsf{E}) \Longrightarrow (\mathsf{F},\mathsf{E}) \oplus \mathsf{f}_0}$$

where $\mathsf{f}_0 = [M \rhd (C[r])\sigma \mid X_1 \rhd x_1\sigma, \ldots, X_k \rhd x_k\sigma]$.

F-Solving

$$\frac{\mathsf{f}_1 = [M \rhd t \mid X_0 \rhd t_0, \ldots, X_k \rhd t_k],\ \mathsf{f}_2 = [N \rhd s \mid Y_1 \rhd y_1, \ldots, Y_\ell \rhd y_\ell] \in \mathsf{F} \quad \text{with } t_0 \notin \mathcal{X},\ \sigma = \mathrm{mgu}(s,t_0) \text{ and } \mathrm{var}(\mathsf{f}_1) \cap \mathrm{var}(\mathsf{f}_2) = \emptyset.}{(\mathsf{F},\mathsf{E}) \Longrightarrow (\mathsf{F},\mathsf{E}) \oplus \mathsf{f}_0}$$

where $\mathsf{f}_0 = [M\{X_0 \mapsto N\} \rhd t\sigma \mid X_1 \rhd t_1\sigma, \ldots, X_k \rhd t_k\sigma, Y_1 \rhd y_1\sigma, \ldots, Y_\ell \rhd y_\ell\sigma]$.

Unifying

$$\frac{\mathsf{f}_1 = [M \rhd t \mid X_1 \rhd x_1, \ldots, X_k \rhd x_k],\ \mathsf{f}_2 = [N \rhd s \mid Y_1 \rhd y_1, \ldots, Y_\ell \rhd y_\ell] \in \mathsf{F} \quad \text{with } \sigma = \mathrm{mgu}(s,t) \text{ and } \mathrm{var}(\mathsf{f}_1) \cap \mathrm{var}(\mathsf{f}_2) = \emptyset.}{(\mathsf{F},\mathsf{E}) \Longrightarrow (\mathsf{F},\mathsf{E} \cup \{\mathsf{f}_0\})}$$

where $\mathsf{f}_0 = [M \sim N \mid \{X_i \rhd x_i\sigma\}_{1 \leq i \leq k} \cup \{Y_i \rhd y_i\sigma\}_{1 \leq i \leq \ell}]$.

E-Solving

$$\frac{\mathsf{f}_1 = [U \sim V \mid Y \rhd s, X_1 \rhd t_1, \ldots, X_k \rhd t_k] \in \mathsf{E}, \mathsf{f}_2 = [M \rhd t \mid Y_1 \rhd y_1, \ldots, Y_\ell \rhd y_\ell\} \in \mathsf{F} \quad \text{with } s \notin \mathcal{X}, \sigma = \mathrm{mgu}(s,t) \text{ and } \mathrm{var}(\mathsf{f}_1) \cap \mathrm{var}(\mathsf{f}_2) = \emptyset.}{(\mathsf{F},\mathsf{E}) \Longrightarrow (\mathsf{F},\mathsf{E} \cup \{\mathsf{f}_0\})}$$

where $\mathsf{f}_0 = [U\{Y \mapsto M\} \sim V\{Y \mapsto M\} \mid \{X_i \rhd t_i\sigma\}_{1 \leq i \leq k} \cup \{Y_i \rhd y_i\sigma\}_{1 \leq i \leq \ell}]$.

Fig. 1. Saturation rules

3.3 Application to deduction and static equivalence

Procedure for deduction. Let φ be a frame and t be a ground term. The procedure for checking $\varphi \vdash_{\mathcal{E}} t$ runs as follows:

1. Apply the saturation rules to obtain (if any) a saturated knowledge base (F,E) such that $\mathrm{Init}(\varphi) \Longrightarrow^* (\mathsf{F},\mathsf{E})$. Let $\mathsf{F}^+ = \mathsf{F} \cup \{[n \rhd n] \mid n \in \mathrm{fn}(t) \smallsetminus \mathrm{bn}(\varphi)\}$.
2. Return *yes* if there exists N such that $\mathsf{F}^+ \vdash^N t\downarrow_{\mathcal{R}_{\mathcal{E}}}$ (that is, the $\mathcal{R}_{\mathcal{E}}$-normal form of t is generated by F with recipe N); otherwise return *no*.

Example 5. We continue our running example. Let (F,E) be the knowledge base obtained from $\mathrm{Init}(\varphi_2)$ described in Example 4. We show that $\varphi_2 \vdash enc(c,k)$ and $\varphi_2 \vdash b$. Indeed we have that $\mathsf{F} \cup \{[c \rhd c]\} \vdash^{mal(w_1,c)} enc(c,k)$ using facts f_9 and $[c \rhd c]$, and $\mathsf{F} \vdash^b b$ using fact f_2.

Procedure for static equivalence. Let φ_1 and φ_2 be two frames. The procedure for checking $\varphi_1 \approx_{\mathcal{E}} \varphi_2$ runs as follows:

1. Apply the transformation rules to obtain (if possible) two saturated knowledge bases $(\mathsf{F}_i, \mathsf{E}_i)$, $i = 1, 2$ such that $\mathrm{Init}(\varphi_i) \Longrightarrow^* (\mathsf{F}_i, \mathsf{E}_i)$, $i = 1, 2$.
2. For $\{i, j\} = \{1, 2\}$, for every solved fact $[M \sim N \mid X_1 \rhd x_1, \ldots, X_k \rhd x_k]$ in E_i, check if $(M\{X_1 \mapsto x_1, \ldots, X_k \mapsto x_k\} =_{\mathcal{E}} N\{X_1 \mapsto x_1, \ldots, X_k \mapsto x_k\})\varphi_j$.
3. If so return *yes*; otherwise return *no*.

Example 6. Consider again the frames φ_1 and φ_2 which are not statically equivalent (see Example 2). Our procedure answers *no* since $[mal(w_1, b) \sim w_1] \in \mathsf{E}_2$ whereas $(mal(w_1, b) \neq_{\mathcal{E}_{mal}} w_1)\varphi_1$.

4 Termination

As already announced the saturation process does not always terminate.

Example 7. Consider the convergent rewriting system consisting of the single rule $f(g(x)) \to g(h(x))$ and the frame $\phi = \nu a.\{w_1 \mapsto g(a)\}$. We have that

$$\mathrm{Init}(\varphi) \supseteq \{[w_1 \rhd g(a)],\ [f(X) \rhd f(x) \mid X \rhd x]\}.$$

By applying the saturation rules, we generate an infinity of solved facts of the form $[f(\ldots f(w_1)\ldots) \rhd g(h(\ldots h(a)\ldots))]$. Intuitively, this happens because our symbolic representation is unable to express that the function h can be nested an unbounded number of times when it occurs under an application of g.

The same kind of limitation already exists in the procedure implemented in YAPA [5]. However, our symbolic representation, that manipulates terms that are not necessarily ground and facts with side conditions, allows us to go beyond YAPA. We are able for instance to treat equational theories such as malleable encryption and trapdoor commitment.

4.1 Applications

We now give several examples for which the saturation procedure indeed terminates. The complete proofs of termination can be found in our research report [8].

Subterm convergent equational theories. Abadi and Cortier [1] have shown that deduction and static equivalence are decidable for *subterm convergent* equational theories in polynomial time. We retrieve the same results with our algorithm. An equational theory $\mathcal{E}$ is subterm convergent if $\mathcal{R}_{\mathcal{E}}$ is convergent and for every rule $l \to r \in \mathcal{R}_{\mathcal{E}}$, we have that either r is a strict subterm of l, or r is a ground term in $\mathcal{R}_{\mathcal{E}}$-normal form.

Malleable encryption. We obtain also termination for the equational theory $\mathcal{E}_{mal}$ described in Example 1. This is a toy example that does not fall in the class studied in [1]. Indeed, this theory is not *locally stable*: the set of terms in normal form deducible from a frame φ cannot always be obtained by applying public contexts over a finite set (called $sat(\varphi)$ in [1]) of ground terms.

As a witness consider the frame $\varphi_2 = \nu a, k.\{w_1 \mapsto enc(b,k)\}$ introduced in Example 2. Among the terms that are deducible from φ_2, we have those of the form $enc(t,k)$ where t represents any term deducible from φ_2. From this observation, it is easy to see that $\mathcal{E}_{mal}$ is not locally stable.

Our procedure does not have this limitation. A prerequisite for termination is that the set of terms in normal form deducible from a frame is exactly the set of terms obtained by nesting in all possible ways a finite set of contexts. The theory $\mathcal{E}_{mal}$ falls in this class. In particular, for the frame φ_2, our procedure produces the fact $\mathsf{f}_9 = [mal(w_1, Y_2) \rhd enc(z,k) \mid Y_2 \rhd z]$ allowing us to capture all the terms of the form $enc(t,k)$ by the means of a single deduction fact.

Trap-door commitment. The following convergent equational theory $\mathcal{E}_{td}$ is a model for trap-door commitment:

$$open(td(x,y,z),y) = x \qquad td(x_2, f(x_1,y,z,x_2), z) = td(x_1,y,z)$$
$$open(td(x_1,y,z), f(x_1,y,z,x_2)) = x_2 \quad f(x_2, f(x_1,y,z,x_2), z, x_3) = f(x_1,y,z,x_3)$$

As said in introduction, we encountered this equational theory when studying electronic voting protocols. The term $td(m,r,td)$ models the commitment of the message m under the key r using an additional trap-door td. Such a commitment scheme allows a voter who has performed a commitment to open it in different ways using its trap-door. Hence, trap-door bit commitment $td(v,r,td)$ does not bind the voter to the vote v. This is useful to ensure privacy-type properties in e-voting and in particular receipt-freeness [13]. With such a scheme, even if a coercer requires the voter to reveal his commitment, this does not give any useful information to the coercer as the commitment can be viewed as the commitment of any vote (depending on the key that will be used to open it).

For the same reason as $\mathcal{E}_{mal}$, the theory of trap-door commitment described below cannot be handle by the algorithms described in [1, 5].

Termination of our procedure is also ensured for theories such as blind signature and addition as defined in [1].

4.2 Going beyond with fair strategies

In [1] decidability is also shown for an equational theory modeling homomorphic encryption. For our procedure to terminate on this theory we use a particular saturation strategy.

Homomorphic encryption. We consider the theory $\mathcal{E}_{\mathsf{hom}}$ of homomorphic encryption that has been studied in [1, 5].

$$fst(pair(x,y)) = x \quad snd(pair(x,y)) = y \quad dec(enc(x,y),y) = x$$
$$enc(pair(x,y),z) = pair(enc(x,z), enc(y,z))$$
$$dec(pair(x,y),z) = pair(dec(x,z), dec(y,z))$$

In general, our algorithm does not terminate under this equational theory. Consider for instance the frame $\phi = \nu a, b.\{w_1 \mapsto pair(a,b)\}$. We have that:

$$\text{Init}(\varphi) \supseteq \{[w_1 \rhd pair(a,b)],\ [enc(X,Y) \rhd enc(x,y) \ \mid \ X \rhd x,\, Y \rhd y]\}.$$

As in Example 7 we can obtain an unbounded number of solved facts whose projections are of the form:

$$[pair(enc(\ldots enc(a,z_1)\ldots,z_n), enc(\ldots enc(b,z_1)\ldots,z_n)) \mid z_1,\ldots,z_n].$$

However, we can guarantee termination by using a *fair* saturation strategy. We say that a saturation strategy is fair if whenever a rule instance is enabled it will eventually be taken. Indeed in the above example using a fair strategy we will eventually add the facts $[fst(w_1) \rhd a]$ and $[snd(w_1) \rhd b]$. Now the "problematic" facts described above become useless and are not added to the knowledge base anymore. One may note that a fair strategy does not guarantee termination in Example 7 (intuitively, because the function g is one-way and a is not deducible in that example).

5 Conclusion and future work

We have proposed a procedure for the intruder deduction problem and the static equivalence problem for the case in which the cryptographic primitives are modeled by a convergent equational theory. Our procedure terminates for a wide range of equational theories relevant to cryptographic protocols. In particular, we obtain a new decidability result for the theory of trapdoor commitment.

A C++ implementation of the procedures described in this paper is provided in the KiSs (Knowledge In Security protocols) tool [7].

In our implementation, we use a DAG representation of terms and specialized F-Solving and E-Solving rules for solving ground side conditions. Indeed, by checking whether the side condition is generated or not we know whether solving it will eventually produce a solved fact. Note that checking generation can be done in polynomial time. This makes the procedure terminate in polynomial time for subterm convergent equational theories, and the theories $\mathcal{E}_{blind}$, $\mathcal{E}_{mal}$ and $\mathcal{E}_{td}$.

Our procedure terminates on all examples of equational theories presented in [5]. In addition, our tool terminates on the theories $\mathcal{E}_{mal}$ and $\mathcal{E}_{td}$ whereas YAPA does not. In [5] a class of equational theories for which YAPA terminates is identified; it is not known whether our procedure terminates for this class of theories. YAPA may also terminate on examples outside this class. Hence the question of whether termination of our procedures encompasses termination of YAPA is still open.

As directions for future work, we plan to investigate how our technique can be extended to handle equational theories which contain AC operators (such as XOR) and how it can be extended to handle the active case.

References

1. M. Abadi and V. Cortier. Deciding knowledge in security protocols under equational theories. *Theoretical Computer Science*, 387(1-2):2–32, 2006.
2. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *Proc. 28th ACM Symposium on Principles of Programming Languages (POPL'01)*. ACM, 2001.
3. S. Anantharaman, P. Narendran, and M. Rusinowitch. Intruders with caps. In *Proc. 18th International Conference on Term Rewriting and Applications (RTA'07)*, volume 4533 of *LNCS*. Springer, 2007.
4. M. Baudet. YAPA (Yet Another Protocol Analyzer), 2008. http://www.lsv.ens-cachan.fr/~baudet/yapa/index.html.
5. M. Baudet, V. Cortier, and S. Delaune. YAPA: A generic tool for computing intruder knowledge. Research Report LSV-09-03, Laboratoire Spécification et Vérification, ENS Cachan, France, Feb. 2009. 26 pages.
6. Y. Chevalier. *Résolution de problèmes d' accessibilité pour la compilation et la validation de protocoles cryptographiques*. PhD thesis, Université Henri Poincaré, Nancy (France), 2003.
7. Ş. Ciobâcă. KiSs, 2009. http://www.lsv.ens-cachan.fr/~ciobaca/kiss.
8. Ş. Ciobâcă, S. Delaune, and S. Kremer. Computing knowledge in security protocols under convergent equational theories. Research Report LSV-09-05, Laboratoire Spécification et Vérification, ENS Cachan, France, Mar. 2009. 42 pages.
9. V. Cortier and S. Delaune. Deciding knowledge in security protocols for monoidal equational theories. In *Proc. 14th Int. Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'07)*, LNAI. Springer, 2007.
10. V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1):1–43, 2006.
11. S. Delaune, S. Kremer, and M. D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 2008. To appear.
12. P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for the equational theory of Abelian groups with distributive encryption. *Information and Computation*, 205(4):581–623, 2007.
13. T. Okamoto. Receipt-free electronic voting schemes for large scale elections. In *Proc. 5th Int. Security Protocols Workshop*, volume 1361. Springer, 1997.
14. M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions and composed keys is NP-complete. *Theoretical Computer Science*, 299:451–475, 2003.