Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Electronic Notes in Theoretical Computer Science 173 (2007) 139–157
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

A Logical Characterisation of Static Equivalence

Hans Hüttel[1] and Michael D. Pedersen[2,b]

BRICS, Department of Computer Science
Aalborg University, Denmark

[b] *LFCS,School of Informatics*
University of Edinburgh, Scotland

Abstract

The work of Abadi and Fournet introduces the notion of a *frame* to describe the knowledge of the environment of a cryptographic protocol. Frames are lists of terms; two frames are indistinguishable under the notion of *static equivalence* if they satisfy the same equations on terms. We present a first-order logic for frames with quantification over environment knowledge which, under certain general conditions, characterizes static equivalence and is amenable to construction of characteristic formulae. The logic can be used to reason about environment knowledge and can be adapted to a particular application by defining a suitable signature and associated equational theory. The logic can furthermore be extended with modalities to yield a modal logic for e.g. the Applied Pi calculus.

Keywords: Frames, static equivalence, logic, cryptographic protocols, Applied Pi

1 Introduction

Formal approaches to the design and analysis of security protocols often rely on a notion of environment knowledge. Following Abadi and Fournet, [3], a *frame* is a substitution with name restriction which represents the messages communicated by principals at any given state in a communication protocol and hence an implicit representation of environment knowledge. Consider for example the frame

$$\varphi_1 \triangleq (\nu k)\{^{enc(b,k^+)}/_{x_1}, {}^{k^-}/_{x_2}\}$$

which represents the environment knowledge in a state of a protocol where two messages have been sent on some open channel: a name b encrypted with a public

[1] Email: hans@cs.aau.dk

[2] Email: m.d.pedersen@sms.ed.ac.uk. This work was supported in part by Microsoft Research through the European PhD Scholarship Programme.

doi:10.1016/j.entcs.2007.02.032

key k^+ and the corresponding private key k^-. The intuition that the seed k should be secret is captured by the restriction (νk) on k. Individual terms can be referred through their respective variables which provides a means of ordering terms.

Arbitrary terms can be built from names and variables in a frame, and equality between terms is defined on a per-application basis by an appropriate *equational theory*, $=_{\mathcal{E}}$. In the frame φ_1, which relies on public-key encryption, one would thus expect that e.g. $dec(x_1, x_2) =_{\mathcal{E}} b$. Two frames are *statically equivalent* if they cannot be distinguished by testing for equality between arbitrary terms built from variables and free names in the frames. Take for example two additional frames, where h is a one-way hash function:

$$\varphi_1' \triangleq (\nu k)\{{}^{enc(b,h(k)^+)}/_{x_1}, {}^{h(k)^-}/_{x_2}\} \qquad \varphi_1'' \triangleq (\nu k)\{{}^{enc(h(b),k^+)}/_{x_1}, {}^{k^-}/_{x_2}\}$$

The two frames differ from φ_1 only in the hash function being applied to the keys and to the clear text b, respectively. We then have that φ_1 and φ_1' are statically equivalent because the same equalities hold in both frames; in particular, the equality $dec(x_1, x_2) =_{\mathcal{E}} b$ holds. On the other hand, this equality does not hold in φ_1'', so φ_1'' is neither statically equivalent to φ_1 nor φ_1'.

In addition to being a self-contained representation of knowledge, frames and static equivalence play a central role in process calculi such as the Applied π calculus [3] and the Spi calculus [4,6] since observational equivalence on processes is defined using standard bisimulation conditions and a condition of static equivalence on the frames arising from processes. The standard definition of static equivalence does however suffer under universal quantification over terms, which raises decidability issues and which complicates a logical characterisation. In fact it has been shown that static equivalence is undecidable for some equational theories, but a class of theories for which static equivalence is decidable is also known [1,2,7].

The main contribution of this paper is a first order logic for frames which characterizes static equivalence and yields characteristic formulae under certain general conditions, namely when the equational theory is an independent convergent subterm theory and allows term reductions to be observable. The logic includes atomic propositions for testing equality (in the equational theory), reductions and syntactic equality between terms. Quantification ranges over the *synthesis* of a frame which, intuitively, is a direct representation of environment knowledge. Hence the logic can be used to reason about environment knowledge and can be adapted to a particular application by defining a suitable function signature and equational theory.

Extending the logic with modalities [9] yields a modal logic for Applied π which characterizes observational equivalence on processes and which allows reasoning about environment knowledge over time. The modal logic can again be adapted to particular applications; for instance, defining an equational theory with private key cryptography results in a logic resembling the Spi logics [8].

In order to obtain the logical characterisation results we first give a refined definition of static equivalence which does not rely on universal quantification over arbitrary terms. It is shown that the refined definition coincides with the standard definition under the general conditions noted above. Our approach is inspired by

the notion of cores introduced in [6] for the Spi calculus.

The paper is structured as follows. In Section 2 we introduce some basic notions and formalise our assumptions on equational theories. Section 3 develops a refined definition of static equivalence which does not rely on universal quantification over arbitrary terms. In Section 4 we present the logic for frames, and in section 5 we briefly show how this can be extended to a modal logic for Applied π. Finally Section 6 concludes.

2 Basic Definitions

2.1 Terms

A signature Σ consists of a finite set of function symbols each associated with an integer arity. Let Σ^k be the function symbols in Σ with arity k, let $\mathcal{N}$ be the set of names, ranged over by $a, b, c, \ldots, k$, let $\mathcal{X}$ be the set of variables, ranged over by x, y, z, and let $\mathcal{U} = \mathcal{N} \cup \mathcal{X}$. Then the set of terms $T(\Sigma, \mathcal{U})$ is defined inductively as follows:

- $\mathcal{U} \subseteq T(\Sigma, \mathcal{U})$.
- for all $k \geq 0$: $f(M_1, \ldots, M_k) \in T(\Sigma, \mathcal{U})$ if $f \in \Sigma^{(k)}$ and $M_1, \ldots, M_k \in T(\Sigma, \mathcal{U})$

The set of names, respectively variables, occurring in a term $M \in T(\Sigma, \mathcal{U})$ will be denoted by $n(M)$, respectively $v(M)$, and we will use the abbreviation $n(M_1, M_2)$ to denote $n(M_1) \cup n(M_2)$. A *context* $C[\tilde{x}]$ is a term where $v(C[\tilde{x}]) = \tilde{x}$ and $n(C[\tilde{x}]) = \emptyset$.

Often we will be interested in identifying substrings at certain positions of a term $M \in T(\Sigma, \mathcal{U})$. For this purpose the structure of M can be illustrated by representing it as a parse tree, where nodes are labelled by elements of $\Sigma \cup \mathcal{U}$, nodes labelled by $\mathcal{U}$ are leaves and the children of a node labelled by a function symbol f are the arguments of f. A position is then a finite sequence $w = w_1, \ldots, w_n$ where $w_i \in \mathbb{N}$. The position w identifies the node found by traversing the tree from the root and for each node at level i following the edge numbered w_i. The subterm of M whose parse tree is rooted at w is denoted $M|_w$ (a formal definition of positions and subterms is given in [11]).

2.2 The Equational Theory

Following [1], we define equality between terms based on a term rewrite system. A *rewrite rule* r is of the form $L >_r R$ where $L, R \in T(\Sigma, \mathcal{X})$ and $v(R) \subseteq v(L)$. A term M_1 *reduces primitively* to M_2 using rule $L >_r R$, written $M_1 >_r M_2$, if $M_1 = L\theta$ and $M_2 = R\theta$ for some substitution θ. A *term rewrite system* $\mathcal{R}$ is then a set of rewrite rules, and we are interested in the rewrite relation $>$ induced by the rewrite system. Define $M_1 > M_2$ if and only if there is a rule $L >_r R$ in $\mathcal{R}$ such that $M_1|_w = L\theta$ and and $M_2 = M_1\{^{R\theta}/_{M_1|_w}\}$ for some substitution θ and position w. The equational theory $=_{\mathcal{E}}$ is now given by the reflexive, symmetric and transitive closure of the reduction relation $>$.

We consider applications with encryption and pairs as a running example.

Example 2.1 For applications with symmetric key encryption and pairs the following self-explanatory signature Σ_{sym} can be adopted:

$$\Sigma_{sym} \triangleq \{enc(\cdot,\cdot), dec(\cdot,\cdot), [\cdot,\cdot], fst(\cdot), snd(\cdot)\}$$

The equational theory $\mathcal{E}S$ is generated from the following rules which specify how decryption and projection works:

$$dec(enc(z_1, z_2), z_2) >_{r_1} z_1 \qquad fst([z_1, z_2]) >_{r_2} z_1 \qquad snd([z_1, z_2]) >_{r_3} z_2$$

and the example equality $fst(dec(enc([a, b], k), k)) =_{\mathcal{E}S} a$ then holds.

Example 2.2 To model public key cryptography we simply add public/private key generator functions to the signature from Example 2.1, i.e. $\Sigma_{pub} \triangleq \Sigma \cup \{\cdot^+, \cdot^-\}$. The equational theory $\mathcal{E}P$ is generated from the following rules:

$$dec(enc(z_1, {z_2}^+), z_2^-) >_{r_1} z_1 \qquad fst([z_1, z_2]) >_{r_2} z_1 \qquad snd([z_1, z_2]) >_{r_3} z_2$$

and the example equality $fst(dec(enc([a, b], k^+), k^-)) =_{\mathcal{E}P} a$ holds.

2.3 Frames and Static Equivalence

As explained in the introduction, environment knowledge is implicitly represented by frames of the form $\varphi = (\nu\tilde{n})\sigma$ where $\tilde{n}$ is a (possibly empty) list of private names and σ is a substitution of the form $\{{}^{M_1}/_{x_1}, \ldots, {}^{M_k}/_{x_k}\}$. That is, a frame is simply a substitution with possible restrictions on names. Considered as such, it can be applied to terms in the expected way, and we write $M\varphi$ for the term where each variable x_i occurring in M is replaced with M_i. We assume that substitutions are cycle-free and that all occurring terms are on normal form (i.e. irreducible in the associated term rewrite system). We denote by $dom(\varphi)$ and $im(\varphi)$ the domain and image of the substitution in φ, respectively. Free names and bound names of a frame are denoted by $fn(\varphi)$ and $bn(\varphi)$ respectively and are defined as expected.

Static equivalence expresses indistinguishability of frames based on equality between terms in frames:

Definition 2.3 Two terms M_1 and M_2 are *equal in frame* φ, written $(M_1 =_{\mathcal{E}} M_2)\varphi$ iff $n(M_1, M_2) \cap bn(\varphi) = \emptyset$ and $M_1\varphi =_{\mathcal{E}} M_2\varphi$.

Definition 2.4 Two frames φ and φ' are *statically equivalent* in $\mathcal{E}$, written $\varphi \approx_{\mathcal{E}} \varphi'$, iff $dom(\varphi) = dom(\varphi')$ and

$$(M_1 =_{\mathcal{E}} M_2)\varphi \Leftrightarrow (M_1 =_{\mathcal{E}} M_2)\varphi' \text{ for all terms } M_1 \text{ and } M_2$$

2.4 Reduction Observable Theories

In order to give a refined definition of static equivalence which coincides with the standard definition, and subsequently give a logical characterisation, we need to impose some assumptions on the theories under consideration.

The first assumption is that of reduction observable theories, defined as follows:

Definition 2.5 An equational theory $\mathcal{E}$ on $T(\Sigma, \mathcal{U})$ generated from a rewrite system $\mathcal{R}$ is *reduction observable* if

(i) There is a constant function symbol ok $\in \Sigma$

(ii) For every $d \in \Sigma$ there is a test function symbol $test\_d \in \Sigma$.

(iii) For every $(d(C[\tilde{z}]) >_r R) \in \mathcal{R}$ there is a reduction test rule $(test\_d(C[\tilde{z}], \mathrm{ok}) >_{r'} \mathrm{ok}) \in \mathcal{R}$.

Note that any theory $\mathcal{E}$ can be extended with appropriate test functions and rewrite rules to become reduction observable, and we shall generally denote this extension by $\mathcal{E}+$. A reduction observable theory essentially allows frames to be distinguished based on reductions in addition to equality. We argue that this is often a reasonable assumption. Consider the following two frames:

$$\varphi_2 \triangleq (\nu k, b, c)\{{}^{enc(b,k^+)}/_{x_1}, {}^{k^-}/_{x_2}\} \qquad \varphi_2' \triangleq (\nu k, b, c)\{{}^{enc(b,k^+)}/_{x_1}, {}^{c^-}/_{x_2}\}$$

The first frame contains an encrypted name and the corresponding private key which can be used for decryption. The second frame contains the same encrypted name but does not contain the corresponding private key for decryption. Now it turns out that $\varphi_2 \approx_{\mathcal{E}P} \varphi_2'$. The only relevant attempts at constructing distinguishing equalities are $dec(x_1, x_2) =_{\mathcal{E}P} b$ and $x_1 =_{\mathcal{E}P} enc(dec(x_1, x_2), k^+)$, but these do not hold in either frame since b and k are in $bn(\varphi_2)$ and $bn(\varphi_2')$.

The fact that $\varphi_2 \approx_{\mathcal{E}P} \varphi_2'$ might be slightly surprising. For we then get that the knowledge arising from a process which outputs an encrypted term together with the corresponding decryption key is semantically equivalent to a process which outputs the same encrypted term together with an unrelated decryption key! On the other hand, $\varphi_2 \not\approx_{\mathcal{E}P+} \varphi_2'$ because the equality $test\_dec(x_1, x_2) > ok$ holds in φ_2 but not in φ_2'.

Is this sensible? Implementations of cryptographic functions (e.g. the OpenSSL library [12]) typically do not provide means of testing whether decryption with a given decryption key is successful or not. However, in many applications it is assumed that this information is available, e.g. by appending a publicly known token to the clear text before encryption; it can then be checked if the decryption output also contains this token.

The second assumption is that of convergent subterm theories, i.e. theories generated from a convergent rewrite system $\mathcal{R} = \{L_i >_{r_i} R_i\}_{i \in I}$ in which R_i is a subterm of L_i for each $i \in I$. Static equivalence has been shown to be decidable for this class of theories [1].

For the third and final assumption we introduce the notion of a destructor context:

Definition 2.6 A context $D[\underline{x}, \tilde{x}]$ is a *destructor context* with target x (identified by underline) if it is unifiable with the left hand side of some rewrite rule $L >_r R$

and the position of x in $D[\underline{x}, \tilde{x}]$ is a proper prefix of the position of R in L.

For example, $snd(\underline{x})$ is a destructor context while $snd([\underline{x}, y])$ is not; this reflects the intuition that the destructor function snd "takes apart" x in the former context but not in the latter. We can now state the third and final assumption, namely that rewrite rules are *independent* in the sense that they do not contain destructor contexts as proper subterms; i.e. we disregard rewrite rules such as

$$dec(enc(fst([z_1, z_2]), z_3), z_3) > z_1$$

which intuitively says that only encrypted first components of pairs can be decrypted. This assumption is not very strong though, for the vast majority of theories with rules on the above form are not convergent. Note that if $\mathcal{E}$ is an independent convergent subterm theory then $\mathcal{E}+$ is also an independent convergent subterm theory.

3 Refining Static Equivalence

In this section we develop a refined definition of static equivalence which does not suffer from universal quantification over arbitrary terms. We start by defining *the analysis* of a frame, which intuitively is the set of terms resulting from taking the frame to bits, e.g. by iteratively decrypting or projecting terms in the frame. The analysis may generally be big though; therefore we also define the *cores* of a frame as the smallest subset of the analysis sufficient for "reproducing" the analysis. The refined definition of static equivalence places conditions on contexts over cores: corresponding terms in two frames must be *equal up to cores* and the same reductions and syntactic equalities over cores must hold in the two frames. In order to limit the number of contexts considered, we furthermore define the notion of *partitioning contexts*.

3.1 Analysis

For any equational theory we assume an associated *revelation relation*, $\succ_S$, where $M_1 \succ_S M_2$ if M_1 can reveal a subterm M_2 based on the set of terms S.

Definition 3.1 Let S be a set of terms. Define $M \succ_S M|_w$ if and only if there is a destructor context $D[\underline{x}, \tilde{x}]$ and terms $\tilde{T} \subseteq S$ such that $D[M, \tilde{T}] >_r M|_w$.

For example, we have that $enc(a, k^+) \succ_S a$ if $k \in S$ or $k^- \in S$. The *analysis* $\mathcal{A}(M, S)$ is then the iterated revelation from M based on the terms in S; this is a generalisation of the analysis for cryptographic protocols given in [10]. It will be important that analysis terms can be ordered by position in their parents, and therefore we define the analysis $\mathcal{A}(M, S)$ as a set of pairs $(M|_w, w)$ where $M|_w$ is a term revealed from M.

Definition 3.2 Let S be a set of terms and let $M \in S$. Then the *analysis* of M

with respect to S is defined inductively as follows:

$$\begin{aligned}\mathcal{A}^0(M,S) &\triangleq \{(M,\epsilon)\}\\ \mathcal{A}^{i+1}(M,S) &\triangleq \mathcal{A}^i(M,S)\cup\\ &\quad \{(M|_w,w)|\exists(M|_{w'},w')\in\mathcal{A}^i(M,S) \text{ s.t. } M|_{w'}\succ_{\bigcup_{T\in S}\mathcal{A}^i(T,S)} M|_w\}\end{aligned}$$

We further define $\mathcal{A}(M,\varphi) \triangleq \mathcal{A}(M, im(\varphi)\cup fn(\varphi))$ for frames φ and $\mathcal{A}(S) \triangleq \bigcup_{M\in S}\mathcal{A}(M,S)$.

Whenever positions are unimportant they will be omitted in the following, in which case the analysis is simply considered a multiset of terms.

3.2 Cores and Ecores

Considered as a set of known terms, the analysis often contains redundancy. For complexity reasons we only wish to consider the smallest subset of the analysis from which the analysis can be reconstructed by applying appropriate contexts. The first step towards this aim is the following definition of cores.

Definition 3.3 Let M be a term and S a set of terms. Then

$$cores(M,S) \triangleq \{(M|_w,w) \mid (M|_w,w)\in\mathcal{A}(M,S)\wedge M|_w\not\succ_{\mathcal{A}(S)}\}$$

Define $cores(M,\varphi) \triangleq cores(M, im(\varphi)\cup fn(\varphi))$, $cores(x,\varphi) \triangleq cores(x\varphi,\varphi)$ and, for $\varphi=(\nu\tilde{n})\{{}^{M_i}/_{x_i}\}_{i\in I}$,

$$cores(\varphi) \triangleq \bigcup_{i\in I}\{(M_i|_w,w,i) \mid (M_i|_w,w)\in cores(M_i,\varphi)\}\cup\{(n,\_,\_)\mid n\in fn(\varphi)\}$$

Note that we have defined the cores of a frame to include all free names in the frame (the place holder _ simply indicates that free names have no index and position) since these are an essential part of environment knowledge. As with the analysis we often take the liberty of ignoring the position information and consider cores as multisets of terms instead of pairs or triples.

Our definition of cores is a generalisation of the definition given in [6] where a core is a single term and defined only for symmetric key encryption. In contrast, our definition works for any convergent subterm rewrite system, resulting in cores which are sets.

The above definition of cores is however sometimes insufficient for capturing the idea that the analysis can be reconstructed by applying appropriate contexts to cores. Take for instance the following simple frame:

$$\varphi \triangleq (\nu k)\{{}^{enc([a,b],k^+)}/_{x_1},\ {}^{k^-}/_{x_2}\}$$

Then $cores(x_1,\varphi)=\{a,b\}$, but k^+ is not a core. Disregarding k^+ altogether results in a loss of information: $M_1=enc([a,b],k^+)$ (and thereby the analysis of φ) cannot

be reconstructed by applying contexts to the cores, and the fact that the equality $dec(M_1, k^-) =_{\mathcal{E}} [a, b]$ holds is lost. This prompts us to define *extended cores* thus:

Definition 3.4 A term $M|_w \in \mathcal{A}(M, S)$ is an *extended core*, or *ecore*, with respect to the set S if either

(i) $(M|_w, w) \in cores(M, S)$.

(ii) There is no context $C[\tilde{x}]$ and no terms $\tilde{M} \subseteq \mathcal{A}(S)$ such that $M|_w = C[\tilde{M}]$.

The set of extended cores of M with respect to S is denoted by $ecores(M, S)$, and we define $ecores(M, \varphi)$, $ecores(x, \varphi)$ and $ecores(\varphi)$ as for cores.

Again we often disregard the position information in ecores.

Example 3.5 Take the frame $\varphi \triangleq (\nu a, b, k_1, k_2)\{{}^{[enc(a,{k_1}^+),enc(b,{k_2}^+)]}/_{x_1}, {k_1}^-/_{x_2}\}$ in the theory $\mathcal{E}P$. Then

$$cores(x_1, \varphi) = \{a, enc(b, {k_2}^+)\}$$
$$ecores(x_1, \varphi) = \{enc(a, {k_1}^+), enc(b, {k_2}^+), a\}$$

Note how each term in $im(\varphi)$ can now be written as a context over ecores, and that *the analysis can be reconstructed from ecores* by applying appropriate function symbols!

We impose a linear ordering on ecores for the purpose of comparing ecores with the same index in different frames and write $ecores(\varphi) = (N)_{i \in J}$ for an ordered sequence of ecores. The ordering is based on positions and indices in frames, but the details are insignificant and have been omitted; they can be found in [11].

3.3 Partitioning Contexts

In order to limit the complexity of the refined definition of static equivalence and subsequently give a construction of finite characteristic formula, it is crucial that we only consider a restricted class of *partitioning contexts*. To that end we first need a definition of a *correlation relation* $\rightleftharpoons$ on variables in a context. Intuitively, $y_1 \rightleftharpoons y_2$ in a context $C[\tilde{y}]$ if y_1 and y_2 depend on each other in a reduction of instances of $C[\tilde{y}]$; in the context $dec(enc(y_1, {y_2}^+), y_3)$ we thus have that y_2 and y_3 are correlated while y_1 is neither correlated to y_2 nor y_3. In other words, encryption keys are mutually dependent but they are independent of the clear text. Here is the general definition:

Definition 3.6 Let $L[\tilde{z}] > R[\tilde{z}]$ be a rewrite rule and let $C[\tilde{y}]$ be a context which is unifiable with $L[\tilde{z}]$. Let w_i and w_j be two positions in $C[\tilde{y}]$ and let w_i' and w_j' be the longest prefixes of w_i, respectively w_j, such that the position w_i', respectively w_j', exists in $L[\tilde{z}]$. We then say that w_i and w_j are *strongly correlated*, written $w_i \rightleftarrows w_j$, if $v(C[\tilde{z}]|_{w_i'}) \cap v(C[\tilde{z}]|_{w_j'}) \neq \emptyset$.

Strong correlation is a reflexive and symmetric binary relation. Define (weak) *correlation*, $\rightleftharpoons$, to be the transitive closure of strong correlation and say that w_i and w_j are correlated if $w_i \rightleftharpoons w_j$.

For each pair of variables $y_i, y_j \in v(C[\tilde{y}])$ let $\tilde{w}_i$ and $\tilde{w}_j$ be the positions of y_i and y_j in $C[\tilde{y}]$, respectively (there will be multiple positions if a variable has multiple occurrences). We then define $y_i \rightleftharpoons y_j$ if $w_i \rightleftharpoons w_j$ for every $w_i \in \tilde{w}_i$ and $w_j \in \tilde{w}_j$.

Definition 3.7 A context $C^{\perp}[\tilde{y}]$ over variables and the distinguished name $\perp$ is *partitioning* if it is unifiable with the LHS of some rewrite rule $L[\tilde{z}] > R[\tilde{z}]$ and the following hold:

(i) $y_i \rightleftharpoons y_j$ for all $y_i, y_j \in \tilde{y}$.

(ii) $C^{\perp}[\tilde{y}]|_w = \perp$ implies $L[\tilde{z}]|_w = z$ for all w and some $z \in \tilde{z}$.

The first condition says that all variables in a partitioning context must be correlated, and the second condition says that all occurrences of $\perp$ must unify trivially with a variable in the rewrite rule.

Example 3.8 The following are examples of contexts which can easily be verified to be partitioning with respect to the public-key rewrite rule:

- $dec(enc(\perp, {y_1}^+), y_2^-)$
- $dec(enc(y_1, \perp^+), \perp^-)$

The following are examples of contexts which are *not* partitioning:

- $dec(enc(y_1, y_2), \perp^-)$ is not partitioning because y_1 and y_2 are not correlated (condition 1 fails).
- $dec(enc(h(\perp), {y_1}^+), y_2)$ is not partitioning because the position of $\perp$ does not match the position of z_1 (condition 2 fails).

3.4 The Refined Definition of Static Equivalence

Say that M_1 is *more general* than M_2, written $M_1 \sqsupset M_2$, if there exists some substitution θ such that $M_2 = M_1\theta$. We shall in addition write $M_1 \square M_2$ to mean that M_1 and M_2 are unifiable but neither is more or less general than the other, i.e. there exists a unifying substitution θ such that $M_1\theta = M_2\theta$, $M_1 \not\sqsupset M_2$ and $M_2 \not\sqsupset M_1$.

We shall mainly be interested in asserting generality on contexts in relation to the LHS of some rewrite rule. For this purpose we also define $\sqsupset$ and $\square$ as unary relation symbols, and write e.g. $C[\tilde{y}]\square$ if there exists a rewrite rule $L[\tilde{z}] >_r R[\tilde{z}]$ in the relevant rewrite system such that $C[\tilde{y}]\square L[\tilde{z}]$ (and similarly for $\sqsupset$). With these notions we are now ready to state the refined definition of static equivalence.

Definition 3.9 Let $\varphi = (\nu\tilde{n})\{{}^{M_i}/_{x_i}\}_{i\in I}$ and $\varphi' = (\nu\tilde{n})\{{}^{M'_i}/_{x_i}\}_{i\in I}$ be two frames with $ecores(\varphi) = (N)_{j\in J}$, $ecores(\varphi') = (N')_{j\in J}$. Then φ and φ' are *refined statically equivalent*, written $\varphi \approx'_{\mathcal{E}} \varphi'$, if each of the following conditions holds:

(i) For each $i \in I$ there is some context $C[\tilde{y}]$ such that $M_i = C[\tilde{N}]$ and $M'_i = C[\tilde{N'}]$.

(ii) For any context $C[\tilde{y}]$ and for all $j \in J$ it holds that

$$C[\tilde{N}] = N_j \Leftrightarrow C[\tilde{N'}] = N'_j$$

(iii) For any partitioning context $C_1^{\perp}[\tilde{y}]$ where $C_1^{\perp}[\tilde{y}] \sqsupset$ or $C_1^{\perp}[\tilde{y}]\Box$ it holds for all $C_2^{\perp}[\tilde{y}]$ that

$$C_1^{\perp}[\tilde{N}] >_r C_2^{\perp}[\tilde{N}] \Leftrightarrow C_1^{\perp}[\tilde{N'}] >_r C_2^{\perp}[\tilde{N'}]$$

Contexts and ecores have been defined in such way that condition 1 is well-defined, and it is easy to see by induction that any term in a frame can indeed be written as a context over ecores. Conditions 2 and 3 contain universal quantifications over contexts, but the key point here is that there are only finitely many equalities and reductions which *do* hold since variables in partitioning contexts must be correlated and there are only finitely many ecores. An appropriate semantics for quantifiers in a first-order logic of frames allows us to express all the (infinitely many) equalities and reductions which do *not* hold, which is the key to deriving characteristic formulae.

3.5 Coincidence Results

The refined definition $\approx'_{\mathcal{E}}$ has been derived with the intention that it should coincide with $\approx_{\mathcal{E}}$, which indeed turns out to be the case. Because of space constraints we only state the main results and refer the reader to [11, Chapter 4] for the full proofs.

Theorem 3.10 $\approx'_{\mathcal{E}} \subseteq \approx_{\mathcal{E}}$.

The proof relies on the following lemma:

Lemma 3.11 *Let φ and φ' be two frames with $\varphi \approx'_{\mathcal{E}} \varphi'$, $ecores(\varphi) = (N)_{j\in J}$ and $ecores(\varphi') = (N')_{j\in J}$. It then holds for any contexts $C_1[\tilde{y}]$ and $C_2[\tilde{y}]$ that:*

(i) $C_1[\tilde{N}] = C_2[\tilde{N}] \Leftrightarrow C_1[\tilde{N'}] = C_2[\tilde{N'}]$

(ii) $C_1[\tilde{N}] >_r C_2[\tilde{N}] \Leftrightarrow C_1[\tilde{N'}] >_r C_2[\tilde{N'}]$

(iii) $C_1[\tilde{N}] =_{\mathcal{E}} C_2[\tilde{N}] \Leftrightarrow C_1[\tilde{N'}] =_{\mathcal{E}} C_2[\tilde{N'}]$

The first result is shown by contradiction. The second result relies on partitioning contexts: any context $C[\tilde{y}]$ which is unifiable with the LHS of a rewrite rule gives rise to a set of partitioning contexts $\{C_i^{\perp}[\tilde{y}]\}_{i\in I}$ (which we call *generated partitioning contexts*), each over an equivalence class of variables (under correlation). We then use that $C[\tilde{N}]$ is reducible iff $C_i^{\perp}[\tilde{N}]$ is reducible for all $i \in I$. The third result uses the first two results together with the fact that, for convergent rewrite systems, $M_1 =_{\mathcal{E}} M_2$ iff $M_1 >^* M_3$ and $M_2 >^* M_3$ for some and M_3.

Theorem 3.10 now follows fairly straight forwardly from Lemma 3.11, the definition of $\approx_{\mathcal{E}}$ and the fact that every term which does not contain bound names can be written as a context over ecores.

Theorem 3.12 $\approx_{\mathcal{E}} \subseteq \approx'_{\mathcal{E}}$.

The proof relies on the following lemma to establish a relationship between contexts over ecores and contexts over frame terms:

Lemma 3.13 *Let* $\varphi = (\nu\tilde{n})\{^{M_i}/_{x_i}\}_{i\in I}$, $\varphi' = (\nu\tilde{n'})\{^{M'_i}/_{x_i}\}_{i\in I}$ *with* $\varphi \approx_{\mathcal{E}} \varphi'$. *Then for each* $M_i|_w \in \mathcal{A}(M_i, \varphi)$ *and* $M'_i|_w \in \mathcal{A}(M'_i, \varphi')$ *there is a context* $\mathcal{R}[\tilde{x}]$, *called an* analysis recipe, *and a* $k \in \mathbb{N}$ *such that*

$$\mathcal{R}[\tilde{M}] >^k M_i|_w \not>$$
$$\mathcal{R}[\tilde{M'}] >^k M'_i|_w \not>$$

The lemma is shown by construction, i.e. an inductive definition of analysis recipes is given and shown to have the desired property. The argument relies on the assumption that theories are reduction observable, which immediately gives that a term can reduce in exactly the same number of steps in statically equivalent frames. Lemma 3.13 can then be used in the proof of Theorem 3.12 to show that $\approx_{\mathcal{E}}$ implies condition 3 of $\approx'_{\mathcal{E}}$: if we suppose that $\varphi \approx_{\mathcal{E}} \varphi'$ and $C_1^{\perp}[\tilde{N}] >_r C_2^{\perp}[\tilde{N}]$, then $C_1^{\perp}[\mathcal{R}[\tilde{M}]] >^l C_1^{\perp}[\tilde{N}] >_r C_2^{\perp}[\tilde{N}]$ for some l, and also $C_1^{\perp}[\mathcal{R}[\tilde{M'}]] >^l C_1^{\perp}[\tilde{N'}]$. Reduction observability then gives that $C_1^{\perp}[\tilde{N'}]$ must be reducible. The assumption that $C_1^{\perp}[\tilde{y}]$ is partitioning, that $C_1^{\perp}[\tilde{y}] \sqsupset$ or $C_1^{\perp}[\tilde{y}]\Box$, and that rewrite systems are independent, then gives us that this last reduction must be primitive, i.e. $C_1^{\perp}[\tilde{N'}] >_r C_2^{\perp}[\tilde{N'}]$ as desired. Condition 2 is shown using similar ideas.

The last result tells us when the assumption of reduction observable theories is redundant:

Theorem 3.14 *If* $ecores(\varphi) = cores(\varphi)$ *then* $\varphi \approx_{\mathcal{E}} \varphi' \Leftrightarrow \varphi \approx_{\mathcal{E}+} \varphi'$.

One direction of the proof is of course immediate. For the other direction we use that when $ecores(\varphi) = cores(\varphi)$, any analysis term which is not a core can be written as a non-trivial context (i.e. a context which is not a variable) over ecores. This, together with confluence, can be used to "force" reductions in a statically equivalent frame using only equality. To see how this works, consider the *symmetric-key* counterparts of φ_2 and φ'_2 from Subsection 2.4:

$$\varphi_3 \triangleq (\nu k, b, c)\{^{enc(b,k)}/_{x_1}, {}^{k}/_{x_2}\} \qquad \varphi'_3 \triangleq (\nu k, b, c)\{^{enc(b,k)}/_{x_1}, {}^{c}/_{x_2}\}$$

We again have that $\varphi_3 \not\approx_{\mathcal{ES}+} \varphi'_3$ because $test\_dec(x_1, x_2) >$ ok holds in φ_3 but not in φ'_3. But suddenly we also have that $\varphi_3 \not\approx_{\mathcal{ES}} \varphi'_3$ because the following equality can be used to express that the above reduction holds in φ_3 but not in φ'_3:

$$x_1 =_{\mathcal{E}} enc(dec(x_1, x_2), x_2)$$

This equality is possible exactly because $enc(b, k)$ can be written as a non-trivial context over the ecores b and k, which in general means that ecores and cores coincide. In contrast, this is not possible for the term $enc(b, k^+)$ in the public-key version because k^+ is not an ecore.

4 A Logic of Frames

In this section we introduce the first-order logic for frames, $\mathcal{LF}$, which characterizes static equivalence and yields characteristic formulae.

4.1 Syntax

The syntax for $\mathcal{LF}$ is defined as follows, where the M_i range over terms and x ranges over variables:

$$A ::= M_1 =_{\mathcal{E}} M_2 \mid M_1 > M_2 \mid M_1 = M_2 \mid A_1 \vee A_2 \mid \neg A_1 \mid (A_1) \mid \exists x(A_1)$$

The full set of logical connectives is defined from $\neg$,$\vee$ and $\exists$ in the usual way.

4.2 Semantics

Common for all three atomic propositions is the requirement that terms being tested do not contain private names since formulae should not be able to distinguish frames based on private names. Hence the satisfaction relation $\vDash$ for the propositional logic is defined as follows:

$\varphi \vDash M_1 =_{\mathcal{E}} M_2$	if $n(M_1, M_2) \cap bn(\varphi) = \emptyset$ and $M_1\varphi =_{\mathcal{E}} M_2\varphi$
$\varphi \vDash M_1 > M_2$	if $n(M_1, M_2) \cap bn(\varphi) = \emptyset$ and $M_1\varphi > M_2\varphi$
$\varphi \vDash M_1 = M_2$	if $n(M_1, M_2) \cap bn(\varphi) = \emptyset$ and $M_1\varphi = M_2\varphi$
$\varphi \vDash A_1 \vee A_2$	if $\varphi \vDash A_1$ or $\varphi \vDash A_2$
$\varphi \vDash \neg A$	if $\varphi \nvDash A$

Quantification should allow reasoning about the knowledge represented by a frame, which we define formally as the *synthesis* $\mathcal{S}(\varphi)$ (a generalisation of the corresponding notion of synthesis introduced by Paulson in [10]):

Definition 4.1

$$\mathcal{S}(\varphi) \triangleq \{C[\tilde{N}] \mid \tilde{N} \subseteq ecore(\varphi) \text{ and } C[\tilde{y}] \text{ is a context } \}$$

The definition of satisfaction can now be completed with the case for existentially quantified formulae:

$$\varphi \vDash \exists x(A_1) \quad \text{if} \quad \varphi\{{}^{M}/_{x}\} \vDash A_1 \text{ for some term } M \in \mathcal{S}(\varphi)$$

Observe how bindings of quantified variables are represented in a natural way by extending the frame with an additional substitution. In cases where $x \in dom(\varphi)$ we assume alpha-conversion of x to some $x' \notin dom(\varphi)$, since otherwise a quantification may overwrite existing terms in φ.

Example 4.2 Consider the frames

$$\varphi_4 \triangleq (\nu k, b)\{{}^{enc(b,k^+)}/_{x_1}, {}^{[c,k]}/_{x_2}\} \qquad \varphi_4' \triangleq (\nu k, b)\{{}^{enc(enc(b,k^+),k^+)}/_{x_1}, {}^{[c,k]}/_{x_2}\}$$

The formula $\exists y_1 \exists y_2 (dec(x_1, y_1) > y_2 \wedge \neg \exists y_3 \exists y_4 (dec(y_2, y_3) > y_4))$ then expresses that x_1 can be decrypted using some known key and that the resulting term cannot be further decrypted. This property holds for φ_4 but not for φ_4'.

4.3 Results

The first major result says that $\mathcal{LF}$ characterizes static equivalence.

Theorem 4.3 ($\mathcal{LF}$ characterizes $\approx_{\mathcal{E}}'$) *Let* $Th_{\mathcal{LF}}(\varphi) \stackrel{\Delta}{=} \{A \in \mathcal{LF} \mid \varphi \vDash A\}$. *It then holds that* $\varphi \approx_{\mathcal{E}}' \varphi' \Leftrightarrow Th_{\mathcal{LF}}(\varphi) = Th_{\mathcal{LF}}(\varphi')$.

The proof is by induction on the structure of formulae, and the induction case for existential quantification relies on the following lemma:

Lemma 4.4 (Extension Lemma) *Let* $\varphi = (\nu \tilde{n})\{^{M_i}/_{x_i}\}_{i \in I}$ *and* $\varphi' = (\nu \tilde{n})\{^{M_i'}/_{x_i}\}_{i \in I}$ *be two frames with* $\varphi \approx_{\mathcal{E}}' \varphi'$, *and let* $ecores(\varphi) = (N)_{j \in J}$ *and* $ecores(\varphi') = (N')_{j \in J}$. *Then for any* $C[\tilde{y}]$ *it holds that* $\varphi\{^{C[\tilde{N}]}/_{x_s}\} \approx_{\mathcal{E}}' \varphi'\{^{C[\tilde{N'}]}/_{x_s}\}$ *(where* s *is any index with* $s \notin I$*).*

The second major result asserts the existence of characteristic formulae.

Theorem 4.5 (Characteristic formulae) *For any frame* φ *there is a finite* characteristic formula $\mathcal{C}_\varphi$ *such that for any frame* φ' *with* $dom(\varphi) = dom(\varphi')$ *it holds that* $\varphi \approx_{\mathcal{E}}' \varphi' \Leftrightarrow \varphi' \vDash \mathcal{C}_\varphi$.

The remaining of this subsection is devoted to a construction of the characteristic formula $\mathcal{C}_\varphi$ for a frame $\varphi = (\nu \tilde{n})\{^{M_i}/_{x_i}\}_{i \in I}$. Again we refer the reader to [11] for the proof that this construction does indeed work. $\mathcal{C}_\varphi$ is of the following form:

$$\begin{aligned} \mathcal{C}_\varphi \stackrel{\Delta}{=} \exists y_1 \ldots \exists y_k (& \\ & \psi_{ecore-1}(y_1) \wedge \cdots \wedge \psi_{ecore-k}(y_k) \wedge \\ & \psi_{cond-1} \wedge \\ & \psi_{cond-2a} \wedge \psi_{cond-2b} \wedge \psi_{cond-2c} \\ & \psi_{cond-3a} \wedge \psi_{cond-3b} \wedge \psi_{cond-3c}) \end{aligned}$$

Each ecore is represented by one of the existentially quantified variables y_i and the formulae $\psi_{ecore-i}(y_i)$ state that y_i is in fact an ecore. Conditions $1-3$ of $\approx_{\mathcal{E}}'$ are then encoded in the remaining conjuncts which we elaborate below.

First note that there are infinitely many contexts over ecores in condition 3 of $\approx_{\mathcal{E}}'$ for which reductions do *not* hold. For example, to express that the ecores bound to y_1 and y_2 are not related public/private key pairs, a characteristic formula should assert that

$$dec(enc(\bot, y_1), y_2) \not\succ, dec(enc(\bot, {y_1}^+), y_2) \not\succ, dec(enc(\bot, h(y_1)^+), y_2) \not\succ, \ldots$$

which would give rise to infinite conjunction. Instead we just choose the first of the

contexts and use existential quantification to express that

$$\neg\exists z^*.dec(enc(\bot, z^*), y_2) >$$

in cases where y_2 is *not* the private-key counterpart of *any* other synthesis term (this works because existential quantification ranges over the synthesis of a frame and any synthesis term can be written as a context over ecores). In cases where y_2 *is* the private-key counterpart of some other synthesis term distinct from y_1, we automatically get that y_2 is not also the private-key counterpart of y_1. The generalisation of this argument relies on the context in question being partitioning and hence $z^* \rightleftharpoons y_2$. The above context of choice is what we refer to as a *minimised context*:

Definition 4.6 A *minimised context* $C[\tilde{y}, \tilde{z^*}]$ is any partitioning context with $C[\tilde{y}, \tilde{z^*}] \sqsupset$. Each $y \in \tilde{y}$ is intended to represent an ecore while each $z^* \in \tilde{z^*}$ is intended to represent an arbitrary synthesis term.

Note that there are only finitely many minimised contexts which is crucial for the following construction of finite characteristic formulae.

Encoding Ecores

First we encode revelation thus:

$$M \succ T \triangleq \bigvee_{\substack{D[\underline{y}, z_1^*, \ldots, z_s^*] \\ \text{is a minimised destructor context}}} \exists z_1^* \ldots \exists z_s^* (D[M, z_1^*, \ldots, z_s^*] > T)$$

Let $\mathcal{R}_j[\tilde{x}]$ be the analysis recipe for each N_j and let $k_j \in \mathbb{N}$ be such that $\mathcal{R}_j[\tilde{M}] >^{k_j} N_j$. The ecore predicate is then encoded as follows:

$$\begin{aligned}\psi_{ecore-j}(y_j) \triangleq\ & \exists z_1 \ldots \exists z_{k_j} (\mathcal{R}_j[\tilde{x}] > z_1 \wedge z_2 > z_3 \wedge \cdots \wedge z_{k_j - 1} > y_j) \wedge \\ & [\neg\exists z_{k_j} (y_j \succ z_{k_j}) \vee \\ & \bigwedge_{f(z_1,\ldots,z_s) \in \Sigma^{(s)}} \forall z_1, \ldots, z_s \neg(y_j = f(z_1, \ldots, z_s))]\end{aligned}$$

Encoding Condition 1

Each M_i can be written as a context over ecores – let $C_i[\tilde{y}]$ be any such context. Condition 1 in $\approx'_{\mathcal{E}}$ is then expressed in the following formulae:

$$\psi_{cond-1} \triangleq \bigwedge_{i \in I} x_i = C_i[\tilde{y}]$$

Encoding Condition 2

The encodings rely on the following sets:

$$\begin{aligned}
\mathcal{S}_{2a} &\triangleq \{(i,j) \mid N_i = N_j\} \quad \bar{\mathcal{S}}_{2a} \triangleq \{(i,j) \mid N_i \neq N_j\} \\
\mathcal{S}_{2b} &\triangleq \{(C[\tilde{y}], j) \mid C[\tilde{N}] = N_j\} \\
\mathcal{S}_{2c} &\triangleq \{j \mid \text{ there is no context } C[\tilde{y}] \text{ such that } C[\tilde{N}] = N_j\}
\end{aligned}$$

$$\begin{aligned}
\psi_{cond-2a} &\triangleq \bigwedge_{(i,j)\in\mathcal{S}_{2a}} y_i = y_j \wedge \bigwedge_{(i,j)\in\bar{\mathcal{S}}_{2a}} y_i \neq y_j \\
\psi_{cond-2b} &\triangleq \bigwedge_{(C[\tilde{y}],j)\in\mathcal{S}_{2b}} C[\tilde{y}] = y_j \\
\psi_{cond-2c} &\triangleq \bigwedge_{j\in\mathcal{S}_{2c}} \bigwedge_{f(z_1,\ldots,z_k)\in\Sigma^k} \forall z_1 \ldots \forall z_k (\neg f(z_1,\ldots,z_k) = y_j)
\end{aligned}$$

Encoding Condition 3

The encodings rely on the following sets:

$$\begin{aligned}
\mathcal{S}_{3a} &\triangleq \{(C_1^{\perp}[\tilde{y}], C_2^{\perp}[\tilde{y}]) \mid C_1[\tilde{y}] \text{ is partitioning } \wedge C_1^{\perp}[\tilde{N}] \sqsupset \wedge C_1^{\perp}[\tilde{N}] > C_2^{\perp}[\tilde{N}]\} \\
\bar{\mathcal{S}}_{3a} &\triangleq \{(C_1^{\perp}[\tilde{y}], C_2^{\perp}[\tilde{y}]) \mid C_1[\tilde{y}] \text{ is partitioning } \wedge C_1^{\perp}[\tilde{N}] \sqsupset \wedge C_1^{\perp}[\tilde{N}] \not> C_2^{\perp}[\tilde{N}]\} \\
\mathcal{S}_{3b} &\triangleq \{(C_1^{\perp}[\tilde{y}], C_2^{\perp}[\tilde{y}]) \mid C_1[\tilde{y}] \text{ is partitioning } \wedge C_1^{\perp}[\tilde{N}] \square \wedge C_1^{\perp}[\tilde{N}] > C_2^{\perp}[\tilde{N}]\} \\
\mathcal{S}_{3c} &\triangleq \{C_1^{\perp}[\tilde{y}, z_1, \ldots, z_s] \mid C_1^{\perp}[\tilde{y}, z_1^*, \ldots, z_s^*] \text{ is a minimised context and} \\
&\qquad \neg\exists T_1, \ldots, T_s \in \mathcal{S}(\varphi). C_1[\tilde{N}, T_s, \ldots, T_s] >\}
\end{aligned}$$

$$\begin{aligned}
\psi_{cond-3a} &\triangleq \bigwedge_{(C_1^{\perp}[\tilde{y}],C_2^{\perp}[\tilde{y}])\in\mathcal{S}_{3a}} C_1^{\perp}[\tilde{y}] > C_2^{\perp}[\tilde{y}] \wedge \bigwedge_{(C_1^{\perp}[\tilde{y}],C_2^{\perp}[\tilde{y}])\in\bar{\mathcal{S}}_{3a}} \neg C_1^{\perp}[\tilde{y}] > C_2^{\perp}[\tilde{y}] \\
\psi_{cond-3b} &\triangleq \bigwedge_{(C_1^{\perp}[\tilde{y}],C_2^{\perp}[\tilde{y}])\in\mathcal{S}_{3b}} C_1^{\perp}[\tilde{y}] > C_2^{\perp}[\tilde{y}] \\
\psi_{cond-3c} &\triangleq \bigwedge_{C_1^{\perp}[\tilde{y},z_1^*,\ldots,z_s^*]\in\mathcal{S}_{3c}} \forall z_1^* \ldots \forall z_s^* \forall z (\neg C_1^{\perp}[\tilde{y}, z_1^*, \ldots, z_s^*] > z)
\end{aligned}$$

5 A Logic for Applied π

In this section we show how the frames logic can be extended to a logic for Applied π by adding suitable modalities. We start by briefly recapping Applied π and refer readers unfamiliar with the calculus to [3] for detailed information.

5.1 The Applied π Calculus

The syntax for Applied π is divided into two categories: plain and extended processes. Plain processes are similar to those of the π calculus except that arbitrary terms may be used in conditionals and in output (recall that u ranges over names and variables):

$$P^p, Q^p ::= \mathbf{0} \mid P^p|Q^p \mid !P^p \mid (\nu n)P^p \mid \mathit{if}\ M_1 = M_2\ \mathit{then}\ P^p\ \mathit{else}\ Q^p \mid u(x).P^p \mid \overline{u}\langle M\rangle.P^p$$

Extended processes add the notion of *active substitutions* which intuitively capture the knowledge of an environment in the process syntax itself:

$$P, Q, R ::= P^p \mid P|Q \mid (\nu n)P \mid (\nu v)P \mid \{^M/_x\}$$

This notion of environment-sensitivity dates back to the semantics for the π-calculus given in [5] and the later semantics for a spi-calculus given in [6]. Here, though, environments were not considered part of the process syntax.

For the logic that follows, we shall refer to the labelled transition semantics of Applied π. We here give an example in place of the formal definition (for this, see [3]).

Example 5.1 The following example in the symmetric key theory $\mathcal{ES}$ shows a transition sequence for a process P which outputs a secret name s encrypted by k, then "accidentally" outputs k, then inputs some term bound to y and checks if it equals s.

$$\begin{aligned}
P \triangleq\ & (\nu s, k)\overline{a}\langle enc(s,k)\rangle.\overline{a}\langle k\rangle.a(y).\text{if } y = s \text{ then } Q \text{ else } R \\
\xrightarrow{(\nu x_1)\overline{a}\langle x_1\rangle}\ & (\nu s, k)(\{^{enc(s,k)}/_{x_1}\}|\overline{a}\langle k\rangle.a(y).\text{if } y = s \text{ then } Q \text{ else } R \\
\xrightarrow{(\nu x_2)\overline{a}\langle x_2\rangle}\ & (\nu s, k)(\{^{enc(s,k)}/_{x_1}\}|\{^{k}/_{x_2}\}|a(y).\text{if } y = s \text{ then } Q \text{ else } R \\
\xrightarrow{a(dec(x_1,x_2))}\ & (\nu c, k)(\{^{enc(c,k)}/_{x_1}\}|\{^{k}/_{x_2}\}|\text{if } dec(x_1,x_2) = s \text{ then } Q \text{ else } R) \\
\xrightarrow{\tau}\ & (\nu c, k)(\{^{enc(c,k)}/_{x_1}\}|\{^{k}/_{x_2}\}|Q)
\end{aligned}$$

The two outputs give rise to bound output transitions and *new active substitutions* representing the terms which have been output. An arbitrary term built from variables in the active substitutions is then sent as a labelled input by the environment; the active substitutions are applied to free variables in the conditional process, and an internal reduction is carried out by testing for equality in the equational theory $\mathcal{ES}$.

The *frame* $\varphi(P)$ of a process P is obtained by "merging" all active substitutions in P while preserving restrictions, normalising terms and renaming if necessary. In the above example we have that $\varphi(P) = (\nu c, k)\{^{enc(c,k)}/_{x_1}, {}^{k}/_{x_2}\}$.

5.2 Syntax for The Process Logic

A logic $\mathcal{LA}$ for Applied π is obtained by adding modalities to the frames logic thus:

$$\begin{aligned} A ::= & M_1 =_{\mathcal{E}} M_2 \mid M_1 > M_2 \mid M_1 = M_2 \mid \exists x(A) \mid \neg A \mid A_1 \vee A_2 \\ & \mid \langle \tau \rangle A \mid \langle \bar{a}u \rangle A \mid \langle \nu \bar{a}u \rangle A \mid \langle a(x) \rangle A \end{aligned}$$

Informally, the modalities express respectively *possibility of internal action*, *possibility of output of u on a*, *possibility of output of bound u on a* and *possibility of input of x on a*. These correspond to each the four possible labels featuring in the labelled semantics for Applied π. The dual modalities (i.e. necessity) can be defined in the usual way.

5.3 Semantics for The Process Logic

Let $\vDash_{\mathcal{LF}}$ be the satisfaction relation for the frames logic $\mathcal{LF}$ defined in Section 4. The satisfaction relation for the process logic $\mathcal{LA}$ can then be defined as follows (where $\xrightarrow{\tau}{}^*$ is the reflexive and transitive closure of $\xrightarrow{\tau}$):

$P \vDash M_1 =_{\mathcal{E}} M_2$	if $\varphi(P) \vDash_{\mathcal{LF}} M_1 =_{\mathcal{E}} M_2$
$P \vDash M_1 = M_2$	if $\varphi(P) \vDash_{\mathcal{LF}} M_1 = M_2$
$P \vDash M_1 > M_2$	if $\varphi(P) \vDash_{\mathcal{LF}} M_1 > M_2$
$P \vDash \exists x(A)$	if there exists $M \in \mathcal{S}(\varphi(P))$ s.t. $(P \mid \{^M/_x\}) \vDash A$
$P \vDash \neg A$	if $P \nvDash A$
$P \vDash A_1 \vee A_2$	if $P \vDash A_1$ or $P \vDash A_2$
$P \vDash \langle \tau \rangle A$	if there exists P' s.t. $P \xrightarrow{\tau}{}^* P'$ and $P' \vDash A$
$P \vDash \langle \bar{a}u \rangle A$	if there exists P' s.t. $P \xrightarrow{\tau}{}^* \xrightarrow{\bar{a}\langle u \rangle} \xrightarrow{\tau}{}^* P'$ and $P' \vDash A$
$P \vDash \langle \nu \bar{a}u \rangle A$	if there exists P' s.t. $P \xrightarrow{\tau}{}^* \xrightarrow{(\nu u)\bar{a}\langle u \rangle} \xrightarrow{\tau}{}^* P'$ and $P' \vDash A$
$P \vDash \langle a(x) \rangle A$	if there exists M, P' s.t. $P \xrightarrow{\tau}{}^* \xrightarrow{a(M)} \xrightarrow{\tau}{}^* P'$ and $P' \vDash A\{^M/_x\}$

Example 5.2 The process logic can be used to reason about knowledge over time. Consider the formula $A_1 = \langle \nu \bar{a} x_1 \rangle \langle \nu \bar{a} x_2 \rangle \exists y_1 \exists y_2 (dec(x_1, y_1) > y_2)$ in the symmetric key theory $\mathcal{E}S$. A_1 asserts that two messages bound to x_1 and x_2 can be output, after which the decryption key for the first message is known by the environment. This formula is satisfied by process P from example 5.1 above. Note that the order of modalities and quantifiers is significant. For example, P does *not* satisfy the formula $A_2 = \exists y_1 \langle \nu \bar{a} x_1 \rangle \langle \nu \bar{a} x_2 \rangle \exists y_2 (dec(x_1, y_1) > y_2)$ because the decryption key is not known by the environment until revealed by the second output in P. In this sense the modal logic can be used to reason about knowledge over time.

The process logic characterizes labelled bisimilarity under the assumption of image-finite processes; we refer the reader to [11] for further details.

6 Conclusion and Future Work

We have introduced a logic for frames which characterizes static equivalence and yields characteristic formulae under the assumptions that the theory under consideration is 1) reduction observable, 2) a convergent subterm theory and 3) independent. In addition we have shown when assumption 1 is unnecessary; this is e.g. the case for symmetric key theories, but not for public key theories. The characterisation results rely on a refined version of static equivalence defined without recourse to quantification over arbitrary terms. Finally, we briefly indicated how the logic for frames extends to a modal logic for Applied π. The resulting logic has been used to describe a well known attack on a security protocol in [11].

An interesting future direction would be to investigate if the restriction to convergent subterm theories can be lifted. A decision procedure for satisfaction in the frame logic would be of practical use in tools; the fact that synthesis membership is decidable for convergent subterm theories [1] raises hope that such a decision procedure does indeed exist. Finally, it may be worthwhile to investigate the complexity of deciding static equivalence using the refined definition in order to match or improve on the polynomial time bound given in [1].

Acknowledgements

The authors would like to thank the anynomous reviewers for their useful comments.

References

[1] Martín Abadi and Véronique Cortier. *Deciding knowledge in security protocols under equational theories.* Proc. 31st Int. Coll. Automata, Languages, and Programming (ICALP'2004), Lecture Notes in Computer Science **3142** (2004), 46–58.

[2] Martín Abadi and Véronique Cortier. *Deciding knowledge in security protocols under (many more) equational theories.* CSFW '05: Proceedings of the 18th IEEE Computer Security Foundations Workshop (CSFW'05), IEEE Computer Society (2005), 62–76.

[3] Martín Abadi and Cedric Fournet. *Mobile values, new names, and secure communication.* POPL '01: Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, ACM Press (2005) 104–115

[4] Martín Abadi and Andrew D. Gordon. *A calculus for cryptographic protocols: The Spi calculus.* Fourth ACM Conference on Computer and Communications Security, ACM Press (1997), 36–47.

[5] M. Boreale and D. Sangiorgi. Bisimulation in Name-Passing Calculi without Matching Proc. of 13th IEEE Symposium on Logic in Computer Science (LICS '98), IEEE Computer Society Press, 1998.

[6] Michele Boreale, Rocco De Nicola, and Rosario Pugliese. *Proof techniques for cryptographic processes.* SIAM J. Comput. **31(3)** (2001), 947–986.

[7] Johannes Borgström. *Static equivalence is harder than knowledge.* Electr. Notes Theor. Comput. Sci., **154(3)** (2006), 45–57.

[8] Ulrik Frendrup, Hans Hüttel, and Jesper Nyholm Jensen *Modal logics for cryptographic processes.* Electr. Netes Theor. Comput. Sci. (1997), 36–47.

[9] R. Milner, J. Parrow, and D. Walker. *Modal logics for mobile processes.* Theoretical Computer Science **114(1)** (1993), 149–171.

[10] Lawrence C. Paulson. *The inductive approach to verifying cryptographic protocols.* Journal of Computer Security **6** (1998), 85–128.

[11] Michael D. Pedersen. "Logics for the Applied Pi calculus." Master's thesis, Department of Computer Science, Aalborg University (2006). BRICS Research Report RS-19-06, http://www.brics.dk/RS/06/19/.

[12] OpenSSL: The open source toolkit for SSL/TLS. http://www.openssl.org/.