Chapter 6

Static equivalence

6.1 Definitions and Applications

In the first part we have seen how to verify trace properties such as secrecy, formulated as non-deducibility, and authentication properties in a symbolic model. Now we will investigate *equivalence* properties. These properties formalize the notion of indistinguishability in a symbolic model.

6.1.1 Static equivalence

Remember that messages can be modelled as terms over an abstract algebra: given a set of function symbols $\mathcal{F}$, a set of names $\mathcal{N}$ and a set of variables $\mathcal{X}$ we denote the set of all terms constructed over these sets by $\mathcal{T}(\mathcal{F}, \mathcal{N} \uplus \mathcal{X})$. We consider here a slightly simplified setting where $\mathcal{F}$ only contains public symbols, *i.e.*, $\mathcal{F} = \mathcal{F}_{\mathsf{pub}}$. Moreover, the algebra will be equipped with an equational theory. Let us now be a bit more precise.

Definition 6.1 *An equational theory $\mathcal{E}$ is a set of pairs $\{(M, N) | M, N \in T(\mathcal{F}, \mathcal{N} \uplus \mathcal{X})\}$. We define equality modulo $\mathcal{E}$ denoted $=_\mathcal{E}$ to be the smallest equivalence relation such that*

- *$(M, N) \in \mathcal{E}$ implies that $M =_\mathcal{E} N$,*
- *$=_\mathcal{E}$ is closed under substitutions of terms for variables,*
- *$=_\mathcal{E}$ is closed under application of function symbols, i.e. $M_1 =_\mathcal{E} N_1, \ldots, M_k =_\mathcal{E} N_k$ implies that $\mathsf{f}(M_1, \ldots, M_k) =_\mathcal{E} \mathsf{f}(N_1, \ldots, N_k)$ for all $\mathsf{f} \in \mathcal{F}$ of arity k,*
- *$=_\mathcal{E}$ is closed under bijective renaming of names.*

For convenience in examples we will often just say that $\mathcal{E}$ is defined by the equations

$$M_1 = N_1 \quad \ldots \quad M_n = N_n$$

rather than writing $\mathcal{E} = \{(M_1, N_1), \ldots, (M_n, N_n)\}$. So we will for instance define the equational theory enc which models symmetric encryption and pairings by the equations

$$\mathsf{proj}_1(\langle x, y \rangle) = x \quad \mathsf{proj}_2(\langle x, y \rangle) = y \quad \mathsf{sdec}(\{x\}_y, y) = x.$$

Hence we have that $\mathsf{proj}_2(\langle n, \mathsf{sdec}(\{s\}_k, k)\rangle) =_{\mathsf{enc}} s$.

In Section 3.1.4 we have already introduced static equivalence. Let us rephrase this definition now in this slightly simplified setting. We write $\mathsf{new}\ \overline{n}_1.\sigma_1 =_\alpha \mathsf{new}\ \overline{n}_2.\sigma_2$ when these two frames are the same up to α-conversion, *i.e.* up to a bijective renaming of the bound names. Before defining static equivalence between frames we define what it means for two terms to be equal in a frame. Let $phi = \mathsf{new}\ \overline{n}.\sigma$. We say that $(M =_\mathcal{E} N)\phi$ iff $\phi =_\alpha \mathsf{new}\ \overline{n}'.\sigma'$ for some $\overline{n}'$ and σ' such that $names(M, N) \cap \overline{n}' = \emptyset$ and $M\sigma' =_\mathcal{E} N\sigma'$.

Definition 6.2 *Two frames $\phi_1 = \mathsf{new}\ \overline{n}_1.\sigma_1$ and $\phi_2 = \mathsf{new}\ \overline{n}_2\sigma_2$ are statically equivalent, written $\phi_1 \sim_{\mathcal{E}} \phi_2$ iff $\mathsf{Dom}(\sigma_1) = \mathsf{Dom}(\sigma_2)$ and for all terms M, N we have that $(M =_{\mathcal{E}} N)\phi_1 \Leftrightarrow (M =_{\mathcal{E}} N)\phi_2$.*

Note that it follows directly from the definition that static equivalence is closed under α-conversion.

6.1.2 Applications of static equivalence

We have already seen some uses of static equivalence in Section 3.3. There we used static equivalence to model guessing attacks and to define equivalence properties. We will give here some additional examples.

Password protocols. It is not always possible to rely on previously shared keys or on an existing public key infrastructure. In these cases protocols often rely on shared passwords. However, such passwords are *weak secrets* and can be subject to brute-force attacks. Brute-force attacks on passwords are also refered to as *dictionnary attacks* (refering to a dictionnary containing all passwords) or *guessing attacks* (the search space is small enough that the attacker can guess the password). As we have seen before (Definition 3.3) resistance against offline guessing attacks can be modelled by the means of static equivalence.

Example 6.1 *Let us now consider an example of a password protocol, which indeed resists offline guessing attacks. The EKE protocol [62] can be described by the following 5 steps.*

$$\begin{array}{lll} 1.\ \mathrm{A} \to \mathrm{B}: & \{(\mathsf{pk}(k))\}_w & \textit{(EKE.1)} \\ 2.\ \mathrm{B} \to \mathrm{A}: & \{\mathsf{aenc}(r, \mathsf{pk}(k))\}w & \textit{(EKE.2)} \\ 3.\ \mathrm{A} \to \mathrm{B}: & \{na\}_r & \textit{(EKE.3)} \\ 4.\ \mathrm{B} \to \mathrm{A}: & \{\langle na, nb\rangle\}_r & \textit{(EKE.4)} \\ 5.\ \mathrm{A} \to \mathrm{B}: & \{nb\}_r & \textit{(EKE.5)} \end{array}$$

This protocol uses both ciphers and public key encryption. It does however not rely on a public key infrastructure, i.e. we do not assume that public keys are known and can be associated to a particular identity. In the first step (EKE.1) A *generates a new private key k and sends the corresponding public key* $\mathsf{pk}(k)$ *to* B*, encrypted (using symmetric encryption) with the shared password w. Then,* B *generates a fresh session key r, which he encrypts (using asymmetric encryption) with the previously received public key* $\mathsf{pk}(k)$*. Finally, he encrypts the resulting ciphertext with the password w and sends the result to* A *(EKE.2). The last three steps (EKE.3-5) perform a handshake to avoid replay attacks. One may note that this is a password-only protocol. A new private and public key are used for each session and the only shared secret between different sessions is the password w.*

We use the equational theory EKE *defined by the equations*

$$\mathsf{adec}(\mathsf{aenc}(x, \mathsf{pk}(y)), y) = x \quad \mathsf{sdec}(\mathsf{senc}(x, y), y) = x \quad \mathsf{senc}(\mathsf{sdec}(x, y), y) = x$$
$$\mathsf{proj}_1(\langle x_1, x_2\rangle) = x_1 \qquad \mathsf{proj}_2(\langle x_1, x_2\rangle) = x_2$$

Note the modelling of ciphers with the additional equation $\mathsf{senc}(\mathsf{sdec}(x, y), y) = x$*. The fact that decryption always succeeds is one of the differences between a cipher and a symmetric* *encryption. The protocol can be modeled in our process calculus by* $\mathsf{new}\ w.(P_A \mid\mid P_B)$ *where*

$$\begin{array}{rcl} P_A & = & \mathsf{new}\ k, na. \\ & & \mathsf{out}(c, \{\mathsf{pk}(k)\}_w)). \\ & & \mathsf{in}(c, x_1). \\ & & \mathsf{let}\ ra = \mathsf{adec}(\mathsf{sdec}(x_1, w), k). \\ & & \mathsf{out}(c, \{na\}_{ra}) \\ & & \mathsf{in}(c, x_2). \\ & & \mathsf{if}\ \mathsf{proj}_1(\mathsf{sdec}(x_2, ra)) = na\ \mathsf{then} \\ & & \mathsf{out}(c, \{\mathsf{proj}_2(\mathsf{sdec}(x_2, ra))\}_{ra}) \end{array} \qquad \begin{array}{rcl} P_B & = & \mathsf{new}\ r, nb. \\ & & \mathsf{in}(c, y_1). \\ & & \mathsf{out}(c, \{\mathsf{aenc}(r, \mathsf{sdec}(y_1, w))\}_w). \\ & & \mathsf{in}(c, y_2). \\ & & \mathsf{out}(c, \{\langle \mathsf{sdec}(y_2, r), nb\rangle\}_r). \\ & & \mathsf{in}(c, y_3) \\ & & \mathsf{if}\ \mathsf{sdec}(y_3, r) = nb\ \mathsf{then} \\ & & 0 \end{array}$$

We use the let construction for readability, with the obvious meaning, i.e., $\mathsf{let}\ x = M.P$ *stands for* $P\{^M/_x\}$*. An honest execution of this protocol yields the frame* $\mathsf{new}\ w, k, r, na, nb.\sigma$ *where*

$$\sigma = \{x_1 \mapsto \{\mathsf{pk}(k)\}_w, x_2 \mapsto \{\mathsf{aenc}(r, \mathsf{pk}(k))\}_w, x_3 \mapsto \{na\}_r, x_4 \mapsto \{\langle na, nb\rangle\}_r, x_5 \mapsto \{nb\}_r\}$$

We indeed have that $\mathsf{new}\ w, k, r, na, nb.\sigma \uplus \{x_w \mapsto w\} \sim_{\mathsf{EKE}} \mathsf{new}\ w, w', k, r, na, nb.\sigma \uplus \{x_w \mapsto w'\}$*. Using the tool ProVerif [71] it is possible to show that this equivalence holds for all reachable frames declaring* w *as* weaksecret.

Remark 6.1 *We have used static equivalence to model resistance against guessing attacks. One can note that the same modelization captures* real-or-random *properties in general.*

Anonymity in electronic voting. Consider the following toy protocol for electronic voting where a voter sends his vote to an administrator, encrypted with the administrator's public key and signed with his private key. When the administrator has received all the votes he decrypts them and publishes the result. This can be modelled by the following voter and administrator processes. We consider the same equational theory for asymmetric encryption as before, with the slight difference that aenc will be a ternary function symbol, modelling randomization of the encryption.

$$V = \mathsf{new}\ r; \mathsf{out}(c_1, \mathsf{sign}(\mathsf{aenc}(v, r, \mathsf{pk}(skA)), skV)$$
$$A = \mathsf{in}(c_1, y); \mathsf{out}(c_2, \mathsf{adec}(\mathsf{check}(y, \mathsf{pk}(skV)), skA))$$

To model anonymity we consider two situations each involving two voters. In the first situation V_1 votes v_1 and V_2 votes v_2; in the second situation V_1 votes v_2 and V_2 votes v_1, i.e. the two voters swap their vote. The protocol provides anonymity if these two situations are indistinguishable. Let $\sigma = \{z_1 \mapsto \mathsf{pk}(skA), z_2 \mapsto \mathsf{pk}(skV_1), z_3 \mapsto \mathsf{pk}(skV_2)$ model the knowledge of public keys. The previous scenario can then be modelled by the following two general process:

$$\begin{array}{rl} VP_1 = & \mathsf{new}\ skA, skV_1, skV_2; \\ & (\sigma \| V\{^{skV_1}/_{skV}\}\{^{v_1}/_v\}\{^{r_1}/_r\} \| V\{^{skV_2}/_{skV}\}\{^{v_2}/_v\}\{^{r_2}/_r\} \| A\{^{skV_1}/_{skV}\} \| A\{^{skV_2}/_{skV}\}) \\ VP_2 = & \mathsf{new}\ skA, skV_1, skV_2; \\ & (\sigma \| V\{^{skV_1}/_{skV}\}\{^{v_2}/_v\}\{^{r_1}/_r\} \| V\{^{skV_2}/_{skV}\}\{^{v_1}/_v\}\{^{r_2}/_r\} \| A\{^{skV_1}/_{skV}\} \| A\{^{skV_2}/_{skV}\}) \end{array}$$

Two frames which can be derived from this protocol are

$$\begin{array}{rl} \phi_1 = & \mathsf{new}\ skA, skV_1, skV_2, v_1, r_1, v_2, r_2; \sigma \uplus \\ & \{x_1 \mapsto \mathsf{sign}(\mathsf{aenc}(v_1, r_1, \mathsf{pk}(skA)), skV_1); x_2 \mapsto \mathsf{sign}(\mathsf{aenc}(v_2, r_2, \mathsf{pk}(skA)), skV_2; \\ & \ x_3 \mapsto v_1; x_4 \mapsto v_2\} \end{array}$$

$$\begin{array}{rl} \phi_2 = & \mathsf{new}\ skA, skV_1, skV_2, v_1, r_1, v_2, r_2; \sigma \uplus \\ & \{x_1 \mapsto \mathsf{sign}(\mathsf{aenc}(v_2, r_1, \mathsf{pk}(skA)), skV_1); x_2 \mapsto \mathsf{sign}(\mathsf{aenc}(v_1, r_2, \mathsf{pk}(skA)), skV_2; \\ & \ x_3 \mapsto v_1; x_4 \mapsto v_2\} \end{array}$$

and anonymity can be modelled as $\phi_1 \sim_{\mathcal{E}} \phi_2$.

6.1.3 Some properties of static equivalence

We will now show some properties of static equivalence.

Proposition 6.1

$$\mathsf{new}\ \overline{n}_1.\sigma_1 \sim_{\mathcal{E}} \mathsf{new}\ \overline{n}_2.\sigma_2 \textit{ iff } \mathsf{new}\ \overline{n}_1 \uplus \overline{m}.(\sigma_1 \uplus \theta) \sim_{\mathcal{E}} \mathsf{new}\ \overline{n}_2 \uplus \overline{m}.(\sigma_2 \uplus \theta)$$

where $\overline{m} \cap (names(\sigma_1) \cup names(\sigma_2)) = \emptyset$ *and* $(\overline{n}_1 \cup \overline{n}_2) \cap names(\theta) = \emptyset$.

A direct corollary of this proposition is the following.

Corollary 6.1

$$\mathsf{new}\ \overline{n}_1.\sigma_1 \sim_{\mathcal{E}} \mathsf{new}\ \overline{n}_2.\sigma_2 \textit{ and } \mathsf{new}\ \overline{m}_1.\theta_1 \sim_{\mathcal{E}} \mathsf{new}\ \overline{m}_2.\theta_2$$
$$\textit{iff}$$
$$\mathsf{new}\ \overline{n}_1 \uplus \overline{m}_1.(\sigma_1 \uplus \theta_1) \sim_{\mathcal{E}} \mathsf{new}\ \overline{n}_2 \uplus \overline{m}_2.(\sigma_2 \uplus \theta_2)$$

where $\overline{m}_i \cap names(\sigma_i) = \emptyset$ *and* $\overline{n}_i \cap names(\theta_i) = \emptyset$ $(i \in \{1,2\})$.

The $\Rightarrow$ direction can be seen as a composition result provided the frames do not share any secret name. The $\Leftarrow$ direction states that whenever two frames are statically equivalent then any subframe is also statically equivalent.

When proving results on static equivalence it is often more convenient to use the following equivalent definition of static equivalence.

Definition 6.3 *Two frames* $\phi_1 = \mathsf{new}\ \overline{n}.\sigma_1$ *and* $\phi_2 = \mathsf{new}\ \overline{n}.\sigma_2$ *(having the same set of restricted names) are statically equivalent, written* $\phi_1 \sim_{\mathcal{E}} \phi_2$ *iff* $\mathsf{Dom}(\sigma_1) = \mathsf{Dom}(\sigma_2)$ *and for all terms* M, N*, such that* $(names(M) \cup names(N)) \cap \overline{n} = \emptyset$ *we have that* $M\sigma_1 =_{\mathcal{E}} N\sigma_1 \Leftrightarrow M\sigma_2 =_{\mathcal{E}} N\sigma_2$.

We leave it as an exercise to show that these two definitions are equivalent.

Proof : [of Proposition 6.1]
$\boxed{\Rightarrow}$ We will show the contrapositive. Suppose that $\mathsf{new}\ \overline{n}_1 \uplus \overline{m}.(\sigma_1 \uplus \theta) \not\sim_{\mathcal{E}} \mathsf{new}\ \overline{n}_2 \uplus \overline{m}.(\sigma_2 \uplus \theta)$. Hence there exist two terms M, N such that $names(M,N) \cap (\overline{n}_1 \cup \overline{n}_2 \cup \overline{m}) = \emptyset$ and $M(\sigma_1 \uplus \theta) =_{\mathcal{E}} N(\sigma_1 \uplus \theta)$ and $M(\sigma_2 \uplus \theta) \neq_{\mathcal{E}} N(\sigma_2 \uplus \theta)$ (or vice-versa). As σ_i are ground $M(\sigma_i \uplus \theta) = (M\theta)\sigma_i$ and $N(\sigma_i \uplus \theta) = (N\theta)\sigma_i$. Because $names(M,N) \cap (\overline{n}_1 \cup \overline{n}_2) = \emptyset$ and $(\overline{n}_1 \cup \overline{n}_2) \cap names(\theta) = \emptyset$ we have that $names(M\theta, N\theta) \cap (\overline{n}_1 \cup \overline{n}_2) = \emptyset$. Hence, $M\theta, N\theta$ gives a valid test to distinguish $\mathsf{new}\ \overline{n}_1.\sigma_1$ and $\mathsf{new}\ \overline{n}_2.\sigma_2$.

$\boxed{\Leftarrow}$ We again show the contrapositive. Suppose that $\mathsf{new}\ \overline{n}_1.\sigma_1 \not\sim_{\mathcal{E}} \mathsf{new}\ \overline{n}_2.\sigma_2$. Hence there exist two terms M, N such that $names(M,N) \cap (\overline{n}_1 \cup \overline{n}_2) = \emptyset$ and $M\sigma_1 =_{\mathcal{E}} N\sigma_1$ and $M\sigma_2 \neq_{\mathcal{E}} N\sigma_2$ (or vice-versa). Let M' and N' be two terms obtained from M and N by applying a bijective renaming of names in $names(M,N) \cap \overline{m}$ and variables in $\mathsf{Dom}(\theta)$ by fresh names. Hence, $names(M',N') \cap (\overline{m} \cup \overline{n}_1 \cup \overline{n}_2) = \emptyset$ and $vars(M',N') \cap \mathsf{Dom}(\theta) = \emptyset$. Because $=_{\mathcal{E}}$ and $\neq_{\mathcal{E}}$ are closed under bijective renaming and $\overline{m} \cap (names(\sigma_1) \cup names(\sigma_2)) = \emptyset$ we have that $M'\sigma_1 =_{\mathcal{E}} N'\sigma_1$ and $M'\sigma_2 \neq_{\mathcal{E}} N'\sigma_2$. As $vars(M',N') \cap \mathsf{Dom}(\theta) = \emptyset$ and σ_i are ground we obtain that $M'(\sigma_1 \uplus \theta) =_{\mathcal{E}} N'(\sigma_1 \uplus \theta)$ and $M'(\sigma_2 \uplus \theta) \neq_{\mathcal{E}} N'(\sigma_2 \uplus \theta)$. □

While composition with shared secrets does not hold in general, resistance against offline guessing attacks (in the presence of a passive attacker) composes when the same password is used.

Proposition 6.2

$$\begin{array}{ll} & \mathsf{new}\ w.\mathsf{new}\ \overline{n}_1.(\sigma_1 \uplus \{x \mapsto w\}) \sim_{\mathcal{E}} \mathsf{new}\ w.\mathsf{new}\ w'.\mathsf{new}\ \overline{n}_1.(\sigma_1 \uplus \{x \mapsto w'\}) \\ \textit{and} & \mathsf{new}\ w.\mathsf{new}\ \overline{n}_2.(\sigma_2 \uplus \{x \mapsto w\}) \sim_{\mathcal{E}} \mathsf{new}\ w.\mathsf{new}\ w'.\mathsf{new}\ \overline{n}_2.(\sigma_2 \uplus \{x \mapsto w'\}) \end{array}$$
$$\textit{implies that}$$
$$\mathsf{new}\ w.\mathsf{new}\ \overline{n}_1 \uplus \overline{n}_2.(\sigma_1 \uplus \sigma_2 \uplus \{x \mapsto w\}) \sim_{\mathcal{E}} \mathsf{new}\ w.\mathsf{new}\ w'.\mathsf{new}\ n_1 \uplus \overline{n}_2.(\sigma_1 \uplus \sigma_2 \uplus \{x \mapsto w'\})$$

where $w' \notin names(\sigma_i)$ *and* $\overline{n}_1 \cap names(\sigma_2) = \overline{n}_2 \cap names(\sigma_1) = \emptyset$.

Exercise 30 will guide us through the proof of this proposition.

6.1.4 Further readings

This section does not give an exhaustive overview of all papers on the subject. The aim is to give some pointers to the interested reader to papers that are closely related (using similar techniques and notations) to the subjects covered in this chapter.

Static equivalence was first introduced in the applied pi calculus [7] which is a process calculus for cryptographic proptocols. In this context it was used to characterize observational equivalence by a labelled bisimulation, relying on static equivalence. Many properties of static equivalence were already introduced in that paper. Deducibility and static equivalence for equational theories are also discussed in detail in [6]. This paper also presents decidability and complexity results, which we will cover in the next section.

Guessing attacks were first formalized using static equivalence in [115] (other previous formalizations not based on static equivalence were presented in [197, 135] but they seem less natural). Composition results for guessing attacks were shown in [136]: this paper includes Proposition 6.2 showing that offline guessing attacks do compose in the presence of a passive adversary, it is shown that offline guessing attacks do not compose in general in the presence of an active adversary, but sufficient conditions are given to achieve this composition.

The modelling of anonymity by the use of static and above all observational equivalence has been adressed for electronic voting in [182, 138, 28]. Anonymity in a different context is studied in [149]: there it is shown how to model anonymity in authentication protocols by the use of observational equivalence.

6.2 Procedure for subterm convergent equational theories

In this chapter we show that for a large class of equational theories, namely *subterm convergent* equational theories, static equivalence can be decided in polynomial time. This part follows closely the paper by Abadi and Cortier [6]. Other results and more practical procedures will be discussed in the further readings section.

6.2.1 Preliminaries

Let us first introduce some preliminary definitions and notations.

Notations. Given a set of function symbols $\mathcal{F}$ we write $\mathsf{ar}(\mathcal{F})$ for the maximal arity of $f \in \mathcal{F}$. Sometimes in this section, we will write $==$ when we want to emphasize that we mean syntactic equality. For a term t we define $|t|$ to be its size as follows: $|t| = 1$ if t is a variable, a name or a constant and $\mathsf{f}(t_1, \ldots, t_n) = 1 + \sum_{1 \leq i \leq n} |t_i|$. We denote by $st(t)$ the set of all subterms of t and by $pos(t)$ the set of its positions. Then, for $p \in pos(t)$ we write $t|_p$ for the subterm of t at position p. $t[u]_p$ is the term obtained by replacing the subterm at position p by u. The notions of size and subterms are naturally lifted to frames: for $\varphi = \mathsf{new}\ \overline{n}.\{x_1 \mapsto M_1, \ldots x_n \mapsto M_n\}$ we define $|\varphi| = \sum_{1 \leq i \leq n} |M_i|$ and $st(\varphi) = \cup_{1 \leq i \leq n} st(M_i)$.

Rewrite systems and equational theories. A rewrite system $\mathcal{R}$ is a set of rewrite rules $\ell \to r$ such that $vars(r) \subseteq vars(\ell)$. We say that a term t rewrites to u by $\mathcal{R}$ if there exists $\ell \to r \in \mathcal{R}$ and $p \in pos(t)$ such that $t|_p = \ell\sigma$ for some substitution σ and $u = t[r\sigma]_p$. A rewrite system $\mathcal{R}$ is terminating if there is no infinite chain $t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} \ldots$. A rewrite system $\mathcal{R}$ is confluent if for any t_1, t_2 such that $t \to^*_{\mathcal{R}} t_1$, $t \to_{\mathcal{R}} t_2$ there exists u such that $t_1 \to^*_{\mathcal{R}} u$ and $t_2 \to^*_{\mathcal{R}} u$. $\mathcal{R}$ is convergent if it is both confluent and terminating. For a convergent rewrite system $\mathcal{R}$ we denote by $t\downarrow_{\mathcal{R}}$ the unique normal form of t. A rewrite system is subterm convergent if for any rule $\ell \to r \in \mathcal{R}$ we have that r is either a strict subterm of ℓ or r is a constant. Given an equational theory $\mathcal{E}$ we associate a rewrite system $\mathcal{R}_{\mathcal{E}}$ to it by orienting the equations in $\mathcal{E}$. For readability we generally write $\to_{\mathcal{E}}$ instead of $\to_{\mathcal{R}_{\mathcal{E}}}$. We say that an equational theory is subterm convergent if it can be oriented to a subterm convergent rewrite system.

Example 6.2 *The equational theories* enc*,* cipher *and* EKE *encountered before are subterm convergent. However, the equational theories for homomorphic encryption* homo*, extending* enc *by the equations*

$$\{\langle x,y\rangle\}_z = \langle\{x\}_z,\{y\}_z\rangle \qquad \mathsf{sdec}(\langle x,y\rangle,z) = \langle\mathsf{sdec}(x,z),\mathsf{sdec}(y,z)\rangle$$

and for blind signatures blind *defined by*

$$\begin{array}{rcl}\mathsf{check}(\mathsf{sign}(x,y),\mathsf{pk}(y)) & = & x\\ \mathsf{unblind}(\mathsf{blind}(x,y),y) & = & x\\ \mathsf{unblind}(\mathsf{sign}(\mathsf{blind}(x,y),z),y) & = & \mathsf{sign}(x,z)\end{array}$$

are not subterm convergent.

DAG **representation of terms.** In order to achieve a polynomial time complexity we will use a DAG representation of terms (which can be exponentially more succint than its tree representation). A DAG representation of a term is a directed acyclic graph (V,l,E,v_0) where

- V is the set of vertices;
- $l : V \to \mathcal{F}\cup\mathcal{N}\cup\mathcal{X}$ is a labelling function;
- $E \subseteq (V\times\{1..\mathsf{ar}(\mathcal{F})\})\times V$ is the edge relation;
- v_0 is the root vertex.

For each vertex $v\in V$ such that $l(v) = \mathsf{f}\in\mathcal{F}$, v has exactly $\mathsf{ar}(\mathsf{f})$ outgoing edges labelled by 1 to $\mathsf{ar}(f)$. Vertices labelled by names and variables have no outgoing edge. We denote by $E(v,i)$ the unique vertex v' such that $(v,i,v')\in E$. A DAG $D = (V,l,E,v_0)$ defines a term $t(D)$ as $t(D) = \mathsf{f}(t(V,l,E,E(v_0,1)),\ldots,t(V,l,E,E(v_0,\mathsf{ar}(\mathsf{f}))))$ if $l(v_0) = f\in\mathcal{F}$ and $t(D) = l(v_0)$ if $l(v_0)\in\mathcal{N}\cup\mathcal{X}$. We say that a DAG representation (V,l,E,v_0) is minimal if there are no two distinct vertices $v_1,v_2\in V$ such that $t(V,l,E,v_1) = t(V,l,E,v_2)$.

The size of a DAG D denoted $|D|$ is the number of its vertices. We define the DAG size of a term t, denoted $|t|_{\mathsf{DAG}}$, as $|st(t)|$ which coincides with $|D|$ when D is a minimal DAG representation of t.

Given a DAG representation D we can compute its minimal DAG representation in $\mathcal{O}(|D|^3)$. We check whether there are two vertices v_1,v_2 (at most $|D|^2$ possibilities) such that $l(v_1) = l(v_2)$ and $E(v_1,i) = E(v_2,i)$ for $1\le i\le \mathsf{ar}(l(v_1))$. In that case we delete v_2 from V and replace any occurence of v_2 by v_1. We iterate this at most $|D|$ times.

Hence, we can also check whether $t(D_1) == t(D_2)$ in polynomial time in $|D_1|+|D_2|$.

Given a subterm convergent rewrite system $\mathcal{R}$ and a minimal DAG representation $D = (V,l,E,v_0)$ of a term t we can compute the minimal DAG representation of $t\downarrow_{\mathcal{R}}$ in $\mathcal{O}(|D|^4)$. To see this note that each rewrite rule is of the form $C[x_1,\ldots,x_n]\to C'[x_1,\ldots,x_n]$ or $C[x_1,\ldots,x_n]\to c$. Starting from the root we check whether the rewrite rule applies in D (at most $|C||D|$ tests). If it applies to some vertex v, i.e. $t(V,l,E,v) = C[x_1,\ldots,x_n]\theta$ for some θ, replace v by the vertex representing $C'[x_1,\ldots,x_n]\theta$ (if it is not a constant this vertex exists because it is a subterm; otherwise add a vertex labelled by the constant c). Then minimize the DAG in $\mathcal{O}(|D|^3)$. At each step (except for a constant number of cases) we delete one vertex. Hence the procedure stops after at most $|D|$ iterations and we compute the DAG representation of $t\downarrow_{\mathcal{R}}$ in $\mathcal{O}(|D|^4)$.

6.2.2 Deciding $\sim_{\mathcal{E}}$ in polynomial time for subterm convergent equational theories

Let $\mathcal{E}$ be a subterm convergent equational theory defined by the axioms $\cup_{1 \leq i \leq n}\{(\ell_i, r_i)\}$. We define the theory constant $c_{\mathcal{E}} = \max_{1 \leq i \leq n}(|\ell_i|, \mathsf{ar}(\mathcal{F}) + 1)$. By convention, we define $c_{\mathcal{E}}$ to be 1 in the case where $\mathcal{E}$ and $\mathcal{F}$ are empty.

Theorem 6.1 *Let φ and φ' be two frames. We can decide whether $\varphi \sim_{\mathcal{E}} \varphi'$ in polynomial time in $|\varphi| + |\varphi'|$.*

The remaining of this section will be dedicated to the proof of this result. The proof can be split into 3 steps:

1. frame saturation,
2. caracterization of a frame by a finite set of equalities,
3. decidability of $\sim_{\mathcal{E}}$.

We now detail each of these steps.

6.2.2.1 Frame saturation

For each frame φ we can compute a set of terms $\mathsf{sat}(\varphi)$. This set contains all terms that are deducible from the frame by applying only "small" contexts.

Definition 6.4 *Given a frame $\varphi = \mathsf{new}\ \overline{n}.\{x_1 \mapsto M_1, \ldots, x_k \mapsto M_k\}$ we define $\mathsf{sat}(\varphi)$ to be the smallest set such that*

1. *$\{M_1, \ldots, M_k\} \subseteq \mathsf{sat}(\varphi)$,*
2. *if $M_1, \ldots, M_n \in \mathsf{sat}(\varphi)$ and $\mathsf{f}(M_1, \ldots, M_n) \in st(\varphi)$ then $\mathsf{f}(M_1, \ldots, M_n) \in \mathsf{sat}(\varphi)$,*
3. *if $M_1, \ldots, M_n \in \mathsf{sat}(\varphi)$ and C is a context such that $C[M_1, \ldots, M_n] \to_{\mathcal{E}} M$ and $|C| \leq c_{\mathcal{E}}$, $names(C) \cap \overline{n} = \emptyset$ and $M \in st(\varphi)$ then $M \in \mathsf{sat}(\varphi)$.*

Example 6.3 *We again consider the equational theory* enc *which can be oriented into the following subterm convergent equational theory:*

$$\begin{array}{rl} \mathsf{sdec}(\{x\}_y, y) \to_{\mathsf{enc}} x & \\ \mathsf{proj}_i(\langle x_1, x_2\rangle) \to_{\mathsf{enc}} x_i & \qquad (i \in \{1,2\}) \end{array}$$

We have that $c_{\mathsf{enc}} = 5$. Consider the frame

$$\varphi = \mathsf{new}\ s_1, s_2, k_1, k_2.\{x_1 \mapsto \{\langle s_1, s_2\rangle\}_{\langle k_1, k_2\rangle}, x_2 \mapsto k_1, x_3 \mapsto k_2\}$$

We have that $\mathsf{sat}(\varphi) = \{\{\langle s_1, s_2\rangle\}_{\langle s_1, s_2\rangle}, k_1, k_2, \langle k_1, k_2\rangle, \langle s_1, s_2\rangle, s_1, s_2\}$. The first three terms result directly from the frame (rule 1). The term $\langle k_1, k_2\rangle$ can be added by applying the pairing function symbol (rule 2). Term $\langle s_1, s_2\rangle$ can be added in two different ways using rule 3: either apply the context $\mathsf{sdec}(\_, \_)$ on $\{\langle s_1, s_2\rangle\}_{\langle s_1, s_2\rangle}$ and $\langle k_1, k_2\rangle$ or $\mathsf{sdec}(\_, \langle \_, \_\rangle)$ on $\{\langle s_1, s_2\rangle\}_{\langle k_1, k_2\rangle}$, k_1 and k_2. s_i can be added by applying $\mathsf{proj}_i(\_)$ on $\langle s_1, s_2\rangle$. Note that we are not allowed to directly use the context $C = \mathsf{proj}_i(\mathsf{sdec}(\_, \langle \_, \_\rangle))$ (corresponding to the actual recipe for deducing s_i directly from φ) because $|C| > 5$.

We now show that this set can be computed in polynomial time in $|\varphi|$

Proposition 6.3 *The set $\mathsf{sat}(\varphi)$ can be computed in $\mathcal{O}(|\varphi|^{\max(\mathsf{ar}(\mathcal{F}), c_{\mathcal{E}})+2})$.*

Proof : We initialize the set with the elements $\{M_1, \ldots, M_k\}$ (rule 1) and then saturate the set using rules 2 and 3. We notice that $\mathsf{sat}(\varphi) \subseteq st(\varphi)$. Hence, we have that $\mathsf{sat}(\varphi)$ contains at most $|\varphi|$ elements and the saturation will take at most $|\varphi|$ steps.

- If the step is an application of rule 2 we have to construct at most $|\mathcal{F}||\varphi|^{\mathsf{ar}(\mathcal{F})}$ terms. For each of these terms we check whether it is in $\mathsf{sat}(\varphi)$ which can be done in linear time in $|\varphi|$. Hence, this step can be computed in $\mathcal{O}(|\varphi|^{\mathsf{ar}(\mathcal{F})+1})$.
- If the step is an application of rule 3 we have to compare whether any context of size $\leq c_{\mathcal{E}}$ applied to some M_is in $\mathsf{sat}(\varphi)$ is an instance of the lhs of a rewrite rule and check whether the resulting term is in $\mathsf{sat}(\varphi)$. This can be computed in $\mathcal{O}(|\varphi|^{c_{\mathcal{E}}+1})$. To see this note that the number of contexts is constant (as $c_{\mathcal{E}}$ is fixed) and that a context has at most $c_{\mathcal{E}}$ holes. Hence, we need to consider at most $\mathcal{O}(|\varphi|^{c_{\mathcal{E}}})$ terms for which we need to check whether they are in $\mathsf{sat}(\varphi)$.

Hence each step can be computed in $\mathcal{O}(|\varphi|^{\max(\mathsf{ar}(\mathcal{F}),c_{\mathcal{E}})+1})$. As there are at most $|\varphi|$ steps we can compute $\mathsf{sat}(\varphi)$ in $\mathcal{O}(|\varphi|^{\max(\mathsf{ar}(\mathcal{F}),c_{\mathcal{E}})+2})$.

Moreover any element of $\mathsf{sat}(\varphi)$ can be deduced using a recipe of small DAG size.

Proposition 6.4 *Let $\varphi = \mathsf{new}\ \overline{n}.\sigma$. For all $M \in \mathsf{sat}(\varphi)$ there exists a recipe R_M such that $|R_M|_{\mathsf{DAG}} \leq c_{\mathcal{E}} \cdot |\varphi|$, names$(R_M) \cap \overline{n} = \emptyset$ and $R_M\sigma =_{\mathcal{E}} M$.*

Proof : For each term $M \in \mathsf{sat}(\varphi)$ we can give a recipe R_M according to the rule of Definition 6.4 which added M to $\mathsf{sat}(\varphi)$.

1. Let $R_M = x_i$ if $x_i\sigma = M$ and M is added by rule 1.
2. Let $R_M = \mathsf{f}(R_{M_1}, \ldots R_{M_n})$ if M is added by rule 2.
3. Let $R_M = C(R_{M_1}, \ldots R_{M_n})$ if M is added by rule 3.

We can now construct a graph which contains each DAG corresponding to R_M for $M \in \mathsf{sat}(\varphi)$. The graph is constructed as follows.

1. For each recipe R_M constructed by rule 1 add a vertex labelled by x_i.
2. For each recipe $R_M = \mathsf{f}(R_{M_1}, \ldots R_{M_n})$ constructed by rule 2 add a vertex labelled by f and connect it to the vertices corresponding to the terms $R_{M_1}, \ldots R_{M_n}$.
3. For each recipe $R_M = C(R_{M_1}, \ldots R_{M_n})$ constructed by rule 3 construct the minimal DAG corresponding to C and connect it to the vertices corresponding to the terms $R_{M_1}, \ldots R_{M_n}$.

Steps 1 and 2 add 1 vertex to the graph. As $|C| \leq c_{\mathcal{E}}$ each step 3 adds at most $c_{\mathcal{E}}$ vertices to the graph. Moreover, we know that the graph can be constructed in $|\varphi|$ steps. Hence its size is bounded by $c_{\mathcal{E}} \cdot |\varphi|$.

Example 6.4 *Continuing Example 6.3, let*

$$\begin{array}{rcl} M_1 & = & \{\langle s_1, s_2\rangle\}_{\langle k_1,k_2\rangle} \\ M_2 & = & k_1 \\ M_3 & = & k_2 \end{array} \qquad \begin{array}{rcl} M_4 & = & \langle k_1, k_2\rangle \\ M_5 & = & \langle s_1, s_2\rangle \\ M_6 & = & s_1 \\ M_7 & = & s_2 \end{array}$$

denote the elements of $\mathsf{sat}(\varphi)$*. We can define the following recipes*

$$\begin{array}{rcl} R_{M_i} & = & x_i \qquad (1 \leq i \leq 3) \\ R_{M_4} & = & \langle x_2, x_3\rangle \end{array} \qquad \begin{array}{rcl} R_{M_5} & = & \mathsf{sdec}(x_1, R_{M_4}) \\ R_{M_6} & = & \mathsf{proj}_1(R_{M_5}) \\ R_{M_7} & = & \mathsf{proj}_2(R_{M_6}) \end{array}$$

6.2.2.2 Caracterization of a frame by a finite set of equalities

To each frame φ, we can associate a set $\mathsf{Eq}(\varphi)$ which is finite (up to renaming) and caracterizes all equalities which hold in φ.

Definition 6.5 *Given a frame φ we define $\mathsf{Eq}(\varphi)$ to be the set of equalities*

$$C_1[R_{M_1}, \dots R_{M_k}] = C_2[R_{N_1}, \dots R_{N_\ell}]$$

such that $|C_1|, |C_2| \leq c_{\mathcal{E}}$, $M_{1\leq i\leq k}, N_{1\leq j\leq \ell} \in \mathsf{sat}(\varphi)$ and $(C_1[R_{M_1}, \dots R_{M_k}] =_{\mathcal{E}} C_2[R_{N_1}, \dots R_{N_\ell}])\varphi$. We write $\varphi' \models \mathsf{Eq}(\varphi)$ iff $(M =_{\mathcal{E}} N)\varphi'$ for all $(M = N) \in \mathsf{Eq}(\varphi)$.

Example 6.5 *Let us continue with Example 6.4. Omitting redundant and trivial equations we have that $\mathsf{Eq}(\varphi) = \{(\{R_{M_5}\}_{R_{M_4}} = R_{M_1}), \langle R_{M_6}, R_{M_7}\rangle = R_{M_5})\}$. Intuitively these equalities state that M_1 is indeed an encryption with the key M_4 and M_1 is the encryption of a pair.*

We now show two crucial properties of $\mathsf{Eq}(\varphi)$.

Lemma 6.1 *Let $\varphi = \mathsf{new}\ \overline{n}.\sigma$ and φ' be two frames such that $\varphi' \models \mathsf{Eq}(\varphi)$. For all contexts C_1 and C_2 such that names$(C_1, C_2) \cap \overline{n} = \emptyset$ and for all terms $M_{1\leq i\leq k}, N_{1\leq j\leq \ell} \in \mathsf{sat}(\varphi)$ if $C_1[M_1, \dots, M_k] == C_2[N_1, \dots, N_\ell]$ then $(C_1[R_{M_1}, \dots, R_{M_k}] =_{\mathcal{E}} C_2[R_{N_1}, \dots, R_{N_\ell}])\varphi'$.*

Proof : The proof is done by induction on $|C_1| + |C_2|$.

Base case. $|C_1|, |C_2| \leq c_{\mathcal{E}}$. We have that

$$C_1[M_1, \dots, M_k] == C_2[N_1, \dots, N_\ell]$$

As $M_i, N_j \in \mathsf{sat}(\varphi)$ we have by Proposition 6.4 that R_{M_i}, resp. R_{N_j}, are recipes for M_i, resp. N_j, in φ. Hence,

$$(C_1[R_{M_1}, \dots, R_{M_k}] =_{\mathcal{E}} C_2[R_{N_1}, \dots, R_{N_\ell}])\varphi$$

As $|C_1|, |C_2| \leq c_{\mathcal{E}}$, we have that $(C_1[R_{M_1}, \dots, R_{M_k}] = C_2[R_{N_1}, \dots, R_{N_\ell}]) \in \mathsf{Eq}(\varphi)$ and hence

$$(C_1[R_{M_1}, \dots, R_{M_k}] =_{\mathcal{E}} C_2[R_{N_1}, \dots, R_{N_\ell}])\varphi'.$$

Inductive case. We consider two cases.

- $C_1 \neq \_$ and $C_2 \neq \_$.
  In that case, we have that $C_1 == \mathsf{f}(C_1^1, \dots C_1^n)$ and $C_2 == \mathsf{f}(C_2^1, \dots C_2^n)$ and for $1 \leq i \leq n$

  $$C_1^i[M_1, \dots, M_k] == C_2^i[N_1, \dots, N_\ell].$$

  By induction hyptothesis, we have that $(C_1^i[R_{M_1}, \dots, R_{M_k}] =_{\mathcal{E}} C_2^i[R_{N_1}, \dots, R_{N_\ell}])\varphi'$. Hence, as $=_{\mathcal{E}}$ is closed under application of function symbols we also have that

  $$(C_1[R_{M_1}, \dots, R_{M_k}] =_{\mathcal{E}} C_2[R_{N_1}, \dots, R_{N_\ell}])\varphi'.$$

- Either $C_1 == \_$ or $C_2 == \_$. Let us suppose that $C_1 == \mathsf{f}(C_1^1, \dots C_1^n)$ and $C_2 == \_$.
  Let $M, M_1, \dots, M_k \in \mathsf{sat}(\varphi)$ such that $C_1[M_1, \dots, M_k] == M$. Hence we have

  $$\mathsf{f}(C_1^1[M_1, \dots, M_k], \dots, C_1^n[M_1, \dots, M_k]) == M.$$

  Let $N_i = C_1^i[M_1, \dots, M_k]$ for $1 \leq i \leq n$. As $M \in \mathsf{sat}(\varphi)$ and $N_i \in st(M)$ we have that $N_i \in st(\mathsf{sat}(\varphi))$. Applying iteratively rule 2 of Definition 6.4 we obtain that $N_i \in \mathsf{sat}(\varphi)$. We can apply the induction hypothesis on $N_i == C_1^i[M_1, \dots, M_k]$ and obtain that $(R_{N_i} =_{\mathcal{E}} C_1^i[R_{M_1},\dots,R_{M_k}])\varphi'$. Moreover, $M == \mathsf{f}(N_1,\dots,N_n)$. Applying the base case (with contexts $\_$ and $\mathsf{f}(\_,\dots,\_)$) we obtain that $(R_M =_{\mathcal{E}} \mathsf{f}(R_{N_1},\dots,R_{N_n}))\varphi'$. From $(R_{N_i} =_{\mathcal{E}} C_1^i[R_{M_1},\dots,R_{M_k}])\varphi'$ and $(R_M =_{\mathcal{E}} \mathsf{f}(R_{N_1},\dots,R_{N_n}))\varphi'$ we conclude that

$$(C_1[R_{M_1},\dots,R_{M_k}] =_{\mathcal{E}} R_M)\varphi'.$$

Lemma 6.2 *Let $\varphi = \mathsf{new}\ \overline{n}.\sigma$ and C_1 a context such that $names(C_1) \cap \overline{n} = \emptyset$. For all $M_1,\dots,M_k \in \mathsf{sat}(\varphi)$ and T such that $C_1[M_1,\dots,M_k] \to_{\mathcal{E}}^* T$ there exist C_2 and $M_1',\dots,M_\ell' \in \mathsf{sat}(\varphi)$ such that $names(C_2) \cap \overline{n} = \emptyset$ and $T == C_2[M_1',\dots M_\ell']$. Moreover, if $\varphi' \models \mathsf{Eq}(\varphi)$ then $(C_1[R_{M_1},\dots R_{M_k}] =_{\mathcal{E}} C_2[R_{M_1'},\dots R_{M_k'}])\varphi'$.*

6.2.2.3 Decidability of $\sim_{\mathcal{E}}$

We now show the key proposition of the decidability proof.

Proposition 6.5

$$\varphi \sim_{\mathcal{E}} \varphi' \ \Leftrightarrow\ \varphi \models \mathsf{Eq}(\varphi') \text{ and } \varphi' \models \mathsf{Eq}(\varphi)$$

Proof :
$\boxed{\Rightarrow}$ It follows directly from the definition of static equivalence that $\varphi \sim_{\mathcal{E}} \varphi' \Rightarrow \varphi \models \mathsf{Eq}(\varphi')$ and $\varphi' \models \mathsf{Eq}(\varphi)$.

$\boxed{\Leftarrow}$ We have that $\varphi' \models \mathsf{Eq}(\varphi)$. Let M, N be such that $(M =_{\mathcal{E}} N)\varphi$, i.e., $\varphi =_\alpha \mathsf{new}\ \overline{n}.\sigma$ such that $names(M,N) \cap \overline{n} = \emptyset$ and $M\sigma =_{\mathcal{E}} N\sigma$. Hence $M\sigma\downarrow_{\mathcal{E}} == N\sigma\downarrow_{\mathcal{E}}$. Let $T = M\sigma\downarrow_{\mathcal{E}}$. By Lemma 6.2 we have that there exist C_M and $M_1,\dots,M_k \in \mathsf{sat}(\varphi)$ such that $names(C_M) \cap \overline{n} = \emptyset$ and

$$T == C_M[M_1,\dots M_k] \text{ and } (M =_{\mathcal{E}} C_M[R_{M_1},\dots R_{M_k}])\varphi'.$$

Similarly, as $T == N\sigma\downarrow_{\mathcal{E}}$ there exist C_N and $N_1,\dots,N_\ell \in \mathsf{sat}(\varphi)$ such that $names(C_N) \cap \overline{n} = \emptyset$ and

$$T == C_N[N_1,\dots N_\ell] \text{ and } (N =_{\mathcal{E}} C_N[R_{N_1},\dots R_{N_\ell}])\varphi'.$$

We obtain that $C_M[M_1,\dots M_k] == C_N[N_1,\dots N_\ell]$ and by Lemma 6.1 we obtain that

$$(C_M[R_{M_1},\dots R_{M_k}] =_{\mathcal{E}} C_N[R_{N_1},\dots R_{N_\ell}])\varphi'$$

and hence $(M =_{\mathcal{E}} N)\varphi'$.

Conversely, assuming $\varphi \models \mathsf{Eq}(\varphi')$ and $(M =_{\mathcal{E}} N)\varphi'$ we obtain that $(M =_{\mathcal{E}} N)\varphi$. We conclude that $\varphi \sim_{\mathcal{E}} \varphi'$.

From this and previous propositions follows a polynomial time decision procedure. To decide $\varphi \sim_{\mathcal{E}} \varphi'$ construct $\mathsf{sat}(\varphi)$ and $\mathsf{sat}(\varphi')$. These constructions can be done in polynomial time (Proposition 6.3). Moreover, each term in $\mathsf{sat}(\varphi) \cup \mathsf{sat}(\varphi')$ has polynomial DAG size (Proposition 6.4). Because of renamings, $\mathsf{Eq}(\varphi)$ and $\mathsf{Eq}(\varphi')$ may be infinite. However, as each of the equalities in $\mathsf{Eq}(\varphi)$ and $\mathsf{Eq}(\varphi')$ is of the form $C_1[R_{M_1},\dots,R_{M_k}] = C_2[R_{N_1},\dots,R_{N_\ell}]$ with $|C_i| \le c_{\mathcal{E}}$ each equality contains at most $2 \cdot c_{\mathcal{E}}$ distinct names which can be fixed. There are at most $\mathcal{O}((|\varphi|^{c_{\mathcal{E}}})^2)$ equalities in $\mathsf{Eq}(\varphi)$ each having a polynomial DAG size. Checking whether two terms in DAG representation are equal can be done in polynomial time as well. Hence, we can check in polynomial time that $\varphi \models \mathsf{Eq}(\varphi')$ and $\varphi' \models \mathsf{Eq}(\varphi)$.

6.2.3 Deciding $\sim_{\mathcal{E}}$ vs deciding $\vdash_{\mathcal{E}}$

An interesting question is what is the relation between the decidability problems for $\sim_{\mathcal{E}}$ and $\vdash_{\mathcal{E}}$. In general, these problems cannot be reduced to each other: there exist an equational theory, such that $\vdash_{\mathcal{E}}$ is decidable and $\sim_{\mathcal{E}}$ is not; there also exists (more surprisingly) an equational theory, such that $\sim_{\mathcal{E}}$ is decidable and $\vdash_{\mathcal{E}}$ is not. Indeed for many equational theories deciding $\vdash_{\mathcal{E}}$ can be reduced to $\sim_{\mathcal{E}}$. In particular, suppose that $\sim_{\mathcal{E}}$ is decidable over a signature $\mathcal{F} \uplus \{\mathsf{h}\}$ where h is a free symbol. Then we can decide $\vdash_{\mathcal{E}}$ over the signature $\mathcal{F}$ by the following encoding:

$$\mathsf{new}\ \overline{n}.\sigma \vdash_{\mathcal{E}} t \text{ iff } \mathsf{new}\ \overline{n}.\sigma \uplus \{x \mapsto \mathsf{h}(t)\} \sim_{\mathcal{E}} \mathsf{new}\ \overline{n}, a.\sigma \uplus \{x \mapsto a\}$$

where $a \notin \mathit{names}(\sigma)$. In particular this implies that $\vdash_{\mathcal{E}}$ is decidable in polynomial time for subterm convergent equational theories.

6.2.4 Further readings

While the here described procedure (from [6]) is indeed effective a direct implementation would not be efficient. Procedures for deciding static equivalence for subterm convergent equational theories have been proposed in [43, 105] and have been implemented in the tools YAPA and KISS. Both procedures can also be used to decide the theories blind and homo presented in Example 6.2. (The procedure presented in [105] and its implementation in the tool KISS is actually the outcome of an M2 internship following this course.) In [6], decidability for blind and homo (and a more general class of equational theories) was already shown, however, no generic procedure was presented.

In [119] it is shown that static equivalence is also decidable for a class of monoidal theories (which are theories over associative commutative operators including theories for exclusive or and abelian groups). Another important result [25] is that (under some mild assumptions) decision procedures for disjoint equational theories can be combined to obtain a procedure for deciding the joint equational theory.

While decidability of static equivalence has been extensively studied, the current knowledge on decidability of obervational equivalence (the generalization to the active case) is very partial. We currently only have approximate procedures for an unboundned number of sessions [79] (implemented in the tool ProVerif), approximate procedures for a bounded number of sessions [41, 42, 137] and a decision procedure for a restricted class of processes [120].

The relationship between deciding deducibility and static equivalence has been first investigated in [6]. There the above encoding of deducibility into static equivalence is proposed and its correctness proven in detail. An equational theory which is decidable for static equivalence but not for deducibility is also presented. In [88] another example of such a theory is given with detailed proofs.

6.3 Exercises

Exercice 27
Consider the equational theory enc defined by the equations

$$\mathsf{sdec}(\{x\}_y, y) = x \quad \mathsf{proj}_1(\langle x_1, x_2\rangle) = x_1 \quad \mathsf{proj}_2(\langle x_1, x_2\rangle) = x_2$$

and cipher which extends enc by the equation $\{\mathsf{sdec}(x, y)\}_y = x$. Which of the following pairs of frames are statically equivalent? Whenever applicable give the distinguishing test.

$$\begin{array}{rcl}
\{x \mapsto a\} & \overset{?}{\sim}_{\mathsf{enc}} & \{x \mapsto b\} \\
\{x \mapsto \{0\}_k\} & \overset{?}{\sim}_{\mathsf{enc}} & \{x \mapsto \{1\}_k\} \\
\mathsf{new}\ k.\{x \mapsto \{0\}_k\} & \overset{?}{\sim}_{\mathsf{enc}} & \mathsf{new}\ k.\{y \mapsto \{1\}_k\} \\
\mathsf{new}\ k.\{x \mapsto \{0\}_k\} & \overset{?}{\sim}_{\mathsf{enc}} & \mathsf{new}\ k.\{x \mapsto \{1\}_k\} \\
\mathsf{new}\ n, k.\{x \mapsto \{n\}_k, y \mapsto k\} & \overset{?}{\sim}_{\mathsf{enc}} & \mathsf{new}\ n, k, k'.\{x \mapsto \{n\}_k, y \mapsto k'\} \\
\mathsf{new}\ n, k.\{x \mapsto \{n\}_k, y \mapsto k\} & \overset{?}{\sim}_{\mathsf{cipher}} & \mathsf{new}\ n, k, k'.\{x \mapsto \{n\}_k, y \mapsto k'\}
\end{array}$$

Exercice 28
In Section 6.1.2 we considered a toy voting protocol. However, the anonymity relied on the fact that the administrator waits to have both votes before publishing the result (to model this we could enhance the language with a synchronization construct). Show that as specified above $VP_1 \not\approx_\ell VP_2$.

Exercice 29
Proposition 6.1 does not hold any more if we drop one of the condition $\overline{m} \cap (names(\sigma_1) \cup names(\sigma_2)) = \emptyset$ and $(\overline{n}_1 \cup \overline{n}_2) \cap names(\theta) = \emptyset$. Give a counter-example in each case when a condition is omitted.

Exercice 30
The aim of this exercise is to guide us through a proof of Proposition 6.2. This proof requires us to prove several properties of static equivalence. Suppose $\mathsf{new}\ \overline{n}_1.\sigma_1 \sim_{\mathcal{E}} \mathsf{new}\ \overline{n}_2.\sigma_2$. Show that

$$\mathsf{new}\ n.\mathsf{new}\ \overline{n}_1.\sigma_1 \sim_{\mathcal{E}} \mathsf{new}\ n.\mathsf{new}\ \overline{n}_2.\sigma_2 \text{ where } n \notin (\overline{n}_1 \cup \overline{n}_2), \tag{6.1}$$

$$\mathsf{new}\ \overline{n}_1.\sigma_1\{^s/_n\} \sim_{\mathcal{E}} \mathsf{new}\ \overline{n}_2.\sigma_2\{^s/_n\} \text{ where } n \notin (\overline{n}_1 \cup \overline{n}_2) \text{ and } s \text{ is a fresh name.} \tag{6.2}$$

Let $s \notin (\overline{n}_1 \cup \overline{n}_2)$. Show that

$$\mathsf{new}\ \overline{n}_1.\sigma_1 \sim_{\mathcal{E}} \mathsf{new}\ \overline{n}_2.\sigma_2 \text{ iff } \mathsf{new}\ s.\mathsf{new}\ \overline{n}_1.\sigma_1 \uplus \{x \mapsto s\} \sim_{\mathcal{E}} \mathsf{new}\ s.\mathsf{new}\ \overline{n}_2.\sigma_2 \uplus \{x \mapsto s\} \tag{6.3}$$

Use these properties (as well as Proposition 6.1) to show that the following 3 statements are equivalent.

1. $\mathsf{new}\ w.\mathsf{new}\ \overline{n}.\sigma \uplus \{x \mapsto w\} \sim_{\mathcal{E}} \mathsf{new}\ w.\mathsf{new}\ w'.\mathsf{new}\ \overline{n}.\sigma \uplus \{x \mapsto w\}$
2. $\mathsf{new}\ \overline{n}.\sigma \sim_{\mathcal{E}} \mathsf{new}\ w.\mathsf{new}\ \overline{n}.\sigma$
3. $\mathsf{new}\ \overline{n}.\sigma \sim_{\mathcal{E}} \mathsf{new}\ \overline{n}.\sigma\{^{w'}/_w\}$ where w' is a fresh name.

From these results show Proposition 6.2.

Exercice 31
Give an example illustrating that the procedure given in Section 6.2 is wrong if we would define the theory constant $c_{\mathcal{E}}$ to be $\max_{1 \leq i \leq n}(|\ell_i|)$ instead of $\max_{1 \leq i \leq n}(|\ell_i|, \mathsf{ar}(\mathcal{F}) + 1)$.

Exercice 32
We discussed the modelling of guessing attacks by static equivalence. Use the procedure of Section 6.2 to show that

$$\mathsf{new}\ w.\mathsf{new}\ n.\{z_1 \mapsto \{n\}_w; x_w \mapsto w\} \not\sim_{\mathsf{enc}} \mathsf{new}\ w.\mathsf{new}\ w'.\mathsf{new}\ n.\{z_1 \mapsto \{n\}_w; x_w \mapsto w'\}$$

and

$$\mathsf{new}\ w.\mathsf{new}\ n.\{z_1 \mapsto \{n\}_w; x_w \mapsto w\} \sim_{\mathsf{cipher}} \mathsf{new}\ w.\mathsf{new}\ w'.\mathsf{new}\ n.\{z_1 \mapsto \{n\}_w; x_w \mapsto w'\}$$

Exercice 33
The here presented procedure terminates in polynomial time given that terms are implemented by DAGs rather than trees. Give an example of two frames φ_1, φ_2 such that $\varphi_1 \not\sim_{\mathsf{enc}} \varphi_2$ but every distinguishing test is of exponential size in $|\varphi_1| + |\varphi_2|$.

Exercice 34
Give an example of two frames φ_1 and φ_2 such that $\varphi_1 \not\sim_{\mathsf{homo}} \varphi_2$ (for the theory of homomorphic encryption introduced in Example 6.2) for which the above procedure fails to distinguish the frames.