Reduction of Equational Theories for Verification of Trace Equivalence: Re-encryption, Associativity and Commutativity*

Myrto Arapinis, Sergiu Bursuc, and Mark D. Ryan

School of Computer Science, University of Birmingham**
{m.d.arapinis,s.bursuc,m.d.ryan}@cs.bham.ac.uk

Abstract. Verification of trace equivalence is difficult to automate in general because it requires relating two infinite sets of traces. The problem becomes even more complex when algebraic properties of cryptographic primitives are taken in account in the formal model. For example, no verification tool or technique can currently handle automatically a realistic model of re-encryption or associative-commutative operators.

In this setting, we propose a general technique for reducing the set of traces that have to be analyzed to a set of local traces. A local trace restricts the way in which some function symbols are used, and this allows us to perform a second reduction, by showing that some algebraic properties can be safely ignored in local traces.

In particular, local traces for re-encryption will contain only a bounded number of re-encryptions for any given ciphertext, leading to a sound elimination of equations that model re-encryption. For associativity and commutativity, local traces will determine a canonical use of the associative-commutative operator, where reasoning modulo AC is no stronger than reasoning without AC.

We illustrate these results by considering a non-disjoint combination of equational theories for the verification of vote privacy in Prêt à Voter. ProVerif can not handle the input theory as it is, but it does terminate with success on the theory obtained using our reduction result.

1 Introduction

Equivalence of formal processes, typically under the form of observational equivalence or trace equivalence, is fundamental in modeling security properties related to privacy. Some examples are strong secrecy [7], resistance against guessing attacks [14], authentication [3], unlinkability and anonymity [4], etc. Process equivalence can also be used to verify that a system implementation

* A long version of the paper and the ProVerif code are available online.

** We gratefully acknowledge financial support from EPSRC via the projects *Trust Domains* (TS/I002529/1) and *Trustworthy Voting Systems* (EP/G02684X/1).

P. Degano and J.D. Guttman (Eds.): POST 2012, LNCS 7215, pp. 169–188, 2012.

conforms to a given system specification [3]. Another example is ballot secrecy in electronic voting [20], which is of particular relevance for this paper.

In order to not miss attacks, and sometimes even to be able to execute the protocols, the formal model has to take into account relevant algebraic properties of cryptographic primitives that are used [17]. The integration of algebraic properties in models and tools for automated verification of reachability properties, like secrecy, has been quite successful. However, only few results are known for verification of process equivalence. They are in general restricted to a bounded number of sessions and a basic Dolev-Yao theory [22,29,9,10], and do not go further than subterm-convergent theories [5,15], where the right-hand side of each equation is either a constant or a subterm of the left-hand side. ProVerif can handle an unbounded number of sessions and a broad class of equational theories [8], but may not terminate and may discover false attacks. None of the above-mentioned techniques can handle associative-commutative properties, like those of XOR, abelian groups, Diffie-Hellman, etc.

The starting point of our work is a case study that can not be handled by ProVerif, namely analysis of vote privacy in Prêt à Voter (PaV) [28]. Privacy in many electronic voting systems, not only in PaV, is based on a re-encryption mixnet, whose role is to break the link between ballots that are cast and ballots that are decrypted. A realistic model for such protocols has to contain not only equations that model re-encryption, but also at least equations for the associative-commutative properties of the underlying group and for the zero-knowledge proofs output by the mixnet. However, ProVerif does not terminate for PaV even when only the single re-encryption equation $\mathsf{renc}(\mathsf{enc}(x, y, z), z') = \mathsf{enc}(x, y, f(z, z'))$ is considered along with the standard Dolev-Yao theory for public-key encryption.

Our contributions. We show how, in general, trace equivalence modulo a non-disjoint combination $\mathcal{E} \cup \mathcal{E}_{\mathsf{renc}} \cup \mathsf{AC}$, can be reduced to trace equivalence modulo $\mathcal{E}'$ - a slightly augmented version of $\mathcal{E}$. If $\mathcal{E}$ is subterm-convergent, then $\mathcal{E}'$ is subterm-convergent as well. In particular, ProVerif terminates with success for $\mathcal{E}'_{\mathsf{PaV}}$ - the result of applying our reduction to the combination of theories suggested above. The main idea in the construction of $\mathcal{E}'$ is to anticipate in advance the maximal number of re-encryptions that are necessary to apply for any given ciphertext. We only prove the soundness of the given reduction in this paper. This means that our reduction may fail to prove that some processes are equivalent, but the value of the proposed approach is shown by our ability to carry an automated proof of privacy for PaV. This is a first automated proof of trace equivalence for protocols relying on re-encryption and AC symbols.

Related work. The idea of bounding the number of application of rewrite rules is similar to the finite variant property [13] and has already been helpful to make ProVerif work modulo XOR [25]. Less like [13], and more like [25], our bound is not intrinsic to the theory, but comes from a restriction on the class of protocols. Another similarity with [25] is in our way of removing AC, but we will show that there is a fundamental difference when one considers equivalence properties. The

reduction of [25], and also the one for Diffie-Hellman in [24], have also been proven to be complete. On the other hand, these reductions are restricted to reachability properties, and do not cover equivalence properties. Furthermore, they consider a representation of protocols in terms of Horn clauses, which limits the aplicability of their results to ProVerif (or other tools based on Horn clauses) and is less general than the applied pi-calculus [2], that we use. [27,26] also consider abstractions of Diffie-Hellman and show that they are sound for reachability properties. Diffie-Hellman has some similarities with re-encryption, but the interaction of re-encryption with encryption is making it significantly different.

If results for process equivalence are limited to subterm-convergent theories [22,29,9,10,5,15], results for static equivalence go further than that. [1] shows decidability in presence of blind signatures and homomorphic encryption (without AC), and there is also a tool available [6]. Furthermore, the theory and implementation of [11] cover also trapdoor bit-commitment and malleable encryption. Malleable encryption is similar to re-encryption, but it is not associative-commutative: the value of the random can be changed, but it does not depend on its previous value. There are also results showing that algorithms for static equivalence can be combined for disjoint theories [16], and that a function symbol can be eliminated from the theory if it respects a hierarchy of sorts [23]. Our theory is a non-disjoint combination of encryption, re-encryption and other cryptographic primitives. Furthermore, [23] can not be applied to separate re-encryption from encryption, because a strict hierarchy of sorts can not be established due to the presence of the equation $\mathsf{renc}(\mathsf{enc}(x,y,z),z') = \mathsf{enc}(x,y,f(z,z'))$.

2 Preliminaries

2.1 Terms and Equational Theories

We start with an infinite set of constants $\mathcal{N}$, called names, and an infinite set of variables $\mathcal{X}$. Given a finite signature $\mathcal{F}$, $\mathcal{N}' \subseteq \mathcal{N}$ and $\mathcal{X}' \subseteq \mathcal{X}$, we denote by $\mathcal{T}(\mathcal{F},\mathcal{N}',\mathcal{X}')$ the set of terms obtained by recursively applying symbols from $\mathcal{F}$ to elements from $\mathcal{N}' \cup \mathcal{X}'$. Terms will be denoted by $u, v, s, t, \ldots$, variables by $x, y, z, \ldots$, and names by $n, m, r, \ldots$ We let $\mathcal{F}_0$ be the set of constants in $\mathcal{F}$.

For a term t, we will denote by $\mathsf{var}(t)$ the set of its variables, by $\mathsf{st}(t)$ the set of its subterms and by $\mathsf{sig}(t)$ the set of function symbols that occur in t. We say that a term is ground if $\mathsf{var}(t) = \emptyset$. A *term context* $\mathsf{C}[\_]$ is a term that has a special symbol $\_$, called hole, in place of a subterm. The application of $\mathsf{C}[\_]$ to a term t is the term $\mathsf{C}[t]$, i.e. the result of replacing the hole with t.

Given three terms t, u, v, we denote by $t\{u \mapsto v\}$ the term obtained from t by replacing every occurence of u with v. A *replacement* ρ is a partial function from terms to terms: if $\rho = \{u_1 \mapsto v_1, \ldots, u_n \mapsto v_n\}$, we have $\mathsf{dom}(\rho) = \{u_1, \ldots, u_n\}$ and $\mathsf{ran}(\rho) = \{v_1, \ldots, v_n\}$. We assume that $u_1, \ldots, u_n$ are ordered such that $u_i \in \mathsf{st}(u_j) \implies j < i$. Then, for any term t, the application of ρ to t is $t\rho = t\{u_1 \mapsto v_1\} \ldots \{u_n \mapsto v_n\}$.

A *substitution* σ is a replacement with $\mathsf{dom}(\sigma) \subseteq \mathcal{X}$. Substitutions will be denoted by $\sigma, \theta, \tau \ldots$, whereas replacements will be denoted by (annotations of) ρ.

The composition of two substitutions σ and θ is a substitution $\sigma \circ \theta$ defined by the set $\{x \mapsto (x\sigma)\theta \mid x \in \mathsf{dom}(\sigma)\} \cup \{x \mapsto x\theta \mid x \in \mathsf{dom}(\theta) \setminus \mathsf{dom}(\sigma)\}$. Given a substitution σ, if the composition $\sigma \circ \ldots \circ \sigma$ has a finite least fix point we denote it by σ^*, i.e. we have $\sigma^* = \sigma \circ \ldots \circ \sigma$ and $\sigma^* \circ \sigma = \sigma^*$. Note that, if the variables in $\mathsf{dom}(\sigma)$ can be ordered as $x_1, \ldots, x_n$ such that $i < j \implies x_j \notin \mathsf{var}(x_i\sigma)$, then σ^* exists. The restriction of a substitution σ to a set $\mathcal{V} \subseteq \mathsf{dom}(\sigma)$ is denoted by $\sigma|_{\mathcal{V}}$.

An equational theory is given by a pair $\mathcal{E} = (\mathcal{R}, \mathsf{AC}_{\mathcal{S}})$, where $\mathsf{AC}_{\mathcal{S}}$ is a set of equations modeling the associativity and commutativity of symbols in $\mathcal{S} \subseteq \mathcal{F}$ and $\mathcal{R}$ is a rewrite system convergent modulo $\mathsf{AC}_{\mathcal{S}}$. The rules of $\mathcal{R}$ are written as $l \to r$, with $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $\mathsf{var}(r) \subseteq \mathsf{var}(l)$. Given a rewrite system $\mathcal{R}$, there is a rewriting step (resp. rewriting step modulo $\mathsf{AC}_{\mathcal{S}}$) from u to v if $u = \mathsf{C}[w]$, $w = l\sigma$ (resp. $w =_{\mathsf{AC}_{\mathcal{S}}} l\sigma$) and $v = \mathsf{C}[r\sigma]$, for some context C, rewrite rule $l \to r \in \mathcal{R}$ and substitution σ. The term w is called a redex. The normal form of a term t with respect to $\mathcal{R}$ (resp. modulo $\mathsf{AC}_{\mathcal{S}}$) will be denoted by $t{\downarrow}$ (resp. $t{\downarrow}_{\mathsf{AC}}$), or by $t{\downarrow}_{\mathcal{R}}$ (resp. $t{\downarrow}_{\mathcal{R},\mathsf{AC}}$) when $\mathcal{R}$ is not clear from the context. Then, we have by definition $u =_{\mathcal{E}} v$ if and only if $u{\downarrow}_{\mathcal{R},\mathsf{AC}} =_{\mathsf{AC}} v{\downarrow}_{\mathcal{R},\mathsf{AC}}$. The existence of a rewriting step (resp. modulo AC) from u to v will be denoted by $u \to v$ (resp. $u \to_{\mathsf{AC}} v$). Relying on the convergence of $\mathcal{R}$, we can restrict ourselves to bottom-up rewriting steps, i.e. all the strict subterms of a redex are in normal form. For two terms u, v, we write $u = v$ to denote their syntactic equality, $u =_{\mathcal{E}} v$ to denote their equality modulo $\mathcal{E}$, and $u =_{\mathsf{AC}} v$ to denote their equality modulo $\mathsf{AC}_{\mathcal{S}}$.

An equational theory $(\mathcal{R}, \emptyset)$ is subterm-convergent if for every rule $l \to r \in \mathcal{R}$, we have $r \in \mathsf{st}(l) \cup \mathcal{F}_0$. To avoid confusion, when the equational theory is not clear from the context, we annotate all our symbols by the theory to which they refer to.

Example 1. The classical Dolev-Yao theory for public-key encryption is modeled by the signature $\mathcal{F}_{\mathsf{DY}} = \{\mathsf{enc}, \mathsf{dec}, \mathsf{pub}, \langle,\rangle, \pi_1, \pi_2\}$ and the subterm-convergent equational theory $\mathcal{E}_{\mathsf{DY}} = (\mathcal{R}_{\mathsf{DY}}, \emptyset)$, where

$$\mathcal{R}_{\mathsf{DY}} = \left\{ \mathsf{dec}(\mathsf{enc}(x, \mathsf{pub}(y), z), y) \to x,\ \pi_1(\langle x, y\rangle) = x,\ \pi_2(\langle x, y\rangle) = y \right\}$$

The re-encryption property of public-key encryption schemes like El-Gamal can be modeled by the signature $\mathcal{F}_{\mathsf{renc}} = \{\mathsf{enc}, \mathsf{renc}, f\}$ and the equational theory $\mathcal{E}_{\mathsf{renc}} = (\mathcal{R}_{\mathsf{renc}}, \mathsf{AC}_f)$, where

$$\mathcal{R}_{\mathsf{renc}} = \begin{cases} \mathsf{renc}(\mathsf{enc}(x, y, z), z') \to \mathsf{enc}(x, y, f(z, z')) \\ \mathsf{renc}(\mathsf{renc}(x, z), z') \to \mathsf{renc}(x, f(z, z')) \end{cases}$$

2.2 Processes and Operational Semantics

To model communication channels we consider a distinct set of constants Ch such that $\mathsf{Ch} \cap (\mathcal{N} \cup \mathcal{F}) = \emptyset$. Elements of Ch are called channel names and will be typically denoted by $a, b, c, \ldots$ Processes of our calculus are defined by the following grammar [2]:

$A, B :=$	0	plain processes			$P, Q :=$	processes	
	$A \mid B$	$\overline{\mathsf{c}}\langle u\rangle.A$	$\mathsf{c}(x).A$	$\nu n.A$		A	$\nu x.P$	$\nu n.P$
	$!A$	if $u = v$ then A else B				$P \mid Q$	$\{x \mapsto u\}$	

A name n that occurs in a process P is bound if it occurs under a νn, otherwise it is free. A variable x that occurs in P is bound if it occurs under a νx or under a $\mathsf{c}(x)$, otherwise it is free. We will denote by $\mathsf{bn}(P), \mathsf{bv}(P), \mathsf{bv}(P)$, resp. $\mathsf{fv}(P)$ the bound names (including channel names), free names, bound variables and resp. free variables of P. By α-conversion of bound names and variables we will always assume that $\mathsf{bn}(P) \cap \mathsf{fn}(P) = \emptyset$, $\mathsf{bv}(P) \cap \mathsf{fv}(P) = \emptyset$, and there are no two distinct binders for the same name or the same variable. We denote by P^α the process obtained by substituting every bound name and variable in P with a fresh one. A process P is *closed* if any variable in P is either bound or occurs in a subprocess of the form $\{x \mapsto u\}$. A *process context* $\mathsf{C}[\_]$ is a process that has a special symbol $\_$, called hole, in the place of a sub-process. The application of $\mathsf{C}[\_]$ to a process P is $\mathsf{C}[P]$, i.e. the result of replacing the hole with P. An *evaluation context* is a process context whose hole is not in the scope of a replication, a conditional, an input, or an output. We let $\mathsf{sp}(P) = \{Q \mid \exists \mathsf{C}[\_].\ P = \mathsf{C}[Q]\}$ be the set of sub-processes, $\mathsf{st}(P)$ be the set of terms (and their subterms) and $\mathsf{sig}(P) = \mathsf{sig}(\mathsf{st}(P))$ be the set of function symbols that occur in P.

Structural equivalence is the smallest equivalence relation $\equiv$ on processes that is closed under the application of evaluation contexts and the application of the following equations:

$$P \mid 0 \equiv P; \quad \nu u.0 \equiv 0; \quad !P \equiv P^\alpha \mid !P; \quad P \mid Q \equiv Q \mid P; \quad \nu u.\nu w.P \equiv \nu w.\nu u.P$$
$$P \mid (Q \mid R) \equiv (P \mid Q) \mid R; \quad P \mid \nu u.Q \equiv \nu u.(P \mid Q), \text{ if } u \notin \mathsf{fn}(P) \cup \mathsf{fv}(P)$$

A *frame* ϕ is a (static) process of the form $\nu\tilde{n}.\nu\tilde{x}.\sigma$, where $\tilde{n}$ is a sequence of names in $\mathcal{N}$, $\tilde{x}$ is a sequence of variables and σ is a substitution such that σ^* exists. We have $\mathsf{bn}(\phi) = \tilde{n}$ and $\mathsf{dom}(\phi) = \mathsf{dom}(\sigma) \setminus \tilde{x}$. The set of *recipes* for ϕ is defined as $\Re(\phi) = \mathcal{T}(\mathcal{F}, \mathcal{N} \setminus \mathsf{bn}(\phi), \mathsf{dom}(\phi))$. Recipes will sometimes be denoted by $\zeta, \chi, \ldots$ For a term u (in particular, u can be a recipe), we define the application of the frame $\phi = \nu\tilde{n}.\nu\tilde{x}.\sigma$ to u as the application of σ^* to u, i.e. we let $u[\phi] = u\sigma^*$. For a frame $\phi = \nu\tilde{n}.\nu\tilde{x}.\sigma$ and a substitution θ, we let $\phi \cup \theta = \nu\tilde{n}.\nu\tilde{x}.(\sigma \cup \theta)$.

For a process P, we let $\mathsf{fr}(P)$ be the frame associated to P, defined as $\mathsf{fr}(P) = \nu\tilde{n}.\nu\tilde{x}.\sigma$, where $\tilde{n} = \mathsf{bn}(P) \cap \mathcal{N}$, $\tilde{x} = \mathsf{bv}(P)$ and σ is the substitution obtained by the union of all the sub-processes of P of the form $\{x \mapsto u\}$. Our transition relation will ensure that all variable x has a single occurence as $\{x \mapsto u\}$ in P, thus σ is well-defined. Furthermore, for all processes P, it will be the case that the variables in $\mathsf{dom}(\sigma)$ can be ordered as $x_1, \ldots, x_n$ such that $i < j \implies x_j \notin \mathsf{var}(x_i\sigma)$. Therefore, σ^* always exists and $\mathsf{fr}(P)$ is indeed a frame.

Labeled reduction is a relation between closed processes defined by the following rules, modulo structural equivalence: for any evaluation context $\mathsf{C}[\_]$ with $\phi_{\mathsf{C}} = \mathsf{fr}(\mathsf{C}[\_])$,

COMM	$\mathsf{C}[\overline{\mathsf{c}}\langle u\rangle.A \mid \mathsf{c}(x).B] \xrightarrow{\tau} \mathsf{C}[\nu x.(A \mid B \mid \{x \mapsto u\})]$	
THEN	$\mathsf{C}[\text{if } u = v \text{ then } A \text{ else } B] \xrightarrow{\tau} \mathsf{C}[A]$	if $u[\phi_{\mathsf{C}}] =_{\mathcal{E}} v[\phi_{\mathsf{C}}]$
ELSE	$\mathsf{C}[\text{if } u = v \text{ then } A \text{ else } B] \xrightarrow{\tau} \mathsf{C}[B]$	if $u[\phi_{\mathsf{C}}] \neq_{\mathcal{E}} v[\phi_{\mathsf{C}}]$
IN	$\mathsf{C}[\mathsf{c}(x).A] \xrightarrow{\mathsf{c}(\zeta)} \mathsf{C}[\nu x.(A \mid \{x \mapsto \zeta\})]$	if $c \notin \mathsf{bn}(\mathsf{C}[\_])$ & $\zeta \in \Re(\phi_{\mathsf{C}})$
OUT	$\mathsf{C}[\overline{\mathsf{c}}\langle u\rangle.A] \xrightarrow{\nu x.\overline{\mathsf{c}}\langle x\rangle} \mathsf{C}[A \mid \{x \mapsto u\}]$	if $c \notin \mathsf{bn}(\mathsf{C}[\_])$ & $x \notin \mathsf{var}(\mathsf{C}[\overline{\mathsf{c}}\langle u\rangle.A])$

The semantics is very similar to the one in [2], with superficial differences that help in our proofs. The most notable difference is that we never apply the frame to the process, but only use the frame where it makes a difference, i.e. in tests. Similarly, equational reasoning is not part of structural equivalence, but is only used in tests. A *trace* is a sequence of labeled reductions $P_1 \xrightarrow{\alpha_1} P_2 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} P_n$. We will denote such a trace by $P_1 \xrightarrow{w} P_n$, where $w = \alpha_1 \ldots \alpha_n$. We let $\mathsf{obs}(w)$ be the sequence of labels obtained by erasing all occurence of τ in w.

Example 2. Consider the theory $\mathcal{E} = \mathcal{E}_{\mathsf{DY}} \cup \mathcal{E}_{\mathsf{renc}}$ from Example 1, and
$P = \nu k.\overline{\mathsf{c}}\langle \mathsf{enc}(a, \mathsf{pub}(k), r)\rangle.\mathsf{c}(x).\overline{\mathsf{c}}\langle\langle x, x\rangle\rangle.\mathsf{c}(y).\mathsf{if}\ x \neq \mathsf{enc}(a, \mathsf{pub}(k), r)\ \&$
$y \neq \mathsf{enc}(a, \mathsf{pub}(k), r)\ \&\ \mathsf{dec}(x, k) = \mathsf{dec}(y, k)\ \ \mathsf{then}\ \overline{\mathsf{c}}\langle \mathsf{dec}(y, k)\rangle$
We have $P \xrightarrow{\nu z_1.\overline{\mathsf{c}}\langle z_1\rangle} \xrightarrow{\mathsf{c}(\mathsf{renc}(z_1, n_1))} \xrightarrow{\nu z_2.\overline{\mathsf{c}}\langle z_2\rangle} \xrightarrow{\mathsf{c}(\mathsf{renc}(\pi_1(z_2), n_2))} \xrightarrow{\tau} P_0$, where

$$P_0 \equiv \nu k.\nu x.\nu y.(\overline{\mathsf{c}}\langle \mathsf{dec}(y, k)\rangle \mid \{z_1 \mapsto \mathsf{enc}(a, \mathsf{pub}(k), r)\} \mid \{x \mapsto \mathsf{renc}(z_1, n_1)\} \mid \{z_2 \mapsto \langle x, x\rangle\} \mid \{y \mapsto \mathsf{renc}(\pi_1(z_2), n_2)\})$$

and $y[\mathsf{fr}(P_0)] = \mathsf{renc}(\pi_1(\langle \mathsf{renc}(z_1[\phi], n_1), x[\phi]\rangle), n_2) =_{\mathcal{E}} \mathsf{enc}(a, \mathsf{pub}(k), f(f(r, n_1), n_2))$. Furthermore, $P_0 \xrightarrow{\nu z_3.\overline{\mathsf{c}}\langle z_3\rangle} P_1$ for some process $P_1 = \nu k.\nu x.\nu y.\mathsf{fr}(P_1)$ such that $\mathsf{fr}(P_1) = \mathsf{fr}(P_0) \cup \{z_3 \mapsto \mathsf{dec}(y, k)\}$. Then, we have $z_3[\mathsf{fr}(P_1)] =_{\mathcal{E}} a$.

2.3 Trace Equivalence and Secrecy

Definition 1 (static equivalence). *We say that two frames ϕ, ψ are in static equivalence modulo an equational theory $\mathcal{E}$, denoted by $\phi \overset{s}{\sim} \psi$, if $\mathsf{dom}(\phi) = \mathsf{dom}(\psi)$ and $\forall \zeta_1, \zeta_2 \in \Re(\phi) \cap \Re(\psi).\ \zeta_1[\phi] =_{\mathcal{E}} \zeta_2[\phi] \Leftrightarrow \zeta_1[\psi] =_{\mathcal{E}} \zeta_2[\psi]$*

When $\mathcal{E}$ is not clear from the context, we use the notation $\phi \overset{s}{\sim}_{\mathcal{E}} \psi$. We say that two traces $P \xrightarrow{w_1} P'$ and $Q \xrightarrow{w_2} Q'$ are in static equivalence if $\mathsf{obs}(w_1) = \mathsf{obs}(w_2)$ and $\mathsf{fr}(P') \overset{s}{\sim} \mathsf{fr}(Q')$.

Definition 2 (trace equivalence). *We say that two plain processes P, Q are in trace equivalence, denoted by $P \sim Q$, if for every trace $P \xrightarrow{w_1} P'$, there exists a trace $Q \xrightarrow{w_2} Q'$ such that $\mathsf{obs}(w_1) = \mathsf{obs}(w_2)$ and $\mathsf{fr}(P') \overset{s}{\sim} \mathsf{fr}(Q')$. Moreover, each trace of Q must have a corresponding statically equivalent trace of P.*

Example 3. Continuing example 2, let us consider the process $Q = P\{a \mapsto b\}$ and let $P \xrightarrow{w_1} P_1$ be the exhibited trace. Then, we have $P \not\sim Q$, because

- for $z_3, a \in \Re(\mathsf{fr}(P_1))$, we have $z_3[\mathsf{fr}(P_1)] =_{\mathcal{E}} a[\mathsf{fr}(P_1)]$
- for every corresponding trace $Q \xrightarrow{w_2} Q_1$ with $\mathsf{obs}(w_1) = \mathsf{obs}(w_2)$, we have $z_3[\mathsf{fr}(Q_1)] \neq_{\mathcal{E}} a[\mathsf{fr}(Q_1)]$, and thus $\mathsf{fr}(P_1) \not\overset{s}{\sim} \mathsf{fr}(Q_1)$. In fact, we have $z_3[\mathsf{fr}(Q_1)] =_{\mathcal{E}} b[\mathsf{fr}(Q_1)]$.

Definition 3 (intruder knowledge and secrecy). *Let $\mathcal{E}$ be an equational theory. For a frame ϕ, we let $\mathcal{I}(\phi, \mathcal{E}) = \{u \mid \exists \zeta \in \Re(\phi).\ \zeta[\phi] =_{\mathcal{E}} u\}$. For all processes P, we let $\mathcal{I}(P, \mathcal{E}) = \{u \mid \exists Q, w.\ P \xrightarrow{w} Q\ \&\ u \in \mathcal{I}(\mathsf{fr}(Q), \mathcal{E})\}$ and $\mathcal{S}(P, \mathcal{E}) = \mathsf{bn}(P) \setminus \mathcal{I}(P, \mathcal{E})$.*

3 Motivation and Statement of the Reduction Result

3.1 Starting Point: The Case Study

In this paper and in the corresponding ProVerif code, we only perform the analysis of PaV for the case of two eligible voters $\mathsf{idA}, \mathsf{idB}$ and two candidates a, b. This is mainly for simplicity of presentation, but also because we believe that a result like the one in [12] could be translated to privacy properties.

In addition to El-Gamal encryption and re-encryption, modeled by $\mathcal{E}_{\mathsf{DY}}$ and $\mathcal{E}_{\mathsf{renc}}$ from example 1, PaV relies on zero-knowledge proofs to provide universal verifiability of the election result. We have to model these proofs in our analysis, to ensure that we do not miss any attacks on privacy that may be made possible by the additional information that is published. In particular, PaV relies on mixnet proofs and on proofs of correct decryption, that we model by the signature $\mathcal{F}_{\mathsf{ver}} = \{\mathsf{mixPf}/6, \mathsf{checkMix}/5, \mathsf{decPf}/3, \mathsf{checkDec}/4, \mathsf{ok}/0\}$ and the equational theories $\mathcal{E}_{\mathsf{decP}} = (\mathcal{R}_{\mathsf{decP}}, \emptyset), \mathcal{E}_{\mathsf{mixP}} = (\mathcal{R}_{\mathsf{mixP}}, \emptyset)$, where:

$$\mathcal{R}_{\mathsf{decP}} = \left\{ \begin{array}{l} \mathsf{checkDec}(\mathsf{decPf}(\mathsf{enc}(x, \mathsf{pub}(y), z), x, y), \\ \qquad\qquad \mathsf{enc}(x, \mathsf{pub}(y), z), x, \mathsf{pub}(y)) \to \mathsf{ok} \end{array} \right.$$

$$\mathcal{R}_{\mathsf{mixP}} = \left\{ \begin{array}{l} \mathsf{checkMix}(\mathsf{mixPf}(x, y, \mathsf{renc}(x, z_x), \mathsf{renc}(y, z_y), z_x, z_y), \\ \qquad\qquad x, y, \mathsf{renc}(x, z_x), \mathsf{renc}(y, z_y)) \to \mathsf{ok} \\ \mathsf{checkMix}(\mathsf{mixPf}(x, y, \mathsf{renc}(y, z_y), \mathsf{renc}(x, z_x), z_y, z_x), \\ \qquad\qquad x, y, \mathsf{renc}(y, z_y), \mathsf{renc}(x, z_x)) \to \mathsf{ok} \end{array} \right.$$

The main idea of PaV is that an election authority A creates ballots that contain the names of the candidates in a random order on the left-hand side and their corresponding encryption in the same order on the right-hand side. This allows the voter to mark a vote for the desired candidate and scan only the encrypted part of the ballot, the right-hand side, to be posted on the bulletin board. Because the random order of candidates in the ballot and the decryption key are assumed to be secret, this ensures vote privacy, and even coercion-resistance, if care is taken to destroy the left-hand side. We use the following equational theory to model the actions of the voter during voting: $\mathcal{F}_{\mathsf{vote}} = \{\mathsf{vote}, \langle, \rangle\}$ and $\mathcal{E}_{\mathsf{vote}} = \{\mathcal{R}_{\mathsf{vote}}, \emptyset\}$, where $\mathcal{R}_{\mathsf{vote}} = \{\mathsf{vote}(\langle x, y\rangle, \langle x_e, y_e\rangle), x) \to x_e,\ \mathsf{vote}(\langle x, y\rangle, \langle x_e, y_e\rangle), y) \to y_e\}$.

After the encrypted votes get to the bulletin board, the design of PaV is similar to other voting systems like JCJ/Civitas or Helios: ballots are anonymized by a re-encryption mixnet and decrypted by the holders of the secret key. Putting it all together, the equational theory and the process that model PaV are given by $\mathcal{E}_{\mathsf{PaV}}$ and P_{PaV} in figure 1. V is the process for a voter, A is the process for the election authority that constructs the ballots, B is the process for the public bulletin board, M is the process for a mix server and T is the process for a trustee holding the decryption key.

In process M, we have $\{i, j\} = \{1, 2\}$ and $\mathsf{mixProof} = \mathsf{mixPf}(x_1, x_2, \mathsf{renc}(\pi_i(x_{\mathsf{ballots}}), n_1), \mathsf{renc}(\pi_j(x_{\mathsf{ballots}}), n_2), n_1, n_2)$. In process T, we have $\mathsf{decP}_i = \mathsf{decPf}(\pi_i(x_{\mathsf{ballots}}), \mathsf{dec}(\pi_i(x_{\mathsf{ballots}}), \mathsf{sk}), \mathsf{sk})$, for all $i \in \{1, 2\}$. The channel c_{printer} is

$$\mathcal{E}_{\mathsf{PaV}} = \mathcal{E}_{\mathsf{DY}} \cup \mathcal{E}_{\mathsf{renc}} \cup \mathcal{E}_{\mathsf{decP}} \cup \mathcal{E}_{\mathsf{mixP}} \cup \mathcal{E}_{\mathsf{vote}}$$
$$P_{\mathsf{PaV}} = \nu \mathsf{sk}.\nu c_{\mathsf{printer}}.\nu c_{\mathsf{ver}}.\nu c_{\mathsf{trustee}}.\ (\mathsf{V}(\mathsf{idA}, v_1) \mid \mathsf{V}(\mathsf{idB}, v_2) \mid \mathsf{A} \mid \mathsf{B} \mid \mathsf{M} \mid \mathsf{T})$$

$$\begin{aligned}
\mathsf{A} &= \nu c_{\mathsf{auth}}.(\overline{\mathsf{c_{auth}}}\langle a, b\rangle \mid \overline{\mathsf{c_{auth}}}\langle b, a\rangle \mid \mathsf{c_{auth}}(x_{\mathsf{can}}).\nu r_1.\nu r_2.\\
&\quad \overline{c_{\mathsf{printer}}}\langle x_{\mathsf{can}}, \langle \mathsf{enc}(\pi_1(x_{\mathsf{can}}), \mathsf{pub}(\mathsf{sk}), r_1), \mathsf{enc}(\pi_2(x_{\mathsf{can}}), \mathsf{pub}(\mathsf{sk}), r_2)\rangle\rangle\\
\mathsf{V}(\mathsf{id}, v) &= \mathsf{c_{printer}}(x_{\mathsf{ballot}}).\overline{\mathsf{c_{scanner}}}\langle \mathsf{id}, \mathsf{vote}(x_{\mathsf{ballot}}, v)\rangle.\overline{\mathsf{c_{ver}}}\langle \mathsf{id}, \mathsf{vote}(x_{\mathsf{ballot}}, v)\rangle\\
\mathsf{B} &= \mathsf{c_{scanner}}(\mathsf{ballot}_1).\mathsf{c_{scanner}}(\mathsf{ballot}_2).\overline{\mathsf{c_{mix}}}\langle \mathsf{ballot}_1, \mathsf{ballot}_2\rangle\\
\mathsf{M} &= \mathsf{c_{ver}}(y).\mathsf{c_{ver}}(z).\mathsf{c_{mix}}(x_{\mathsf{ballots}}).\mathsf{if}\ \langle \pi_1(y), \pi_1(z), \langle \pi_2(y), \pi_2(z)\rangle\rangle = \langle \mathsf{idA}, \mathsf{idB}, x_{\mathsf{ballots}}\rangle\\
&\quad \mathsf{then}\ \nu n_1.\nu n_2.\overline{\mathsf{c_{trustee}}}\langle \mathsf{renc}(\pi_i(x_{\mathsf{ballots}}), n_1), \mathsf{renc}(\pi_j(x_{\mathsf{ballots}}), n_2)\rangle.\\
&\qquad \overline{\mathsf{c_{board}}}\langle \mathsf{renc}(\pi_i(x_{\mathsf{ballots}}), n_1), \mathsf{renc}(\pi_j(x_{\mathsf{ballots}}), n_2), \mathsf{mixProof}\rangle\\
\mathsf{T} &= \overline{\mathsf{c_{board}}}\langle \mathsf{pub}(\mathsf{sk})\rangle.\mathsf{c_{trustee}}(x_{\mathsf{ballots}}).\overline{\mathsf{c_{board}}}\langle \pi_1(x_{\mathsf{ballots}}), \mathsf{dec}(\pi_1(x_{\mathsf{ballots}}), \mathsf{sk}), \mathsf{decP}_1\rangle.\\
&\qquad \overline{\mathsf{c_{board}}}\langle \pi_2(x_{\mathsf{ballots}}), \mathsf{dec}(\pi_2(x_{\mathsf{ballots}}), \mathsf{sk}), \mathsf{decP}_2\rangle
\end{aligned}$$

Fig. 1. Formal model of Prêt à Voter

where V gets the ballots - it is private because the order of candidates should be kept secret. c_{ver} is a private channel whose role is to enforce eligibility: exactly the ballots of idA and idB go into the mix. Without this, the intruder could mount an attack against the privacy of idA by replacing the ballot of idB with a copy of the ballot of idA in one of the public channels c_{scanner} or c_{mix}, as in e.g. [19]. The channel c_{trustee} has to be private to ensure that the ballots that are decrypted are indeed the ones that are mixed, and are not supplied by the intruder. Note that the channels c_{scanner}, c_{board} and c_{mix} are public, and all information that goes on private channels is also published on c_{board}.

To verify that P_{PaV} satisfies vote-privacy we check that $P_{\mathsf{PaV}}\{v_1 \mapsto a\}\{v_2 \mapsto b\} \sim_{\mathcal{E}_{\mathsf{PaV}}} P_{\mathsf{PaV}}\{v_1 \mapsto b\}\{v_2 \mapsto a\}$ [20]. The motivation of our work is that, when given as input this task (even without AC_f, $\mathcal{E}_{\mathsf{mixP}}$ and $\mathcal{E}_{\mathsf{decP}}$), ProVerif does not terminate. The non-termination is certainly due to $\mathcal{E}_{\mathsf{renc}}$, because $\mathcal{E}_{\mathsf{DY}} \cup \mathcal{E}_{\mathsf{decP}} \cup \mathcal{E}_{\mathsf{mixP}} \cup \mathcal{E}_{\mathsf{vote}}$ is a subterm-convergent theory and is easily handled by ProVerif.

3.2 General Setting for the Reduction

For a term t and process P, we let:
$\mathsf{re}(t) = \{u \in \mathsf{st}(t) \mid \mathsf{top}(u) \in \{\mathsf{enc}, \mathsf{renc}\}\}$; $\mathsf{re}(P) = \{u \in \mathsf{st}(P) \mid \mathsf{top}(u) \in \{\mathsf{enc}, \mathsf{renc}\}\}$
$\mathsf{ran}(t) = v$, if $t = \mathsf{enc}(t_1, t_2, v) \vee t = \mathsf{renc}(t_1, v)$; $\mathsf{ran}(P) = \{\mathsf{ran}(u) \mid u \in \mathsf{re}(P)\}$

Assumptions about the equational theory. In general, we consider a signature $\mathcal{F}$ such that $\{\mathsf{enc}, \mathsf{renc}, f\} \subseteq \mathcal{F}$ and a class of equational theories InpTh such that for all $\mathcal{E} \in \mathsf{InpTh}$, we have $\mathcal{E} = \mathcal{E}' \cup \mathcal{E}_{\mathsf{renc}}$, for some equational theory $\mathcal{E}' = (\mathcal{R}', \emptyset)$. We assume furthermore that for each rule $l \to r \in \mathcal{R}'$:

(ae_1) $\mathsf{top}(l) \notin \{\mathsf{enc}, \mathsf{renc}\}$ and $f \notin \mathsf{sig}(l)$
(ae_2) $\mathsf{sig}(r) \cap \{\mathsf{renc}, \mathsf{enc}, f\} = \emptyset$
(ae_3) for all $t, t' \in \mathsf{re}(l)$, we have:
· $\mathsf{ran}(t) \in \mathsf{var}(l) \setminus \mathsf{var}(r)$
· $\mathsf{ran}(t) = \mathsf{ran}(t') \implies t = t'$

It is easy to see that $\mathcal{E}_{\mathsf{PaV}}$ satisfies (ae_1)-(ae_3).

Assumptions about the class of processes. We define the weak symbols of $\mathcal{E} = (\mathcal{R}, \mathsf{AC}_f)$ by $\mathsf{W}(\mathcal{E}) = \{g \in \mathcal{F} \mid l \to r \in \mathcal{R} \ \&\ g \in \mathsf{sig}(l) \implies r \in \mathcal{F}_0\}$. For example, we have $\mathsf{W}(\mathcal{E}_{\mathsf{PaV}}) = \mathcal{F}_{\mathsf{ver}} \cup \{f\}$. For all term u, we define:

$$\mathsf{st}_p(u) = \{t \in \mathsf{st}(u) \mid u = \mathsf{C}[t] \implies \exists \mathsf{C}_1, \mathsf{C}_2.\ \mathsf{C}[\_] = \mathsf{C}_1[\mathsf{C}_2[\_]] \ \& \\ (\mathsf{top}(\mathsf{C}_2) \in \mathsf{W}(\mathcal{E}) \vee \mathsf{C}_2 = \mathsf{renc}(t_1, \mathsf{C}'_2[\_]) \vee \mathsf{C}_2 = \mathsf{enc}(t_1, t_2, \mathsf{C}'_2[\_]))\}$$

Intuitively, $\mathsf{st}_p(t)$ are the subterms of t whose every occurence is "protected" by a weak symbol or by $\{\mathsf{enc}, \mathsf{renc}\}$. For example, $\mathsf{st}_p(\langle a, c, \mathsf{renc}(b, c), \mathsf{renc}(b, d)\rangle) = \{d\}$. We assume the following properties about every process P that we consider:

(ap_1) $f \notin \mathsf{sig}(P)$	(ap_3) $\cdot\ u, u' \in \mathsf{re}(P)\ \&\ \mathsf{ran}(u) = \mathsf{ran}(u') \implies u = u'$
(ap_2) $\mathsf{ran}(P) \subseteq \mathsf{bn}(P)$	$\cdot\ t \in \mathsf{ran}(P)\ \&\ u \in \mathsf{st}(P)\ \&\ t \in \mathsf{st}(u) \implies t \in \mathsf{st}_p(u)$
	(ap_4) $!Q \in \mathsf{sp}(P) \implies \{\mathsf{renc}, \mathsf{enc}\} \cap \mathsf{sig}(Q) = \emptyset$

The main goal of assumptions (ap_2),(ap_3) is to ensure that the elements of $\mathsf{ran}(P)$ are kept secret by the process. It is easy to see that P_{PaV} satisfies (ap_1)-(ap_4).

The reduced theory. Given an input theory $\mathcal{E} = (\mathcal{R}' \cup \mathcal{R}_{\mathsf{renc}}, \mathsf{AC}_f) \in \mathsf{InpTh}$, let $\mathcal{R}_{\mathsf{renc}}^{-1}$ be the inverse of $\mathcal{R}_{\mathsf{renc}}$, that is: $\mathcal{R}_{\mathsf{renc}}^{-1} = \begin{cases} \mathsf{enc}(x, y, f(z, z')) \to \mathsf{renc}(\mathsf{enc}(x, y, z), z') \\ \mathsf{renc}(x, f(z, z')) \to \mathsf{renc}(\mathsf{renc}(x, z), z') \end{cases}$

Note that $\mathcal{R}_{\mathsf{renc}}^{-1}$ is convergent modulo AC. Given a term t, we let $\mathsf{var}_f(t) = \{s \in \mathsf{var}(t) \mid \mathsf{enc}(u, v, s) \in \mathsf{st}(t) \ \vee\ \mathsf{renc}(u, s) \in \mathsf{st}(t)\}$. Then, for all $m \geq 1$, an (f, m)-substitution for t is a substitution σ such that $\mathsf{dom}(\sigma) \subseteq \mathsf{var}_f(t)$ and $\forall x \in \mathsf{dom}(\sigma).\ x\sigma = f(\ldots f(x_0, x_1) \ldots, x_k)$, where $1 \leq k \leq m\ \&\ x_0, \ldots, x_k \notin \mathsf{var}(t)\ \&\ \forall x' \neq x. x_0, \ldots, x_k \notin \mathsf{var}(x'\sigma)$. We denote by $\Theta_{f,m}(t)$ the set of (f, m)-substitutions for t. Now, we consider a particular case of narrowing [21] with respect to $\mathcal{R}_{\mathsf{renc}}^{-1}$ to define a set of variants [13] of a term: $\mathcal{V}_m^{\mathsf{renc}}(t) = \{(t\sigma)\downarrow_{\mathcal{R}_{\mathsf{renc}}^{-1}} \mid \sigma \in \Theta_{f,m}(t)\}$.

Example 4. We have $\mathcal{V}_2^{\mathsf{renc}}(\mathsf{enc}(a, b, x)) = \{\mathsf{enc}(a, b, x), \mathsf{renc}(\mathsf{enc}(a, b, x_0), x_1), \mathsf{renc}(\mathsf{renc}(\mathsf{enc}(a, b, x_0), x_1), x_2)\}$ and $\mathcal{V}_2^{\mathsf{renc}}(\mathsf{renc}(a, x)) = \{\mathsf{renc}(a, x), \mathsf{renc}(\mathsf{renc}(a, x_0), x_1), \mathsf{renc}(\mathsf{renc}(\mathsf{renc}(a, x_0), x_1), x_2)\}$

Finally, we can define the reduced theory that corresponds to $\mathcal{E}$ and m:

$$\mathcal{E}_m = (\mathcal{R}_m, \emptyset) \qquad \mathcal{R}_m = \{l' \to r \mid \exists l \to r \in \mathcal{R}'.\ l' \in \mathcal{V}_m^{\mathsf{renc}}(l)\}$$

We let OutTh be the set of all reduced theories that correspond to some $\mathcal{E} \in \mathsf{InpTh}$ and some $m \geq 1$.

Example 5. Let us consider the theory $\mathcal{E}_{\mathsf{PaV}}$ and $m = 1$. We have $\mathcal{E}_{\mathsf{PaV},m} = (\mathcal{E}_{\mathsf{PaV}} \setminus \mathcal{E}_{\mathsf{renc}}) \cup (\mathcal{S}, \emptyset)$, where

$$\mathcal{S} = \begin{cases} \mathsf{dec}(\mathsf{renc}(\mathsf{enc}(x, \mathsf{pub}(y), z_0), z_1), y) \to x \\ \mathsf{checkDec}(\mathsf{decPf}(\mathsf{renc}(\mathsf{enc}(x, \mathsf{pub}(y), z_0), z_1), x, y), \\ \qquad \mathsf{renc}(\mathsf{enc}(x, \mathsf{pub}(y), z_0), z_1), x, \mathsf{pub}(y)) \to \mathsf{ok} \\ \mathsf{checkMix}(\mathsf{mixPf}(x, x, \mathsf{renc}(\mathsf{renc}(x, z_x^0), z_x^1), \mathsf{renc}(y, z_y), f(z_x^0, z_x^1), z_y), \\ \qquad x, y, \mathsf{renc}(\mathsf{renc}(x, z_x^0), z_x^1), \mathsf{renc}(y, z_y)) \to \mathsf{ok} \\ \text{plus rules for } \mathsf{checkMix} \text{ corresponding to other permutations and} \\ \text{to substitutions } \{z_y \mapsto f(z_y^0, z_y^1)\} \text{ and } \{z_x \mapsto f(z_x^0, z_x^1), z_y \mapsto f(z_y^0, z_y^1)\} \end{cases}$$

Lemma 1. *For all $\mathcal{E} = \mathcal{E}' \cup \mathcal{E}_{\mathsf{renc}} \in \mathsf{InpTh}$ such that $\mathcal{E}'$ is subterm-convergent, $\mathcal{E}_m$ is subterm-convergent.*

We let $\mathsf{ran}_r(P) = \{t \mid \mathsf{renc}(u,t) \in \mathsf{st}(P)\}$. Our main result is a reduction of trace equivalence modulo any $\mathcal{E} \in \mathsf{InpTh}$ to trace equivalence modulo $\mathcal{E}_m \in \mathsf{OutTh}$, for a well-chosen m:

Theorem 1 (Main theorem). *For all processes P, Q that satisfy (ap_1)-(ap_4) let $m = 2 * \mathsf{max}(|\mathsf{ran}_r(P)|, |\mathsf{ran}_r(Q)|) + 1$. Then, for all equational theory $\mathcal{E} \in \mathsf{InpTh}$, we have*

$$P \sim_{\mathcal{E}_m} Q \implies P \sim_{\mathcal{E}} Q$$

where $\mathcal{E}_m \in \mathsf{OutTh}$ is the reduced theory that corresponds to $\mathcal{E}$ and m.

4 Reduction of the Set of Traces

As a first step in the proof of theorem 1, we introduce a restricted notion of trace equivalence, $P \simeq Q$, that depends only on so-called local traces of P and Q. We show that $P \sim_{\mathcal{E}_m} Q \implies P \simeq_{\mathcal{E}_m} Q$ and $P \simeq_{\mathcal{E}} Q \implies P \sim_{\mathcal{E}} Q$.

4.1 Occurence Order for a Frame

In the following, we assume that for all frame ϕ we are given a partial order $\prec$ on variables in $\mathsf{var}(\phi)$. For all P, Q such that $P \xrightarrow{w} Q$, we associate such an order $\prec$ to $\mathsf{fr}(Q)$: for all $x, y \in \mathsf{var}(\mathsf{fr}(Q))$ we have $x \prec y$ iff $w = w_1 w_2$ and there is a P_1 such that $P \xrightarrow{w_1} P_1 \xrightarrow{w_2} Q$, $x \in \mathsf{var}(\phi(P_1))$ and $y \notin \mathsf{var}(\mathsf{fr}(P_1))$. We let $x \preceq y$ if $x \prec y \vee x = y$. The orders $\prec$ and $\preceq$ are extended to sets of variables from $\mathsf{var}(\phi)$ by: $S_1 \preceq S_2 \Leftrightarrow \forall x \in S_1 \exists y \in S_2 . x \preceq y$ and $S_1 \prec S_2 \Leftrightarrow S_1 \preceq S_2 \;\&\; \exists y \in S_2 \forall x \in S_1 . x \prec y$. Finally, we extend $\prec$ (and $\preceq$) to a pre-order on terms in $\mathcal{T}(\mathcal{F}, \mathcal{N}, \mathsf{var}(\phi))$ by letting $u \prec v \Leftrightarrow \mathsf{var}(u) \prec \mathsf{var}(v)$ (and $u \preceq v \Leftrightarrow \mathsf{var}(u) \preceq \mathsf{var}(v)$).

Example 6. Let $P = \overline{\mathsf{c}}\langle a \rangle . \overline{\mathsf{c}}\langle b \rangle . \overline{\mathsf{c}}\langle \mathsf{renc}(a,b) \rangle$. Then, we have

$$P \xrightarrow{\nu x_1 . \overline{\mathsf{c}}\langle x_1 \rangle} \xrightarrow{\nu x_2 . \overline{\mathsf{c}}\langle x_2 \rangle} \xrightarrow{\nu x_3 . \overline{\mathsf{c}}\langle x_3 \rangle} \{x_1 \mapsto a\} \mid \{x_2 \mapsto b\} \mid \{x_3 \mapsto \mathsf{renc}(a,b)\} = Q$$

and $x_1 \prec x_2 \preceq \mathsf{renc}(x_1, x_2) \prec x_3$ and $\mathsf{renc}(x_1, x_2)[\mathsf{fr}(Q)] = x_3[\mathsf{fr}(Q)]$.

For the trace $P \xrightarrow{w} P_1$ of example 2, we have $z_1 \preceq \mathsf{renc}(z_1, n_1) \prec x \prec z_2 \preceq \mathsf{renc}(\pi_2(z_2), n_2) \prec y$.

4.2 Local Traces in General

The definitions and results from this section stand for any equational theory $\mathcal{E}$, with no restriction on $\mathcal{E}$ and no restriction on the class of processes.

In the following, we consider given a *locality function* $\mathcal{L}$, that associates to each frame ϕ a subset of its recipes, i.e. $\mathcal{L}(\phi) \subseteq \Re(\phi)$.

Definition 4 (local static equivalence). *Let $\mathcal{L}$ be a locality function. We say that two frames ϕ, ψ are in $\mathcal{L}$-local static equivalence, denoted by $\phi \overset{s}{\simeq} \psi$, if* $\mathsf{dom}(\phi) = \mathsf{dom}(\psi)$ *and* $\forall \zeta_1, \zeta_2 \in \mathcal{L}(\phi) \cap \mathcal{L}(\psi).\ \zeta_1[\phi] =_{\mathcal{E}} \zeta_2[\phi] \Leftrightarrow \zeta_1[\psi] =_{\mathcal{E}} \zeta_2[\psi]$.

When $\mathcal{L}$ or $\mathcal{E}$ is not clear from the context, we use the notation $\phi \overset{s}{\simeq}_{\mathcal{L},\mathcal{E}} \psi$. We say that two traces $P \xrightarrow{w_1} P'$ and $Q \xrightarrow{w_2} Q'$ are in $\mathcal{L}$-local static equivalence if $\mathsf{obs}(w_1) = \mathsf{obs}(w_2)$ and $\mathsf{fr}(P') \overset{s}{\simeq}_{\mathcal{L}} \mathsf{fr}(Q')$. For a sequence of labels w, we let $\mathsf{inp}(w)$ be the set of recipes that occur as inputs in w, i.e. $\mathsf{inp}(w) = \{\zeta \mid \exists w_1, w_2, c.\ w = w_1 \mathsf{c}(\zeta) w_2\}$. A trace $P \xrightarrow{w} Q$ is $\mathcal{L}$-local if $\mathsf{inp}(w) \subseteq \mathcal{L}(\mathsf{fr}(Q))$.

Definition 5 (local trace equivalence). *Let $\mathcal{L}$ be a locality function. We say that two processes P, Q are in $\mathcal{L}$-local trace equivalence, denoted by $P \simeq Q$, if for all $\mathcal{L}$-local trace $P \xrightarrow{w_1} P'$ there exists a $\mathcal{L}$-local trace $Q \xrightarrow{w_2} Q'$ such that* $\mathsf{obs}(w_1) = \mathsf{obs}(w_2)$ *and* $\mathsf{fr}(P') \overset{s}{\simeq} \mathsf{fr}(Q')$. *Moreover, for all $\mathcal{L}$-local trace of Q there must exist a corresponding $\mathcal{L}$-local statically equivalent $\mathcal{L}$-local trace of P.*

When $\mathcal{L}$ is not clear from the context, we use the notation $P \simeq_{\mathcal{L},\mathcal{E}} Q$.

The idea of $\mathcal{L}$-locality is to restrict the set of traces that have to be considered. A locality function is especially useful if it admits a *normalization function* that assigns to each recipe an equivalent local recipe:

Definition 6 (Normalization function). *Given two locality functions $\mathcal{L}_1, \mathcal{L}_2$ and a frame ϕ, a* normalization function from $\mathcal{L}_1$ to $\mathcal{L}_2$ *associates to each recipe $\zeta \in \mathcal{L}_1(\phi)$ an equivalent $\mathcal{L}_2$-*local recipe $\mathsf{N}(\zeta)$ *that is smaller wrt $\preceq$ than ζ, i.e. we have $\forall \zeta \in \mathcal{L}_1(\phi)$.* $\mathsf{N}(\zeta) \in \mathcal{L}_2(\phi)$ & $\mathsf{N}(\zeta)[\phi] =_{\mathcal{E}} \zeta[\phi]$ & $\mathsf{N}(\zeta) \preceq \zeta$.

We denote by $\mathsf{norm}_{\mathcal{L}_1,\mathcal{L}_2}(\phi)$ the set of normalization functions from $\mathcal{L}_1$ to $\mathcal{L}_2$ for ϕ. When $\mathcal{L}_1 = \Re$, we may use the notation $\mathsf{norm}_{\mathcal{L}_2}(\phi)$ for this set.

We say that a frame ϕ is *issued from a ($\mathcal{L}$-local) trace* if there is a process P and a ($\mathcal{L}$-local) trace $P \xrightarrow{w} P'$ such that $\phi = \mathsf{fr}(P')$. The following two propositions show under which conditions trace equivalence and local trace equivalence coincide.

Proposition 1. *Let $\mathcal{L}$ be a locality function such that, for all frames ϕ, ψ that are issued from two statically equivalent traces, we have $\mathcal{L}(\phi) = \mathcal{L}(\psi)$. Then, for all plain processes P, Q, we have $P \sim Q \implies P \simeq_{\mathcal{L}} Q$.*

Proposition 2. *Let $\mathcal{L}$ be a locality function such that, for all frames ϕ, ψ that are issued from two $\mathcal{L}$-statically equivalent and $\mathcal{L}$-local traces, there exists a normalization function $\mathsf{N} \in \mathsf{norm}_{\mathcal{L}}(\phi) \cap \mathsf{norm}_{\mathcal{L}}(\psi)$. Then, for all processes P, Q, we have $P \simeq_{\mathcal{L}} Q \implies P \sim Q$.*

To ease the construction of a normalization function for a locality function $\mathcal{L}_2$, we can introduce an intermediary locality function $\mathcal{L}_1$, with $\mathcal{L}_2(\phi) \subseteq \mathcal{L}_1(\phi) \subseteq \Re(\phi)$, and provide two normalization functions: one from $\Re$ to $\mathcal{L}_1$, and one from $\mathcal{L}_1$ to $\mathcal{L}_2$. This is the role of the following corollary:

Corollary 1. *Let $\mathcal{L}_1, \mathcal{L}_2$ be two locality functions such that,*

- *for all frames ϕ, ψ that are issued from two $\mathcal{L}_2$-statically equivalent and $\mathcal{L}_2$-local traces, there exists a normalization function* $\mathsf{N}_2 \in \mathsf{norm}_{\mathcal{L}_1,\mathcal{L}_2}(\phi) \cap \mathsf{norm}_{\mathcal{L}_1,\mathcal{L}_2}(\psi)$.
- *for all frames ϕ, ψ that are issued from two $\mathcal{L}_1$-statically equivalent and $\mathcal{L}_1$-local traces, there exists a normalization function* $\mathsf{N}_1 \in \mathsf{norm}_{\Re,\mathcal{L}_1}(\phi) \cap \mathsf{norm}_{\Re,\mathcal{L}_1}(\psi)$.
- *for all frames ϕ, ψ that are issued from two statically equivalent traces, we have* $\mathcal{L}_1(\phi) = \mathcal{L}_1(\psi)$

Then, for all processes P, Q, we have $P \simeq_{\mathcal{L}_2} Q \implies P \simeq_{\mathcal{L}_1} Q \implies P \sim Q$

4.3 Local Traces for Re-encryption

We define a locality function $\mathcal{L}_{\mathsf{renc}}$ that will allow us to infer a bound on the number of re-encryptions applied to any given ciphertext. The main idea of $\mathcal{L}_{\mathsf{renc}}$ is to disalow nested applications of the function renc in recipes. In spite of this strong restriction, $\mathcal{L}_{\mathsf{renc}}$ will admit a normalization function, because nested applications of renc can be replaced with equivalent terms that are somehow smaller, e.g. $\mathsf{renc}(\pi_1(\langle \mathsf{renc}(\zeta_1, \zeta_2), \chi\rangle, \zeta_3)$ can be replaced with $\mathsf{renc}(\zeta_1, f(\zeta_2, \zeta_3))$.

Let ϕ be a frame and $\prec$ be an occurence order on recipes associated to ϕ. For two recipes $\zeta_1, \zeta_2 \in \Re(\phi)$, we define

$$\zeta_1 \ll \zeta_2 \Leftrightarrow \zeta_2 = \mathsf{renc}(\chi_1, \chi_2) \;\&\; \zeta_1[\phi] =_{\mathcal{E}} \chi_1[\phi] \;\&\; (\zeta_1 \in \mathsf{st}(\chi_1) \;\vee\; \zeta_1 \prec \chi_1)$$

Intuitively, we have $\zeta_1 \ll \zeta_2$ if ζ_2 is a re-encryption of ζ_1. The role of the intermediary recipe χ_1 in the definition of $\ll$ is to take into account the case where ζ_2 is not a direct re-encryption of ζ_1, but there exists a context inbetween ζ_2 and ζ_1 that may dissapear by rewriting. This context may be entirely contained in χ_1, and then we have $\zeta_1 \in \mathsf{st}(\chi_1)$, or it may descent into the substitution part of $\chi_1[\phi]$, and then we have $\zeta_1 \prec \chi_1$ (note that we require $\prec$, and not simply $\preceq$).

Example 7. Let us consider the trace $P \xrightarrow{w} P_1$ of example 2. Let $\phi = \mathsf{fr}(P_1)$ and $\prec$ be the corresponding occurence order. Then, we have $\mathsf{renc}(z_1, n_1) \ll \mathsf{renc}(\pi_2(z_2), n_2)$, because $\pi_2(z_2)[\phi] = \pi_2(\langle x[\phi], x[\phi]\rangle) =_{\mathcal{E}} x[\phi] = \mathsf{renc}(z_1, n_1)[\phi]$ and $\mathsf{renc}(z_1, n_1) \prec x \prec z_2 \preceq \pi_2(z_2)$.

Now, given a frame ϕ, we can define the sets of recipes $\mathsf{RR}(\phi)$ and respectively $\mathsf{RE}(\phi)$ that represent a nested application of re-encryptions and respectively the re-encryption of an encryption. Local traces will avoid the use of such recipes. Formally, we have:

$\mathsf{RR}(\phi) = \{\zeta_0 \in \Re(\phi) \mid \exists \zeta_1 \in \Re(\phi).\ \zeta_1 \ll \zeta_0 \;\&\; \mathsf{top}(\zeta_1) = \mathsf{renc}\}$
$\mathsf{RE}(\phi) = \{\zeta_0 \in \Re(\phi) \mid \exists \zeta_1 \in \Re(\phi).\ \zeta_1 \ll \zeta_0 \;\&\; \mathsf{top}(\zeta_1) = \mathsf{enc}\}$

Definition 7 (Locality function $\mathcal{L}_{\mathsf{renc}}$). *For all frame ϕ, we let*

$$\mathcal{L}_{\mathsf{renc}}(\phi) = \{\zeta \in \Re(\phi) \mid \mathsf{st}(\zeta) \cap (\mathsf{RR}(\phi) \cup \mathsf{RE}(\phi)) = \emptyset\}$$

Example 8. Continuing example 7, we have $\mathsf{renc}(\pi_2(z_2), n_2) \notin \mathcal{L}_{\mathsf{renc}}(\phi)$, because $\mathsf{renc}(z_1, n_1) \ll \mathsf{renc}(\pi_2(z_2), n_2)$ and $\mathsf{top}(\mathsf{renc}(z_1, n_1)) = \mathsf{renc}$.

Lemma 2. *For all equational theory $\mathcal{E}$ and all frames ϕ, ψ that are issued from two statically equivalent traces, we have $\mathcal{L}_{\mathsf{renc}}(\phi) = \mathcal{L}_{\mathsf{renc}}(\psi)$.*

Lemma 3 (Normalization function $\mathsf{N}_{\mathsf{renc}}$). *Consider the locality function $\mathcal{L}_{\mathsf{renc}}$ and an equational theory $\mathcal{E} \in \mathsf{InpTh}$. For all frames ϕ, ψ that are issued from two $\mathcal{L}_{\mathsf{renc}}$-statically equivalent and $\mathcal{L}_{\mathsf{renc}}$-local traces, there exists a normalization function $\mathsf{N}_{\mathsf{renc}} \subseteq \mathsf{norm}_{\mathcal{L}_{\mathsf{renc}}}(\phi) \cap \mathsf{norm}_{\mathcal{L}_{\mathsf{renc}}}(\psi)$.*

We prove lemma 3 by replacing every recipe $\mathsf{renc}(\zeta_0, \chi_0) \in \mathsf{st}(\zeta) \cap (\mathsf{RR}(\phi) \cup \mathsf{RR}(\psi))$, such that $\mathsf{renc}(\zeta_1, \chi_1) \ll \mathsf{renc}(\zeta_0, \chi_0)$, with $\mathsf{renc}(\zeta_1, f(\chi_0, \chi_1))$. We rely on $\phi \overset{\mathsf{s}}{\simeq} \psi$ and on a well-chosen ordering of replacements to ensure that their application is consistent in both frames. None of assumptions (ae_1)-(ae_3) or (ap_1)-(ap_4) are used in the proof.

Example 9. Continuing example 8, we have $\mathsf{N}_{\mathsf{renc}}(\mathsf{renc}(\pi_2(z_2), n_2)) = \mathsf{renc}(z_1, f(n_1, n_2))$. Indeed, we have $\mathsf{renc}(z_1, f(n_1, n_2)) \in \mathcal{L}_{\mathsf{renc}}(\phi)$, $\mathsf{renc}(z_1, f(n_1, n_2))[\phi] = \mathsf{renc}(\pi_2(z_2), n_2)[\phi]$ and $\mathsf{renc}(z_1, f(n_1, n_2)) \prec \mathsf{renc}(\pi_2(z_2), n_2)$.

4.4 Local Traces for Associativity-Commutativity

We define a locality function $\mathcal{L}_f$ that will ensure a canonical use of the AC symbol f: the nested application of f-symbols always follows the same pattern and arguments of f always respect a well-chosen order.

We consider multi-hole term contexts: $\mathsf{C}[\_, \ldots, \_]_n$ is a context with n holes, and the subscript n will be dropped when n is clear from the context. For all $n \geq 1$, we let C_f^n be the n-hole context $f(\ldots f(f(\_, \_), \_) \ldots, \_)$.

Definition 8. *Assume that t is a term such that $t = \mathsf{C}[t_1, \ldots, t_n]$, with $\mathsf{sig}(\mathsf{C}) = \{f\}$ and $\mathsf{top}(t_1) \neq f, \ldots, \mathsf{top}(t_n) \neq f$. Then, we define $\mathsf{Fact}_f(t) = (t_1, \ldots, t_n)$ and $\mathsf{C}_f(t) = \mathsf{C}[\_, \ldots, \_]$. Sometimes we use the notation $\mathsf{Fact}_f(t)$ to also denote the set $\{t_1, \ldots, t_n\}$.*

For example, $\mathsf{C}_f(f(f(a,b), f(a,b))) = f(f(\_,\_), f(\_,\_))$ and $\mathsf{Fact}_f(f(f(a,b), f(a,b))) = (a,b,a,b)$.

Let ϕ be a frame and $\prec$ be the associated occurence ordering. We consider any total extension of $\prec$, denoted by $\prec_f$, that is compatible with the subterm ordering, i.e. $u \in \mathsf{st}(v) \setminus \{v\} \implies u \prec_f v$. For all $\zeta \in \mathfrak{R}(\phi)$, we define $\mathsf{min}_f^{\phi}(\zeta)$ to be the minimal wrt $\prec_f$ recipe that is equivalent to ζ in ϕ, i.e. we have $\mathsf{min}_f^{\phi}(\zeta)[\phi] =_{\mathcal{E}} \zeta[\phi]$ and $\forall \zeta' \in \mathfrak{R}(\phi).\ \zeta'[\phi] =_{\mathcal{E}} \zeta[\phi] \implies \mathsf{min}_f^{\phi}(\zeta) \prec_f \zeta'$.

Now, given a frame ϕ, we can define the set of terms $\mathsf{GF}(\phi)$ that will determine the restricted use of the symbol f:

$$\mathsf{GF}(\phi) = \{t \mid \mathsf{Fact}_f(t) = (t_1, \ldots, t_n) \implies \mathsf{C}_f(t) = \mathsf{C}_f^n \ \&\ \exists \zeta_1, \ldots, \zeta_n \in \mathfrak{R}(\phi).\ \zeta_1[\phi]{\downarrow} = t_1, \ldots, \zeta_n[\phi]{\downarrow} = t_n \ \&\ \mathsf{min}_f^{\phi}(\zeta_1) \preceq_f \ldots \preceq_f \mathsf{min}_f^{\phi}(\zeta_n)\}$$

Intuitively, $\mathsf{GF}(\phi)$ requires that its members are in canonical form wrt associativity of f, via $\mathsf{C}_f(t) = \mathsf{C}_f^n$, and in canonical form wrt commutativity of f, via

$\mathsf{min}_f^{\phi}(\zeta_1) \preceq_f \ldots \preceq_f \mathsf{min}_f^{\phi}(\zeta_n)$. Recall that, for all term u, $u{\downarrow}$ represents the normalization of u wrt to $\mathcal{R}$ only, not considering AC_f:

Definition 9 (Locality function $\mathcal{L}_f$). *For all frame ϕ, we let*

$$\mathcal{L}_f(\phi) = \{\zeta \in \Re(\phi) \mid \forall \zeta' \in \mathsf{st}(\zeta).\ \zeta'[\phi]{\downarrow} \in \mathsf{GF}(\phi)\}$$

One may wonder why, in the definition of $\mathsf{GF}(\phi)$, we compare the minimal recipes $\mathsf{min}_f^{\phi}(\zeta_i)$ and not simply the terms t_i, like in [25], or the recipes ζ_i. We do not compare the terms t_i, because they may contain information that is irrelevant to observations that can be made in a frame, in particular they may contain secret names. We want to abstract away from such details, especially since we want to relate two frames that may well be distinct on their non-observable parts. Furthermore, we do not compare the recipes ζ_i because that would not be sufficient to eliminate AC properties in a $\mathcal{L}_f$-local trace: we may have $\zeta_1 \prec_f \zeta_2$, $\chi_1 \prec_f \chi_2$, $f(\zeta_1[\phi]{\downarrow}, \zeta_2[\phi]{\downarrow}) =_{\mathsf{AC}} f(\chi_1[\phi]{\downarrow}, \chi_2[\phi{\downarrow}])$ and $f(\zeta_1[\phi]{\downarrow}, \zeta_2[\phi]{\downarrow}) \neq f(\chi_1[\phi]{\downarrow}, \chi_2[\phi]{\downarrow})$. On the other hand, comparing $\mathsf{min}_f^{\phi}(\zeta_i)$ ensures an ordering on equivalence classes, and not merely an ordering on recipes.

Example 10. Consider the frames $\phi = \nu a.\nu b.\ \{x_1 \mapsto a, x_2 \mapsto b, x_3 \mapsto \langle b, a\rangle\}$ and $\psi = \nu a.\nu b.\ \{x_1 \mapsto b, x_2 \mapsto a, x_3 \mapsto \langle a, b\rangle\}$, such that $x_1 \prec x_2 \prec x_3$. Then, $\mathsf{min}_f^{\phi}(\pi_1(x_3)) = x_1$ and $\mathsf{min}_f^{\phi}(\pi_2(x_3)) = x_2$. The recipes $f(x_1, f(x_1, x_1)), f(x_2, x_1)$ and $f(\pi_1(x_3), \pi_2(x_3))$ are not in $\mathcal{L}_f(\phi)$ and not in $\mathcal{L}_f(\psi)$.

In example 9, if we assume $n_2 \prec_f n_1$, the recipe $\mathsf{renc}(z_1, f(n_1, n_2))$ is in $\mathcal{L}_{\mathsf{renc}}(\phi) \setminus \mathcal{L}_f(\phi)$.

Definition 10 (Locality function $\mathcal{L}_{\mathsf{rf}}$). *For all frame ϕ, we let*

$$\mathcal{L}_{\mathsf{rf}}(\phi) = \mathcal{L}_{\mathsf{renc}}(\phi) \cap \mathcal{L}_f(\phi)$$

The following lemma is crucial in defining a normalization function for $\mathcal{L}_f$, because it will allow us to obtain recipes of terms in $\mathsf{Fact}_f(\zeta[\phi]{\downarrow})$, that we can re-arrange to transform a non-local recipe into a local one:

Lemma 4. *Let P, Q be processes such that $P \xrightarrow{w} Q$ and let $\phi = \mathsf{fr}(Q)$. Then, for all recipe $\zeta \in \Re(\phi)$ such that $\zeta[\phi]{\downarrow} = f(t_1, t_2)$ there exist $f(\zeta_1, \zeta_2) \in \Re(\phi)$ such that*

- *$f(\zeta_1, \zeta_2) \in \mathsf{st}(\zeta)$ or $f(\zeta_1, \zeta_2) \in \mathsf{st}(\mathsf{inp}(w))$ & $f(\zeta_1, \zeta_2) \prec \zeta$*
- *and $f(\zeta_1, \zeta_2)[\phi]{\downarrow} = f(t_1, t_2)$*

Note that $f(\zeta_1, \zeta_2) \in \Re(\phi) \Leftrightarrow \{\zeta_1, \zeta_2\} \subseteq \Re(\phi)$. The proof of lemma 4 relies on assumptions (ae_2),(ae_3) and (ap_1) to deduce that, whenever $f(t_1, t_2)$ is deducible in a trace, it must be the case that both t_1 and t_2 are deducible.

Lemma 5. *For all equational theory $\mathcal{E} = (\mathcal{R}, \emptyset) \in \mathsf{OutTh}$ and all frames ϕ, ψ that are issued from two statically equivalent traces, we have $\mathcal{L}_f(\phi) = \mathcal{L}_f(\psi)$.*

The proof of lemma 5 is eased by the absence of AC symbols, and it relies on lemma 4 and $\phi \overset{s}{\sim} \psi$ to transfer the $\mathcal{L}_f$-locality of a recipe from ϕ to ψ.

Lemma 6 (Normalization function N_f). *Consider the locality functions $\mathcal{L}_{\mathsf{renc}}, \mathcal{L}_{\mathsf{rf}}$ and an equational theory $\mathcal{E} \in \mathsf{InpTh}$. For all frames ϕ, ψ that are issued from two $\mathcal{L}_{\mathsf{rf}}$-statically equivalent and $\mathcal{L}_{\mathsf{rf}}$-local traces, there exists a normalization function $\mathsf{N}_f \subseteq \mathsf{norm}_{\mathcal{L}_{\mathsf{renc}},\mathcal{L}_{\mathsf{rf}}}(\phi) \cap \mathsf{norm}_{\mathcal{L}_{\mathsf{renc}},\mathcal{L}_{\mathsf{rf}}}(\psi)$.*

We prove lemma 6 by replacing all $\zeta' \in \mathsf{st}(\zeta)$ such that $\zeta'[\phi]{\downarrow} \notin \mathsf{GF}(\phi)$ and $\mathsf{Fact}_f(\zeta'[\phi]{\downarrow}) = (t_1, \ldots, t_n)$ with an equivalent ζ'' such that $\zeta''[\phi]{\downarrow} \in \mathsf{GF}(\phi)$, to obtain an $\mathcal{L}_f$-local recipe. To construct ζ'', we start with a sequence of recipes $\zeta_1, \ldots, \zeta_n$ such that $\zeta_1[\phi]{\downarrow} = t_1, \ldots, \zeta_n[\phi]{\downarrow} = t_n$, whose existence is ensured by lemma 4. Then, we consider $\zeta'' = \mathsf{C}^n_f[\zeta_{i_1}, \ldots, \zeta_{i_n}]$, where $\zeta_{i_1}, \ldots, \zeta_{i_n}$ is a reordering of $\zeta_1, \ldots, \zeta_n$ such that $\mathsf{min}^{\phi}_f(\zeta_{i_1}) \prec_f \ldots \prec_f \mathsf{min}^{\phi}_f(\zeta_{i_n})$. We rely on $\phi \overset{\mathsf{s}}{\simeq} \psi$ to ensure that these replacements are consistent in both frames ϕ and ψ.

Example 11. Continuing example 10, we have $\mathsf{N}_f(f(x_1, f(x_1, x_1)) = f(f(x_1, x_1), x_1)$, $\mathsf{N}_f(f(x_2, x_1)) = f(x_1, x_2)$ and $\mathsf{N}_f(f(\pi_1(x_3), \pi_2(x_3))) = f(x_1, x_2)$.

4.5 Main Results of This Section

From proposition 1, lemma 2 and lemma 5, we have

Corollary 2. *Consider the locality function $\mathcal{L}_{\mathsf{rf}}$ and any equational theory $\mathcal{E}_m \in \mathsf{OutTh}$. Then, for all plain processes P, Q, we have $P \sim Q \implies P \simeq_{\mathcal{L}_{\mathsf{rf}}} Q$.*

From corollary 1 (applied to $\mathcal{L}_{\mathsf{renc}}$ and $\mathcal{L}_{\mathsf{rf}}$) and lemmas 6, 3 and 2, we have:

Corollary 3. *Consider the locality function $\mathcal{L}_{\mathsf{rf}}$ and any equational theory $\mathcal{E} \in \mathsf{InpTh}$. Then, for all plain processes P, Q, we have $P \simeq_{\mathcal{L}_{\mathsf{rf}}} Q \implies P \sim Q$.*

To bridge the gap between $P \sim_{\mathcal{E}_m} Q$ and $P \sim_{\mathcal{E}} Q$ it is sufficient now to show that $P \simeq_{\mathcal{L}_{\mathsf{rf}},\mathcal{E}_m} Q \implies P \simeq_{\mathcal{L}_{\mathsf{rf}},\mathcal{E}} Q$. This is the subject of the next section.

5 Reduction of Equational Theories

In this section, we use $\mathcal{R}$ for a rewrite system corresponding to a theory in InpTh, and $\mathcal{R}_m$ for a corresponding rewrite system of a theory in OutTh. Lemma 7 simplifies reasoning modulo AC, by showing that AC_f does not interfere with rewriting. The proof relies on assumption (ae_1), in particular on the fact that f does not occur on the left-hand side of rewrite rules:

Lemma 7. *For any equational theory $\mathcal{E} \in \mathsf{InpTh}$ and for all terms u, v, we have $u \to^*_{\mathsf{AC}} v$ if and only if $u \to^* v'$, for some term v' such that $v' =_{\mathsf{AC}} v$.*

Lemma 8 simplifies reasoning modulo re-encryption, by showing that nonces from the protocol always stay secret:

Lemma 8. *For all process P that satisfies (ap_1)-(ap_4) and all equational theory $\mathcal{E}$ that satisfies (ae_1)-(ae_3), we have $\mathsf{ran}(P) \subseteq \mathcal{S}(P, \mathcal{E})$.*

The proof relies on assumptions (ap_2) and (ap_3), to ensure that elements of $\mathsf{ran}(P)$ are handled in a restricted way. Then, assumption (ae_3) and the definitions of $\mathsf{st}_p, \mathsf{W}(\mathcal{E})$ ensure that this indeed guarantees secrecy.

We will show that $\mathcal{L}_{\mathsf{rf}}$-local traces have a bounded *re-encryption depth*. To define it, we must identify the chain of re-encryptions that have been applied to obtain a given ciphertext. This chain will be the limit of recursively identifying *re-encryption witnesses*:

Definition 11 (Re-encryption witness). *Assume $u \to_{\mathcal{R}}^* v$. Then, for all term $t \in \mathsf{st}(v)$ with $\mathsf{top}(t\downarrow) \in \{\mathsf{enc}, \mathsf{renc}\}$, a re-encryption witness for t in u is a term $\mathsf{rw}_u(t) \in \mathsf{st}(u)$, such that $\mathsf{rw}_u(t) \to_{\mathcal{R}}^* t\downarrow$ and $\mathsf{top}(\mathsf{rw}_u(t)) \in \{\mathsf{enc}, \mathsf{renc}\}$.*

For intuition, note that $u = \mathsf{C}'[\mathsf{rw}_u(t)] \to_{\mathcal{R}}^* \mathsf{C}[t] = v$, for some contexts C and C'.

Example 12. Let $u = \pi_1(\langle \mathsf{renc}(\pi_1(\langle \mathsf{enc}(a, b, r_1), c\rangle), r_2), c\rangle)$. We have $u \to v_1 \to^* v_2$ where $v_1 = \pi_1(\langle \mathsf{renc}(\mathsf{enc}(a, b, r_1), r_2), c\rangle)$ and $v_2 = \mathsf{enc}(a, b, f(r_1, r_2))$. Then, for $t = \mathsf{enc}(a, b, r_1) \in \mathsf{st}(v_1)$, we have $\mathsf{rw}_u(t) = t$. For $t = \mathsf{enc}(a, b, f(r_1, r_2)) \in \mathsf{st}(v_2)$, we have $\mathsf{rw}_{v_1}(t) = \mathsf{renc}(\mathsf{enc}(a, b, r_1), r_2)$ and $\mathsf{rw}_u(t) = \mathsf{renc}(\pi_1(\langle \mathsf{enc}(a, b, r_1), c\rangle), r_2)$.

Lemma 9. *Assume $u \to_{\mathcal{R}}^* v$. Then, for all term $t \in \mathsf{st}(v)$ with $\mathsf{top}(t\downarrow) \in \{\mathsf{enc}, \mathsf{renc}\}$, there always exists a re-encryption witness $\mathsf{rw}_u(t)$.*

In particular, the previous lemma shows that for all term t with $\mathsf{top}(t\downarrow) \in \{\mathsf{enc}, \mathsf{renc}\}$ there exists a re-encryption witness of $t\downarrow$ in t, that is $\mathsf{rw}_t(t\downarrow)$.

Definition 12 (Re-encryption depth). *For all term t, we define its re-encryption depth $\mathsf{rd}(t)$ as follows:*

- $\mathsf{rd}(t) = 0$, *if* $\mathsf{top}(t\downarrow_{\mathcal{R}}) \notin \{\mathsf{enc}, \mathsf{renc}\}$
- $\mathsf{rd}(t) = 1$, *if* $\mathsf{top}(t\downarrow_{\mathcal{R}}) = \mathsf{enc}$ *and* $\mathsf{top}(\mathsf{rw}_t(t\downarrow)) = \mathsf{enc}$
- $\mathsf{rd}(t) = \mathsf{rd}(t') + 1$, *if* $\mathsf{top}(t\downarrow_{\mathcal{R}}) \in \{\mathsf{enc}, \mathsf{renc}\}$ *and* $\mathsf{rw}_t(t\downarrow) = \mathsf{renc}(t', t'')$

Example 13. We have $\mathsf{rd}(\mathsf{renc}(\pi_1(\langle \mathsf{renc}(\mathsf{enc}(a, b, r_1), r_2), \mathsf{renc}(a, r_0)\rangle), r_3)) = 3$. Continuing example 12, we have $\mathsf{rd}(u) = 2$.

Next we show that for all term u with a re-encryption depth bounded by m, its normal form modulo $\mathcal{R}_m$ is the normal form modulo $\mathcal{R}_m$ of its re-encryption witness modulo $\mathcal{R}$:

Lemma 10. *Let u be a term such that, for all $t \in \mathsf{st}(u)$, $\mathsf{rd}(t) \leq m + 1$. Then, for all term v such that $u \to_{\mathcal{R}}^* v$ following a bottom-up rewriting sequence, we have $u \to_{\mathcal{R}_m}^* v\rho$ where*

$$\rho = \{t \mapsto \mathsf{rw}_u(t)\downarrow_{\mathcal{R}_m} \mid t \in \mathsf{st}(v) \ \&\ \mathsf{top}(t) \in \{\mathsf{enc}, \mathsf{renc}\} \ \&\ t = t\downarrow_{\mathcal{R}}\}$$

Example 14. Consider the terms $u = \mathsf{renc}(\pi_1\langle \mathsf{enc}(a, \mathsf{pub}(k), r_1), a\rangle, r_2)$ and $u_1 = \mathsf{dec}(u, k)$. Then, $\mathsf{rw}_u(u\downarrow_{\mathcal{R}}) = u$, and we obviously have $u\downarrow_{\mathcal{R}_m} = (u\downarrow_{\mathcal{R}})\rho = \mathsf{rw}_u(u\downarrow_{\mathcal{R}})\downarrow_{\mathcal{R}_m} = \mathsf{renc}(\mathsf{enc}(a, \mathsf{pub}(k), r_1), r_2)$. On the other hand, we have $u_1 \to_{\mathcal{R}_m} \mathsf{dec}(u\downarrow_{\mathcal{R}_m}, k) \to_{\mathcal{R}_m} a = u_1\downarrow_{\mathcal{R}} = u_1\downarrow_{\mathcal{R}}\rho$, where we have used the rule $\mathsf{dec}(\mathsf{renc}(\mathsf{enc}(x, \mathsf{pub}(y), z_0), z_1), y) \to x$ for the last rewriting step.

The following lemma bridges the gap backwards, from $\to^*_{\mathcal{R}_m}$ to $\to^*_{\mathcal{R}}$, and is an easy consequence of assumption (ae_3):

Lemma 11. *For all terms u, v such that $u \to^*_{\mathcal{R}_m} v$, we have $u \to^*_{\mathcal{R}} v$.*

Relying on (ap_2)-(ap_4) and lemma 8, we can show that for all trace $P \xrightarrow{w} Q$ and all nonce $t \in \mathsf{ran}(P)$, there exists a unique term $\mathsf{ciph}(t)$ such that $t = \mathsf{ran}(\mathsf{ciph}(t))$ and $\mathsf{ciph}(t) = u[\mathsf{fr}(Q)]$, for some term $u \in \mathsf{st}(P) \cup \mathcal{L}_{\mathsf{rf}}(\mathsf{fr}(Q))$.

Lemma 12. *Let $P \xrightarrow{w}_{\mathcal{R}} Q$ be a $\mathcal{L}_{\mathsf{rf}}$-local trace and $\phi = \mathsf{fr}(Q)$. Let T be a term in $\mathsf{st}(P) \cup \mathcal{L}_{\mathsf{rf}}(\mathsf{fr}(Q))$. Then, for all $t \in \mathsf{st}(T[\phi])$, we have:*

(a) $\mathsf{rd}(t) \leq 2 * |\mathsf{ran}_r(P)| + 2$
(b) $\mathsf{rd}(t) > 1 \implies \mathsf{Fact}_f(\mathsf{ran}(t{\downarrow})) \cap \mathsf{ran}(P) \neq \emptyset$
(c) $\mathsf{rw}_t(t{\downarrow}) = \mathsf{renc}(u, v) \implies$
- *either* $v \in \mathsf{ran}(P)$ *and* $\mathsf{renc}(u, v) = \mathsf{ciph}(v)$
- *or else* $\mathsf{Fact}_f(v{\downarrow}) \cap \mathsf{ran}(P) = \emptyset$ *and*
  - *either* $\mathsf{top}(u{\downarrow}) \notin \{\mathsf{enc}, \mathsf{renc}\}$
  - *or else* $\exists t' \in \mathsf{ran}(P).\ \mathsf{rw}_u(u{\downarrow}) = \mathsf{ciph}(t')$

The point (a) is obviously useful because it bounds the re-encryption depth of all term t that occurs in $\mathcal{L}_{\mathsf{rf}}$-local traces. It is proved by considering any chain of re-encryption witnesses starting from $\mathsf{rw}_t(t{\downarrow})$ and showing that each re-encryption performed by the environment (i.e. in recipes) must be followed by a re-encryption performed in P. Because $\mathsf{ciph}(t)$ is unique for all $t \in \mathsf{ran}_r(P)$, we can deduce from assumption (ap_4) that the total number of re-encryptions performed by P for every single ciphertext is bounded by $|\mathsf{ran}_r(P)|$, thus we deduce a bound of $2 * |\mathsf{ran}_r(P)| + 1$ for the length of any re-encryption chain. The points (b) and (c) will be useful to show that equalities of terms with positive re-encryption depth transfer to equalities of their respective re-encryption witnesses. In conjunction with lemma 10, this will allow us to transfer equalities modulo $\mathcal{E}$ to equalities modulo $\mathcal{E}_m$.

Finally, we prove a lemma showing that equivalence classes of $\mathcal{E}$ and $\mathcal{E}_m$ are the same in local traces:

Lemma 13. *Let $\mathcal{E} \in \mathsf{InpTh}$ and $P \xrightarrow{w}_{\mathcal{E}} Q$ be a $\mathcal{L}^{\mathcal{E}}_{\mathsf{rf}}$-local trace. Let $\phi = \mathsf{fr}(Q)$, $m = 2 * |\mathsf{ran}_r(P)| + 1$ and $\mathcal{E}_m \in \mathsf{OutTh}$ be the reduced theory that corresponds to $\mathcal{E}$ and m. Then,*

A. *for all terms $u_1, u_2 \in \mathsf{st}(P) \cup \mathcal{L}_{\mathsf{rf}}(\mathsf{fr}(Q))$, we have $u_1[\phi] =_{\mathcal{E}} u_2[\phi] \implies u_1[\phi] =_{\mathcal{E}_m} u_2[\phi]$*
B. *for all terms u_1, u_2, we have $u_1[\phi] =_{\mathcal{E}_m} u_2[\phi] \implies u_1[\phi] =_{\mathcal{E}} u_2[\phi]$*

To prove A, we first eliminate the AC equations, relying on lemma 7, the point (c) of lemma 12, lemma 8 and the definition of $\mathcal{L}_f$. This gives us $u_1[\phi] =_{\mathcal{E}} u_2[\phi] \implies u_1[\phi]{\downarrow_{\mathcal{R}}} = u_2[\phi]{\downarrow_{\mathcal{R}}}$. To complete the reduction, we first use the point (a) of lemma 12 and lemma 10 to show that $u_1[\phi]{\downarrow_{\mathcal{R}_m}} = u_1[\phi]{\downarrow_{\mathcal{R}}}\rho_1$ and $u_2[\phi]{\downarrow_{\mathcal{R}_m}} = u_2[\phi]{\downarrow_{\mathcal{R}}}\rho_2$, for some replacements ρ_1, ρ_2. Then, we use the points (b),(c) of lemma 12 to show that we must have $\rho_1 = \rho_2$. Therefore, we have $u_1[\phi]{\downarrow_{\mathcal{R}}} = u_2[\phi]{\downarrow_{\mathcal{R}}} \implies u_1[\phi]{\downarrow_{\mathcal{R}_m}} = u_2[\phi]{\downarrow_{\mathcal{R}_m}}$. As a consequence of lemma 13, we have:

Proposition 3. *Consider given the locality function* $\mathcal{L}_{\mathsf{rf}}$ *and* $\mathcal{E} \in \mathsf{InpTh}$. *Then, for all plain processes* P, Q, *we have* $P \simeq_{\mathcal{E}_m} Q \Leftrightarrow P \simeq_{\mathcal{E}} Q$, *where* $m = 2 * \mathsf{max}(|\mathsf{ran}_r(P)|, |\mathsf{ran}_r(Q)|) + 1$.

We conclude the proof of theorem 1 as a consequence of corollary 2, proposition 3 and corollary 3.

Application to vote privacy in Prêt à Voter. We have $\mathcal{E}_{\mathsf{PaV}} = (\mathcal{R}_{\mathsf{PaV}}, \mathsf{AC}_f)$, with $\mathcal{R}_{\mathsf{PaV}} = \mathcal{R}_{\mathsf{DY}} \cup \mathcal{R}_{\mathsf{renc}} \cup \mathcal{R}_{\mathsf{decP}} \cup \mathcal{R}_{\mathsf{mixP}} \cup \mathcal{R}_{\mathsf{vote}}$. Each of these five rewrite systems is AC-convergent, and so is the system $\mathcal{R}_{\mathsf{PaV}} \setminus \mathcal{R}_{\mathsf{mixP}}$. However, $\mathcal{R}_{\mathsf{PaV}}$ is not AC-confluent (and therefore not AC-convergent) due to critical pairs between rules in $\mathcal{R}_{\mathsf{renc}}$ and rules in $\mathcal{R}_{\mathsf{mixP}}$. The system $\mathcal{R}_{\mathsf{PaV}}$ can be made AC-convergent by completion, but this introduces other problems, notably violation of conditions (ae_1)-(ae_3) and the addition of a significant number of new rules, that may pose problems for ProVerif. That is why our current analisys does not cover the mixnet proofs in $\mathcal{R}_{\mathsf{mixP}}$ - we leave it as a subject for future work.

Note that $|\mathsf{ran}_r(P_{\mathsf{PaV}}\{v_1 \mapsto a\}\{v_2 \mapsto b\})| = |\mathsf{ran}_r(P_{\mathsf{PaV}}\{v_1 \mapsto b\}\{v_2 \mapsto a\})| = 2$, corresponding to the re-encryption of ballots from idA and idB, and we deduce a bound $m = 5$ in the application of Theorem 1. Therefore, from Theorem 1 and the result returned by the ProVerif code available online, we conclude

Corollary 4. *Prêt à Voter (without mixnet proofs) satisfies vote privacy for two eligible voters and two candidates.*

6 Conclusion and Future Work

Note that proposition 3 shows that abstracting $\mathcal{E}$ with $\mathcal{E}_m$ is not only sound, but also complete in local traces. This means that, to derive the completeness of our reduction for the full set of traces, we only have to extend lemma 5 to theories in InpTh and lemma 3 to theories in OutTh. In conjunction with lemma 1 and [5,15,10], this would lead to a first decision procedure for trace equivalence outside the class of subterm-convergent systems.

To be really faithul to algebraic properties of ElGamal re-encryption, AC_f should probably be extended to AG_f, unless e.g. it is computationally sound to consider only AC_f [18].

It would be nice to see how some of our restrictions can be lifted, in particular the quite strong restrictions on the occurence of the AC symbol in the protocol and in the theory. The restriction not to have re-encryptions in replicated processes, (ap_4), looks natural for the reduction that we have in mind, but is it really necessary in order to be able to handle re-encryption automatically?

Observational equivalence is stronger than trace equivalence, but it is unclear whether it is more appropriate for definitions of security. In any case, our reduction could also be used for verification of observational equivalence, and we would have to consider a notion of local trees instead of local traces for the correctness proof.

This paper shows that there exists an interesting relation between restrictions of the set of traces and restrictions of the equational theory. We have exhibited this relation only for two particular algebraic properties, and it would be interesting to see how it can be formulated in general.

References

1. Abadi, M., Cortier, V.: Deciding knowledge in security protocols under equational theories. Theoretical Computer Science 367(1-2), 2–32 (2006)
2. Abadi, M., Fournet, C.: Mobile values, new names, and secure communication. In: Proceedings of the 28th ACM Symposium on Principles of Programming Languages (POPL 2001), pp. 104–115 (January 2001)
3. Abadi, M., Gordon, A.D.: A calculus for cryptographic protocols: the spi calculus. Information and Computation 148(1) (1999)
4. Arapinis, M., Chothia, T., Ritter, E., Ryan, M.: Analysing unlinkability and anonymity using the applied pi calculus. In: CSF, pp. 107–121. IEEE Computer Society (2010)
5. Baudet, M.: Deciding security of protocols against off-line guessing attacks. In: Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS 2005), Alexandria, Virginia, USA, pp. 16–25. ACM Press (November 2005)
6. Baudet, M., Cortier, V., Delaune, S.: YAPA: A Generic Tool for Computing Intruder Knowledge. In: Treinen, R. (ed.) RTA 2009. LNCS, vol. 5595, pp. 148–163. Springer, Heidelberg (2009)
7. Blanchet, B.: Automatic proof of strong secrecy for security protocols. In: IEEE Symposium on Security and Privacy, pp. 86–100 (2004)
8. Blanchet, B., Abadi, M., Fournet, C.: Automated verification of selected equivalences for security protocols. Journal of Logic and Algebraic Programming 75(1), 3–51 (2008)
9. Cheval, V., Comon-Lundh, H., Delaune, S.: Automating Security Analysis: Symbolic Equivalence of Constraint Systems. In: Giesl, J., Hähnle, R. (eds.) IJCAR 2010. LNCS, vol. 6173, pp. 412–426. Springer, Heidelberg (2010)
10. Cheval, V., Comon-Lundh, H., Delaune, S.: Trace equivalence decision: Negative tests and non-determinism. In: Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS 2011), Chicago, Illinois, USA. ACM Press (October 2011)
11. Ciobâcă, Ş., Delaune, S., Kremer, S.: Computing knowledge in security protocols under convergent equational theories. Journal of Automated Reasoning (2011)
12. Comon-Lundh, H., Cortier, V.: Security Properties: Two Agents Are Sufficient. In: Degano, P. (ed.) ESOP 2003. LNCS, vol. 2618, pp. 99–113. Springer, Heidelberg (2003)
13. Comon-Lundh, H., Delaune, S.: The Finite Variant Property: How to Get Rid of Some Algebraic Properties. In: Giesl, J. (ed.) RTA 2005. LNCS, vol. 3467, pp. 294–307. Springer, Heidelberg (2005)
14. Corin, R., Doumen, J., Etalle, S.: Analysing password protocol security against off-line dictionary attacks. Electr. Notes Theor. Comput. Sci. 121, 47–63 (2005)
15. Cortier, V., Delaune, S.: A method for proving observational equivalence. In: Proceedings of the 22nd IEEE Computer Security Foundations Symposium (CSF 2009), Port Jefferson, NY, USA, pp. 266–276. IEEE Computer Society Press (July 2009)

16. Cortier, V., Delaune, S.: Decidability and combination results for two notions of knowledge in security protocols. Journal of Automated Reasoning (2011)
17. Cortier, V., Delaune, S., Lafourcade, P.: A survey of algebraic properties used in cryptographic protocols. Journal of Computer Security 14(1), 1–43 (2006)
18. Cortier, V., Kremer, S., Warinschi, B.: A survey of symbolic methods in computational analysis of cryptographic systems. Journal of Automated Reasoning 46(3-4), 225–259 (2010)
19. Cortier, V., Smyth, B.: Attacking and fixing helios: An analysis of ballot secrecy. In: Proc. of the 24th IEEE Computer Security Foundations Symposium, pp. 297–311 (2011)
20. Delaune, S., Kremer, S., Ryan, M.D.: Verifying privacy-type properties of electronic voting protocols. Journal of Computer Security 17(4), 435–487 (2009)
21. Dershowitz, N., Jouannaud, J.-P.: Rewrite systems. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, vol. B, pp. 243–309. North-Holland (1990)
22. Durante, L., Sisto, R., Valenzano, A.: Automatic testing equivalence verification of spi calculus specifications. ACM Trans. Softw. Eng. Methodol. 12(2), 222–284 (2003)
23. Kremer, S., Mercier, A., Treinen, R.: Reducing equational theories for the decision of static equivalence. Journal of Automated Reasoning (2011)
24. Küsters, R., Truderung, T.: Using proverif to analyze protocols with Diffie-Hellman exponentiation. In: CSF, pp. 157–171. IEEE Computer Society (2009)
25. Küsters, R., Truderung, T.: Reducing protocol analysis with XOR to the XOR-free case in the horn theory based approach. J. Autom. Reasoning 46(3-4), 325–352 (2011)
26. Lynch, C., Meadows, C.: Sound Approximations to Diffie-Hellman Using Rewrite Rules. In: López, J., Qing, S., Okamoto, E. (eds.) ICICS 2004. LNCS, vol. 3269, pp. 262–277. Springer, Heidelberg (2004)
27. Mödersheim, S.: Diffie-Hellman without difficulty. In: FAST (2011)
28. Ryan, P.Y.A., Schneider, S.A.: Prêt à Voter with Re-encryption Mixes. In: Gollmann, D., Meier, J., Sabelfeld, A. (eds.) ESORICS 2006. LNCS, vol. 4189, pp. 313–326. Springer, Heidelberg (2006)
29. Tiu, A., Dawson, J.E.: Automating open bisimulation checking for the spi calculus. In: Proc. of the 23rd IEEE Computer Security Foundations Symposium, pp. 307–321 (2010)