Encoding cryptographic primitives in a calculus with polyadic synchronization

Joana Martinho [*]

Dep. of Mathematics, Instituto Superior Técnico,
Technical University of Lisbon.
Av. Rovisco Pais, 1. 1049-001 Lisboa, Portugal
E-mail: joana_dk@portugalmail.pt

António Ravara [†]

Security and Quantum Information Group,
Instituto de Telecomunicações, and
Dep. of Mathematics, Instituto Superior Técnico,
Technical University of Lisbon.
Av. Rovisco Pais, 1. 1049-001 Lisboa, Portugal
E-mail: aravara@ist.utl.pt

Abstract

We thoroughly study the behavioral theory of epi, a π-calculus extended with polyadic synchronization. We show that the natural contextual equivalence, barbed congruence, coincides with early bisimilarity, which is thus its co-inductive characterization. Moreover, we relate early bisimilarity with the other usual notions, ground, late and open, obtaining a lattice of equivalence relations that clarifies the relashionship among the "standard" bisimilarities.

Furthermore, we apply the theory developed to obtain an expressiveness result: epi extended with (symmetrical) key encryption primitives may be fully abstractly encoded in the original epi calculus. The proposed encoding is sound and complete with respect to barbed congruence. Therefore, cryptographic epi (crypto-epi) gets behavioral theory for free, what contrasts with other process languages with cryptographic constructs that usually require a big effort to develop such theory.

Therefore, it is thus possible to use crypto-epi to analyze and to verify properties of security protocols using equational reasoning. To illustrate this claim, we prove the correctness of a protocol of secure message exchange.

Categories and Subject Descriptors D.2.4 [*Program Verification*]: Correctness Proofs; F.1.1 [*Models of Computation*]: Process Algebra; F.1.3 [*Specification, Verification, and Reasoning about Programs*]: Logics and Meanings of Programs; F.3.3 [*Program constructs*]: Control Primitives

General Terms Behavioral Theory, Cryptographic Mobile Calculus

Keywords Barbed Congruence, (Early) Bisimulation, Cryptographic Primitives, Fully Abstract Encoding, π-calculus, Polyadic Synchronization

1. Introduction

We study herein the behavioral theory of a π-calculus where, instead of on simple names, processes synchronize in vectors of names (epi, acronym standing for *extended* π-calculus). To illustrate the expressive power of the calculus and a possible application area, we show that some cryptographic primitives are derivable in epi. Therefore, if one extends the calculus with such primitives, the resulting process language enjoys the same theory of the original language, and thus one may use it to prove properties of security protocols.

Extended π-calculus. The π-calculus with polyadic synchronization (epi), proposed by Carbone and Maffeis [CM03] is an extension of the π-calculus of Milner, Parrow, and Walker [MPW92, SW01] that generalizes the synchronization mechanism, based on handshaking, *i.e.*, the simultaneous execution of input/output actions, by allowing channel names to be composite.

The fact that in epi communication is only established if the channel vectors match element-wise, enhances its expressive power with respect to the π-calculus. In particular, Carbone and Maffeis show that the matching construct[1] can be encoded in the π-calculus with polyadic synchronisation but not in the π-calculus. In addition, they also prove that the higher the degree of synchronization (*i.e.* the maximum length of the channel vectors), the higher the expressive power of the calculus.

Carbone and Maffeis did not fully developed the behavioral theory of the process language they proposed. Defining a grammar and an operational semantics yields a description language and a rigorous definition of its computational behavior, but a calculus (in the logical sense) requires a theory to equate terms. A process *calculus* is achieved either by axiomatically, inductively, or co-inductively defining a behavioral equivalence (ideally a congruence).

Goals and contributions. The aim of this paper is twofold: to develop the behavioral theory of epi, defining a contextual equivalence and looking for its co-inductive characterization; and to use this theory to show how to define cryptographic primitives preserving it, thus allowing the calculus extended with such primitives to be used to analyze and to verify security protocols.

The first goal is hence to study in detail the behavioral semantics of epi: (1) defining an operational semantics (a late labeled transition system semantics); (2) defining the usual equivalence notions (in the context of mobile calculi): contextual and co-inductive

[*] Affiliation when developing a preliminary version of this work, until October 2004.

[†] Corresponding author.

[1] The matching construct is a process like **if** $x = y$ **then** P which compares names x and y and, if they coincide, behaves like process P; otherwise does nothing.

International Conference on Theoretical and Mathematical Foundations of Computer Science (TMFCS-08) July 7-10 2008, Orlando, FL, USA

(ground, late, early and open bisimilarities); and (3) extending results from the π-calculus to epi, namely obtaining congruence results, finding which (if any) bisimilarity coincides with the contextual equivalence, and establishing a lattice of inter-relations between the various equivalence relations;[2]. We find that, in epi, like in π, barbed congruence coincides with early congruence (early bisimilarity closed for all substitutions), and thus, we have a co-inductive characterization of the "natural" contextual equivalence of the calculus. Moreover, we relate all these "standard" notions of co-inductive equivalences, ground, late, early, and open, bisimilarities and congruences, obtaining a lattice of equivalence relations that clarifies the relationship among them. To our knowledge, this is original work, and provides to epi the basic behavioral theory of a mobile calculus.

The second goal is to use the theory developed to show that epi extended with nondeterministic, symmetrical, key encryption primitives may be used to analyze and to verify security protocols. To explore this possible application area, and to further show the expressive power of epi, we define a new calculus, the cryptographic π-calculus with polyadic synchronisation (crypto-epi), an extension of epi with the referred cryptographic primitives (in the spirit of the spi-calculus of Abadi and Gordon [AG97] — itself an extension of the π-calculus with constructs that allow for encryption and decryption of messages — or of the applied π-calculus of Abadi and Fournet [AF01] — another extension of the π-calculus with a term algebra). These two primitives are suggested by Carbone and Maffeis in the introduction of their paper [CM03] as another argument to support the expressiveness of the calculus they propose—epi. However, they did not defined nor studied its extension with such primitives. Herein, we formally define crypto-epi: (1) adding to the grammar of epi primitives for (symmetrically) encoding and decoding names; (2) providing transition rules to deal with these constructs, enriching the labeled transition system of epi; (3) extending results from epi to crypto-epi, namely showing that the new constructs preserve early bisimilarity. Then we show that crypto-epi is fully abstractly encoded, with respect to barbed congruence, in the original π-calculus with polyadic synchronisation, thus reflecting the behavioral theory of epi back to crypto-epi, and allowing the usual reasoning principles using behavioral equivalences to be used in the latter.

The encoding is also proposed by Carbone and Maffeis in the introduction of their paper, but they do not study its properties. To our knowledge, our result is original: not only it shows that these cryptographic primitives may be defined in epi as programming constructs and do not need to be primitive, re-enforcing the expressive power of epi, but also it provides standard behavioral theory to a cryptographic mobile process calculus. Moreover, since the results closely follow those of the pi-calculus, it should be straightforward to adapt tools like the Mobility Workbench [Vic94, VM94] to epi and crypto-epi, achieving a powerful tool to prove by equational reasoning properties of security protocols. Note that other cryptographic calculi like the spi-calculus or the applied pi-calculus have a more evolved and sometimes cumbersome behavioral theory. The extra structure for data handling severely complicates equational reasoning: naïve adaptation of bisimulations are not adequate; new notions developed are "heavy", and difficult to automate [AF01, AG97, BAF07, BNP02, BN05]. To illustrate the use of the theory developed, we prove the correctness of a protocol of secure message exchange.

Structure of the paper. We structure the presentation of our work in the following manner:

$P ::=$	processes	$\pi ::=$	prefixes
$\mathbf{0}$	*inaction*	τ	*internal*
$\mid \pi.P$	*prefix*	$\mid x_1 \cdot \ldots \cdot x_k(y)$	*input*
$\mid !P$	*replication*	$\mid \overline{x_1 \cdot \ldots \cdot x_k}\langle y\rangle$	*output*
$\mid (\nu x)P$	*restriction*		
$\mid (P \mid P)$	*parallel composition*		
$\mid (P+P)$	*choice*		

Figure 1. crypto-epi syntax.

- In Section 2 we introduce the syntax and a late labeled transition semantics of the π-calculus with polyadic synchronisation, as first proposed by Carbone and Maffeis.
- In Section 3 we define the four usual co-inductive notions of equivalence (ground, late, early and open bisimilarity), and compare these notions, concluding that they relate to each other just as in the π-calculus. We further introduce the notions of barbed bisimilarity, equivalence and congruence, and conclude the latter coincides with early congruence. Although relying on a similar result obtained for the π-calculus [San92], the proof of the coincidence of the notions in epi requires several adjustments.
- In Section 4 we extend the π-calculus with polyadic synchronisation with the cryptographic primitives proposed by Carbone and Maffeis in the introduction of their paper [CM03]. In addition to their work, we give an operational semantics to the new calculus, adding new rules to the original labeled transition system, and moreover, we analyze in detail a simple cryptographic protocol, proving it correct. Furthermore, we prove fully abstract (with respect to barbed congruence) the encoding they propose of the cryptographic constructs in epi.
- Section 5 concludes the paper, listing our contributions and giving directions for future research.

Due to the lack of space, we do not present herein the proofs of the results obtained. These may be found in a technical report [MR07].

2. epi: π-calculus with Polyadic Synchronization

The π-calculus with polyadic synchronization, epi, is a variant of the π-calculus where the channels can consist of sequences of names and communication is established if and only if the channel vectors match element-wise.[3]

2.1 Syntax

We introduce the syntax of the calculus in detail and also mention some of the main differences between this and the π-calculus. These differences will be explained in further detail in subsequent parts throughout this section.

DEFINITION 2.1. *Processes*
Let N be a countable set of names and $x, x_1, ..., x_k, y$ range over N for some $k \in \mathbb{N}$. The grammar in Figure 1 defines the class of processes *$\mathcal{P}_S$, ranged over by P, Q.*

The decreasing order of precedence of operators follows that of the definition, where the prefix operator has the highest precedence. In what follows we use the notation π for $\pi.\mathbf{0}$, and $(\nu z, w)P$ for $(\nu z)(\nu w)P$.

All operators used here are also present in the π-calculus and their behavior is as expected. Nonetheless, note that restriction is

[2] The result is a lattice similar to that of the π-calculus.

[3] We call π-calculus with biadic synchronization the particular case of the π-calculus with polyadic synchronization where the composite channels have at most two names.

Action	Description	$\mathsf{fn}(\alpha)$	$\mathsf{bn}(\alpha)$
τ	internal	$\emptyset$	$\emptyset$
$\overline{u}\langle y\rangle$	free output	$\mathsf{nm}(u) \cup \{y\}$	$\emptyset$
$\overline{u}(y)$	bound output	$\mathsf{nm}(u)$	$\{y\}$
$u(y)$	input	$\mathsf{nm}(u)$	$\{y\}$

Table 1. Actions.

P	$\mathsf{fn}(P)$	$\mathsf{bn}(P)$
$\mathbf{0}$	$\emptyset$	$\emptyset$
$\pi.Q$	$\mathsf{fn}(\pi) \cup (\mathsf{fn}(Q) \backslash \mathsf{bn}(\pi))$	$\mathsf{bn}(\pi) \cup \mathsf{bn}(Q)$
$!Q$	$\mathsf{fn}(Q)$	$\mathsf{bn}(Q)$
$(\nu y)Q$	$\mathsf{fn}(Q)\backslash\{y\}$	$\{y\} \cup \mathsf{bn}(Q)$
$(Q \mid R)$	$\mathsf{fn}(Q, R)$	$\mathsf{bn}(Q, R)$
$(Q + R)$	$\mathsf{fn}(Q, R)$	$\mathsf{bn}(Q, R)$

Table 2. Names in Processes.

made on names as in the π-calculus and not on composite channels: this allows for *partial restriction*.

One should also note that in the π-calculus with polyadic synchronization it is not necessary to include the match operator since it can be encoded in the calculus. This is not possible in a 'sensible' manner using the original π-calculus that, therefore, takes the match operator as a primitive. This important separation result between the two calculi was obtained by Carbone and Maffeis [CM03], and it is the central expressiveness result about epi.

Consider $u = x_1 \cdot ... \cdot x_k$ and $\overline{u} = \overline{x_1 \cdot ... \cdot x_k}$, where $k \in \mathbb{N}$, represent respectively the input and output channel vectors. Then, $\mathsf{nm}(u) = \mathsf{nm}(\overline{u}) = \{x_1, ..., x_k\}$. As in the π-calculus, there are four possible kinds of *actions* α in the present calculus, as seen in Table 1. Let $\mathsf{bn}(\alpha)$ denote the set of *bound names* in α, $\mathsf{fn}(\alpha)$ the set of *free names* in α and $\mathsf{nm}(\alpha)$ the set of all *names* in α (the union of the previous two sets). The respective notions for prefixes, *i.e.*, $\mathsf{fn}(\pi)$, $\mathsf{bn}(\pi)$, and $\mathsf{nm}(\pi)$, are defined similarly. Furthermore, the notions of bound and free names in a process P, denoted by $\mathsf{bn}(P)$ and $\mathsf{fn}(P)$ respectively, follow from those of the π-calculus. Table 2 presents the rigorous definition of these notions, where $\mathsf{nm}(P)$ denotes the names in the process P. Let $\mathsf{fn}(P_1, P_2) = \mathsf{fn}(P_1) \cup \mathsf{fn}(P_2)$, and consider similar definitions for $\mathsf{bn}(P_1, P_2)$ and $\mathsf{nm}(P_1, P_2)$.

Note that we sometimes use polyadic CCS-like prefixes $\overline{a \cdot w}$ and $a \cdot y$ where no item is being sent or expected to be received. We do this to highlight the fact that what could be transmitted is irrelevant, the problem lies in the synchronization of the composite channels. In general, $\overline{u}.P$ will be used as shorthand for $\overline{u}\langle y\rangle .P$ for some y, and $u.P$ will be used as shorthand for $u(y).P$ where $y \notin \mathsf{fn}(P)$.

Substitution and α-convertibility are defined as in the π-calculus [SW01], though we now require that the latter takes into account the possibility of composite channels. Note that given a substitution $\sigma = \{w/z\}$ we denote the result of applying σ to z as $\sigma(z)$. In this case, we then have that $\sigma(z) = w$. Moreover, substitution may imply the renaming via α-conversion of bound actions to avoid unwanted captures of free names.

2.2 Late Labeled Transition Semantics

We define herein a late labeled transition semantics of the π-calculus with polyadic synchronization. In addition, we provide examples that reflect the differences between this calculus and the π-calculus.

(PREFIX)
$$\frac{-}{\alpha.P \xrightarrow{\alpha} P}$$

(CH1)
$$\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$$

(PAR1)
$$\frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \qquad \text{where } \mathsf{bn}(\alpha) \cap \mathsf{fn}(Q) = \emptyset$$

(RES)
$$\frac{P \xrightarrow{\alpha} P'}{(\nu x)P \xrightarrow{\alpha} (\nu x)P'} \qquad \text{where } x \notin \mathsf{nm}(\alpha)$$

(REP-ACT)
$$\frac{P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P'|!P}$$

(REP-COMM)
$$\frac{P \xrightarrow{\overline{u}\langle x\rangle} P' \quad P \xrightarrow{u(z)} P''}{!P \xrightarrow{\tau} (P'|P''\{x/z\})|!P}$$

(REP-CLOSE)
$$\frac{P \xrightarrow{\overline{u}(x)} P' \quad P \xrightarrow{u(x)} P''}{!P \xrightarrow{\tau} (\nu x)(P'|P'')|!P} \qquad \text{where } x \notin \mathsf{fn}(P)$$

(OPEN)
$$\frac{P \xrightarrow{\overline{u}\langle x\rangle} P'}{(\nu x)P \xrightarrow{\overline{u}(x)} P'} \qquad \text{where } x \notin \mathsf{nm}(\overline{u})$$

(CLOSE1)
$$\frac{P \xrightarrow{\overline{u}(x)} P' \quad Q \xrightarrow{u(x)} Q'}{P|Q \xrightarrow{\tau} (\nu x)(P'|Q')}$$

(COMM1)
$$\frac{P \xrightarrow{\overline{u}\langle x\rangle} P' \quad Q \xrightarrow{u(z)} Q'}{P|Q \xrightarrow{\tau} P'|Q'\{x/z\}}$$

(CONV)
$$\frac{P \xrightarrow{\alpha} P'}{Q \xrightarrow{\alpha} P'} \qquad \text{if } Q =_{\alpha} P$$

Figure 2. Late transition rules.

DEFINITION 2.2. *Late labeled transition relation*
Let $u = x_1 \cdot ... \cdot x_k$*, where* $k \in \mathbb{N}$*. The* late labeled transition relation $\xrightarrow{\alpha} \subseteq \mathcal{P}_S \times \mathcal{P}_S$*, where* α *is an action, is the smallest relation generated by the set of rules in Figure 2.*[4]

The rules follow in a straightforward manner those of the π-calculus, considering now vectors of names as channels. Note once again that the restriction rule, RES, considers singular and not composite names, *i.e.*, restriction is *partial*. Nevertheless, we enforce an all-or-nothing behavior, that is, we require the match of all the names in the vector channel to allow synchronization. The following example reflects the consequences of this type of restriction.

EXAMPLE 2.3. *Let* $P = (\nu x_1)\overline{x_1 \cdot x_2}\langle y\rangle$ *and* $Q = \overline{x_1 \cdot x_2}\langle y\rangle$*. Then, P cannot perform the input action because of the restriction in one of its channel names, while Q can. Consider now* $P = x(y)\overline{y \cdot z}\langle v\rangle \mid \overline{x}\langle w\rangle$*. Its reduction performs a substitution in (only) one of the channel names, yelding* $\overline{w \cdot z}\langle v\rangle$*.*

[4] Note that not included in the figure are four rules: the symmetric form CH2 of CH1 which has $Q + P$ instead of $P + Q$, and the symmetric forms PAR2, COMM2 and CLOSE2 of PAR1, COMM1, CLOSE1 in which the roles of the left and right components are swapped.

3. Observational Semantics

In this section we develop the behavioral theory of epi. In short, we define a contextual equivalence for epi—barbed congruence—and find its co-inductive characterization. Following the literature for the π-calculus [MPW92, San96], with the necessary adjustments we introduce the "standard" notions of bisimilarity: ground, late, early, and open. Then study their preservation by the operators of epi and inter-relate these notions, getting a lattice of discriminating power. Finally, we show that early congruence (early bisimilarity closed for all substitutions) coincides with barbed congruence, being thus its co-inductive characterization.

Although not surprising, these results are technically difficult, and some proofs deviate from those in the π-calculus. This work is necessary to provide to epi behavioral theory.

3.1 A Contextual Equivalence

In this part we prove that a "natural" contextual equivalence coincides with early bisimilarity, which is thus a co-inductive characterization of the latter. We rely on a similar result obtained for the π-calculus by Sangiorgi [San92].

DEFINITION 3.1. *Barbs*
The predicate 'P exhibits barb *β', written $P \downarrow_\beta$, is defined by:*

- $P \downarrow_u$ *if P can perform an input action on channel u*
- $P \downarrow_{\overline{u}}$ *if P can perform an output action on channel u*

A *barb* is an input or output channel identifier. Note that the predicate just defined concerns only visible and immediate possible action. We now introduce the notion of barbed bisimilarity as proposed by Milner and Sangiorgi [MS92].

DEFINITION 3.2. *Barbed bisimilarity*

1. *A binary symmetric relation $\mathcal{S}$ is a* barbed bisimulation *if $P\mathcal{S}Q$ implies:*
   - *if $P \downarrow_\beta$ then $Q \downarrow_\beta$ for each barb β*
   - *if $P \xrightarrow{\tau} P'$ then there exists a Q' such that $Q \xrightarrow{\tau} Q'$ and $P'\mathcal{S}Q'$*
2. *Processes P and Q are* barbed bisimilar *if $P\mathcal{S}Q$ for some barbed bisimulation $\mathcal{S}$.*
3. Barbed bisimilarity*, written $\sim_b$, is the greatest barbed bisimulation.*

Barbed bisimilarity is a much coarser relation than the ones introduced so far. The following example illustrates the difference between barbed bisimilarity and those notions of bisimilarity.

EXAMPLE 3.3. *Let $P = \overline{m}\langle n\rangle\,.\overline{m}\langle n\rangle$ and $Q = \overline{m}\langle n\rangle$. Then, P and Q are barbed bisimilar since their only barb is $\overline{m}$. However, P and Q are not ground, nor late, nor early nor open bisimilar since $P \xrightarrow{\overline{m}\langle n\rangle} \overline{m}\langle n\rangle$ and $Q \xrightarrow{\overline{m}\langle n\rangle} \mathbf{0}$, which are obviously not bisimilar.*

Note that barbed bisimilarity is not a congruence since it is not preserved by parallel composition, nor by replication, nor by substitution. Nonetheless, barbed bisimilarity is preserved by the remaining operators.

PROPOSITION 3.4. *The relation $\sim_b$ is preserved by prefixing, restriction and choice operators.*

Closing barbed bisimilarity for parallel composition yields an equivalence notion.

DEFINITION 3.5. *Barbed equivalence*
Two processes P and Q are barbed equivalent*, written $\sim_{beq}$, if $P \mid R \sim_b Q \mid R$ for every process R.*

In order to define barbed congruence we must first introduce the notion of context. Contexts are processes with a "hole".

DEFINITION 3.6. *Barbed congruence*

1. *A* context *is obtained when a 'hole' $[\cdot]$ replaces a process in $P \in \mathcal{P}_S$.*
2. *The process obtained by replacing the $[\cdot]$ in C by P, where C is a context and P a process, is denoted by $C[P]$.*
3. *Two processes P and Q are* barbed congruent*, written $\simeq_b$, if $C[P] \sim_b C[Q]$ for every context $C[\cdot]$.*

We now extend the result that establishes an alternative definition of barbed congruence in the π-calculus, as done by Sangiorgi and Walker [SW01], to epi.

LEMMA 3.7. *$P \simeq_b Q$ if and only if $P\sigma \sim_{beq} Q\sigma$ for any substitution σ*

The notion of barbed congruence was proposed by Milner and Sangiorgi [MS92], while the less demanding notion of barbed equivalence was later proposed by Sangiorgi [San92]. Note that barbed equivalence and barbed congruence do not coincide, as there are processes barbed equivalent but not barbed congruent (see Example 3.34 in the long version of this paper [MR07]).

3.2 Four Notions of Bisimilarity

Seeking for a co-inductive characterization of barbed congruence, we define the usual notions of bisimilarity, and inter-relate them. The first notion we will consider is that of ground bisimilarity, where there is no name instantiation.

DEFINITION 3.8. *Ground bisimilarity*

1. *A binary symmetric relation $\mathcal{S}$ is a* ground bisimulation *if $P\mathcal{S}Q$ implies:*
   if $P \xrightarrow{\alpha} P'$ where $\mathsf{bn}(\alpha) \cap \mathsf{fn}(P, Q) = \emptyset$ then there is a Q' such that $Q \xrightarrow{\alpha} Q'$ and $P'\mathcal{S}Q'$.
2. *Processes P and Q are* ground bisimilar *if $P\mathcal{S}Q$ for some ground bisimulation $\mathcal{S}$.*
3. Ground bisimilarity*, written $\sim_g$, is the largest ground bisimulation.*[5]

The notion of ground bisimilarity is very simple since a process merely has to imitate the other in its possible transitions and vice versa without considering name instantiation. Unfortunately, as in the π-calculus, a consequence of this is that ground bisimilarity is not preserved by the parallel composition operator, as seen in the following example.

EXAMPLE 3.9. *Let $P = (\nu a)(z(w).\overline{a \cdot w}\langle c\rangle \mid a \cdot y(b))$ and $Q = z(w)$. Then both P and Q are ground bisimilar since after performing the input action they both become inactive. Conversely,*

$$P' = P \mid \overline{z}\langle y\rangle \xrightarrow{\tau} (\nu a)(\overline{a \cdot y}\langle c\rangle \mid a \cdot y(b)),$$

which can also perform an internal action, while $Q' = Q \mid \overline{z}\langle y\rangle$ can only perform one internal action and then becomes inactive. We can then conclude that although P and Q are ground bisimilar, P' and Q' are not ground bisimilar.

Ground bisimilarity is not preserved by replication either. A counter-example may be found in the long version of this paper.[6]

[5] The existence and uniqueness of a largest bisimulation is a direct consequence of the Knäster-Tarski's Fixed Point Theorem.

[6] Several propositions in the remaining of this section present results with strict inclusions. The counter-examples may be found in the long version of this paper.

Nonetheless, ground bisimilarity is preserved, just like in the π-calculus, by the remaining operators.

LEMMA 3.10. *The relation $\sim_g$ is preserved by the restriction, the prefixing and the choice operators.*

We now introduce the notions of *late* and *early bisimilarity*, which differ in their treatment of name instantiation for input actions. The definitions of these notions are standard. In *late* bisimilarity we require that the derivative of a process simulates the derivative of the other process (and vice versa) for all possible instantiations of the bound parameter. It is called late because the choice of the name instantiation is made *after* the choice of the derivative.

DEFINITION 3.11. *Late bisimilarity*
Let $u = x_1 \cdot \ldots \cdot x_k$ where $k \in \mathbb{N}$.

1. *A binary symmetric relation $\mathcal{S}$ is a* late bisimulation *if $P\mathcal{S}Q$ implies:*
   - *if $P \xrightarrow{\alpha} P'$ where $\alpha = \overline{u}\langle y\rangle\,, \overline{u}(y)$ or τ and $\mathsf{bn}(\alpha) \cap \mathsf{fn}(P, Q) = \emptyset$ then there is a Q' such that $Q \xrightarrow{\alpha} Q'$ and $P'\mathcal{S}Q'$.*
   - *if $P \xrightarrow{u(y)} P'$ where $y \notin \mathsf{fn}(P, Q)$ then there is a Q' such that $Q \xrightarrow{u(y)} Q'$ and for each w, $P'\{w/y\}\mathcal{S}Q'\{w/y\}$.*
2. *Two processes P and Q are* late bisimilar *if $P\mathcal{S}Q$ for some late bisimulation $\mathcal{S}$.*
3. Late bisimilarity*, written $\sim_l$, is the largest late bisimulation.*

In *early* bisimilarity we require that under the same possible name instantiation there is a derivative of each of the processes that simulates the other and vice versa. It is named early because the choice of the name instantiation is made *before* the choice of the derivative.

DEFINITION 3.12. *Early bisimilarity*
Let $u = x_1 \cdot \ldots \cdot x_k$ where $k \in \mathbb{N}$.

1. *A binary symmetric relation $\mathcal{S}$ is an* early bisimulation *if $P\mathcal{S}Q$ implies:*
   - *if $P \xrightarrow{\alpha} P'$ where $\alpha = \overline{u}\langle y\rangle\,, \overline{u}(y)$ or τ and $\mathsf{bn}(\alpha) \cap \mathsf{fn}(P, Q) = \emptyset$ then there is a Q' such that $Q \xrightarrow{\alpha} Q'$ and $P'\mathcal{S}Q'$.*
   - *if $P \xrightarrow{u(y)} P'$ where $y \notin \mathsf{fn}(P, Q)$ then for each w there is a Q' such that $Q \xrightarrow{u(y)} Q'$ and $P'\{w/y\}\mathcal{S}Q'\{w/y\}$.*
2. *Two processes P and Q are* early bisimilar *if $P\mathcal{S}Q$ for some early bisimulation $\mathcal{S}$.*
3. Early bisimilarity*, written $\sim_e$, is the largest early bisimulation.*

Similarly to what happens in the π-calculus, in the π-calculus with polyadic synchronization both late and early bisimilarity are not preserved by input prefixing, but are preserved by all other operators.

PROPOSITION 3.13. *The relations $\sim_l$ and $\sim_e$ are preserved by all operators except input prefixing.*

Moreover, as in the original π-calculus congruences for late and early bisimilarity, $\simeq_l$ and $\simeq_e$, are achieved by closing the equivalences over all name substitutions [MPW92]. The relation between the notions of late bisimilarity and late congruence, and of early bisimilarity and early congruence, are shown in the following proposition.

PROPOSITION 3.14. *$\simeq_l \subset \sim_l$ and $\simeq_e \subset \sim_e$.*

The notion of open bisimilarity was introduced by Sangiorgi and proved to be a congruence relation in the π-calculus [San96]. That is also the case here: in epi, open bisimilarity is a congruence.

DEFINITION 3.15. *Open bisimilarity*

1. *A binary symmetric relation $\mathcal{S}$ is an* open bisimulation *if $P\mathcal{S}Q$ implies for every substitution σ:*
   If $P\sigma \xrightarrow{\alpha} P'$ where $\mathsf{bn}(\alpha) \cap \mathsf{fn}(P\sigma, Q\sigma) = \emptyset$ then there is a Q' such that $Q\sigma \xrightarrow{\alpha} Q'$ and $P'\mathcal{S}Q'$.
2. *Two processes P and Q are* open bisimilar *if $P\mathcal{S}Q$ for some open bisimulation $\mathcal{S}$.*
3. Open bisimilarity*, written $\sim_o$, is the largest open bisimulation.*

As expected, open bisimilarity is a congruence.

PROPOSITION 3.16. *The relation $\sim_o$ is preserved by all operators.*

Thus the congruence properties appear to stem directly from those of the π-calculus. However, ground bisimilarity is a full congruence in the asynchronous π-calculus without match [San00] (and a similar result holds for late and for early bisimilarity [HHK95]), but this result does *not* hold if we consider the asynchronous π-calculus with polyadic synchronization, as seen in Example 3.9. Matching does not need to be considered as a primitive in the π-calculus with polyadic synchronization (synchronous or asynchronous) since it can be derived. Therefore, ground, late and early bisimilarities are *not* congruences in the asynchronous π-calculus with polyadic synchronisation (without match).

We now analyze the relationships between the bisimilarity relations previously defined and present a general diagram that summarizes these results in Corollary 3.20. The results and proofs are similar to those presented for the π-calculus [MPW92, Qua99]. The largest open bisimulation is itself a late bisimulation, and it is also included in late congruence.

PROPOSITION 3.17. *$\sim_o \subset \sim_l$ and $\sim_o \subset \simeq_l$.*

Late bisimilarity is itself an early bisimulation, although the reverse does not hold. The same result holds if we consider the notions of late and early congruences instead of late and early bisimilarity.

PROPOSITION 3.18. *$\sim_l \subset \sim_e$ and $\simeq_l \subset \simeq_e$.*

Our last result shows that if two processes are early bisimilar then they are also ground bisimilar, although the reverse does not hold.

PROPOSITION 3.19. *$\sim_e \subset \sim_g$.*

We now summarize the results presented in the following diagram where $\rightarrow$ stands for strict inclusion $\subset$.

COROLLARY 3.20.

$$\begin{tikzcd}
\sim_o \arrow[r] \arrow[dr] & \sim_l \arrow[r] & \sim_e \arrow[r] & \sim_g \\
 & \simeq_l \arrow[u] \arrow[r] & \simeq_e \arrow[u] \arrow[ur] &
\end{tikzcd}$$

Sangiorgi obtained an alternative characterization of barbed equivalence by proving it coincided with early bisimilarity [San92]. We extend that result for the π-calculus with polyadic synchronisation, completing the behavioral theory.

THEOREM 3.21. *$\sim_e = \sim_{beq}$*

COROLLARY 3.22. *$\simeq_e = \simeq_b$*

4. Encoding Cryptographic Primitives

To our knowledge, the first mention of a possible encoding of a calculus with cryptographic primitives into a calculus with polyadic synchronization was put forth by Abadi and Gordon [AG97]. The idea can be summarized in the following way: the sending of a message m encrypted under a key k over a channel a can be seen as $\overline{a \cdot k}\langle m\rangle\,.P$. In order to receive this message, the other party needs to know the channel where the message is being transmitted and the key, which could be represented as $a \cdot k(m).P$.

An encoding of symmetrical key encryption primitives into π-calculus with polyadic synchronisation is proposed by Carbone and Maffeis in the introduction of their paper to further illustrate its expressive power [CM03].

$$[|\,\mathsf{encrypt}\, m \looparrowright^k x \,\mathsf{in}\, P\,|] \stackrel{\text{def}}{=} (\nu x)(!\overline{x \cdot k}\langle m\rangle \mid [|\,P\,|])$$

$$[|\,\mathsf{decrypt}\, x \looparrowright^k m \,\mathsf{in}\, P\,|] \stackrel{\text{def}}{=} x \cdot k(m).[|\,P\,|]$$

However, Carbone and Maffeis do not define the semantics of the primitives, and thus do not study the properties of the encoding (as moreover, they have not developed the behavioral theory of epi). Herein we do all that work: we first add to epi two nondeterministic, symmetrical, key encryption primitives, encrypt and decrypt, defining the cryptographic π-calculus with polyadic synchronization (crypto-epi), and extend epi labeled transition system with rules dealing with these new constructs. Then we show that the new constructors preserve the bisimilarity relations defined to epi, and finally, we prove that these cryptographic primitives are derivable constructs: crypto-epi can be fully abstractly encoded in epi; thus we prove that the original calculus does not need to be extended with those primitives, at least from the point of view of expressiveness. Moreover, since the encoding is fully abstract, crypto-epi enjoys of all the behavioral theory of epi.

The main achievement here is thus a mobile calculus with cryptographic primitives enjoying the "standard" behavioral theory. Adapting analysis tools like the Mobility Workbench [Vic94, VM94] should be straightforward.

4.1 Cryptographic epi

Syntax. Consider two extra productions in the syntax of epi (*cf.* Definition 2.1): $\mathsf{encrypt}\, m \looparrowright^k x \,\mathsf{in}\, P$ and $\mathsf{decrypt}\, x \looparrowright^k m \,\mathsf{in}\, P$. The first construct nondeterministically encrypts the cipher text m under key k and returns the encrypted message as the fresh name x, to be used in the scope of P, where it occurs bound. The decryption of message x through the key k (used to encrypt the message) binds the name m in the continuation P to the original message. Notice that one, when encrypting, does not expect free occurrences of m and k in P; and when decrypting, does not expect free occurrences of x and k in P.

Labeled Transition Semantics. The rules of epi in Figure 2 (page 3), together with the rules in Figure 3 inductively define the transition semantics of crypto-epi.

The behavioral theory of epi extends naturally to this new setting. The notions of bisimilarity, introduced in Section 3, enjoy similar properties when consider the new constructs. Notice that the decrypt primitive behaves like an input prefix, thus it does not preserve ground, early or late bisimilarity, but naturally, it preserves open bisimilarity. The notion of early congruence in crypto-epi is obtained in the same manner, and the results in Corollary 3.22 also extend straightforwardly to this new setting. Therefore, the theory developed can be used to analyze and (equationally) prove properties of security protocols. To present the examples below we need to introduce some results. First, notice that syntactical equality is an early bisimilarity, and that any strong bisimilarity is strictly included in the corresponding weak version. In particular, $= \subset \simeq_e \subset \approx_e$.

$$\text{(ENC)}\qquad \frac{P \xrightarrow{\alpha} P'}{\mathsf{encrypt}\, m \looparrowright^k x \,\mathsf{in}\, P \xrightarrow{\alpha} \mathsf{encrypt}\, m \looparrowright^k x \,\mathsf{in}\, P'}$$

where $\alpha \neq \overline{u}\langle x\rangle$ and if $\alpha \in \{\overline{u}\langle y\rangle\,, \overline{u}(y), u(y)\}$ then $x \notin \mathsf{nm}(u)$

$$\text{(ENC-OPEN)}\qquad \frac{P \xrightarrow{\overline{u}\langle x\rangle} P'}{\mathsf{encrypt}\, m \looparrowright^k x \,\mathsf{in}\, P \xrightarrow{\overline{u}(x)} !\overline{x \cdot k}\langle m\rangle \mid P'}$$

where $x \notin \mathsf{nm}(u)$

$$\text{(DEC)}\qquad \frac{--}{\mathsf{decrypt}\, x \looparrowright^k y \,\mathsf{in}\, P \xrightarrow{x \cdot k(y)} P}$$

Figure 3. Late transition rules for the cryptographic constructs.

Second, the usual structural congruence laws of the π-calculus [MPW92, SW01] also hold in any bisimilarity. Therefore, we use below instances of the following laws.[7]

LEMMA 4.1 (Structural Laws).

1. *$(\mathcal{P}_S, |, \mathbf{0}\,)$ is a commutative monoïd with respect to $\simeq_e$.*
2. *$(\nu x)\mathbf{0} \simeq_e \mathbf{0}$ and $(\nu x)!\overline{x \cdot k}\langle m\rangle \simeq_e \mathbf{0}$*
3. *$(\nu x)(P \mid Q) \simeq_e (P \mid (\nu x)Q)$, if $x \notin \mathsf{fn}(P)$*

LEMMA 4.2.

1. *$\mathsf{encrypt}\, m \looparrowright^k x \,\mathsf{in}\, P \simeq_e (\nu x)(!\overline{x \cdot k}\langle m\rangle \mid P)$*
2. *$\mathsf{decrypt}\, x \looparrowright^k m \,\mathsf{in}\, P \simeq_e x \cdot k(m).P$*

PROOF. Construct the respective bisimulations containing the pair in question and, in the two last cases, the identity relation on processes. □

4.2 A secure message exchange

Sending a value in a free (*i.e.* public) channel is *insecure*, as any context (*i.e.* observer) can have access to it. Bound (*i.e.* private) channels are, in this framework, consider secure. Since one often needs to send sensitive data in public channels, we would like to show two basic properties: (1) decrypting an encrypted value with the correct key gives back the original value, and no other key produces it; and (2) sending encrypted values in public channels is secure, as observers without the right keys cannot decrypt them.

To illustrate the use of these properties (and their correctness), consider a cryptographic protocol for secure message exchange, proposed by Carbone and Maffeis [CM03], defined as $(\nu sec)(P \mid Q)$ where P and Q are the following processes.

$$P \stackrel{\text{def}}{=} (\nu k)\overline{sec}\langle k\rangle\,.public(y).\mathsf{decrypt}\, y \looparrowright^k w \,\mathsf{in}\, R$$

$$Q \stackrel{\text{def}}{=} (\nu m)sec(z).\mathsf{encrypt}\, m \looparrowright^z x \,\mathsf{in}\, \overline{public}\langle x\rangle\,.S$$

Assume that sec does not occur free neither in R nor in S, m does not occur free in R, and k and z do not occur free in S. We show the correctness of the protocol (with respect to weak bisimilarity, to ignore silent moves): an external observer cannot get neither the key k nor the clear text message m during the execution of the protocol since the transfer of the knowledge of the key is done on a secure—since private—channel (sec). Moreover, decrypting the encrypted value x with the key k (and with it only) gives back the original value m.

[7] Instead of proving each of these laws one may prove the "Harmony Lemma", allowing to establish that structural congruence is a bisimulation.

The following equation captures the correctness of the protocol.

$$(\nu sec)(P \mid Q) \approx_e (\nu k, m)(R\{m/w\} \mid \mathsf{encrypt}\, m \looparrowright^{k} x \,\mathsf{in}\, S)$$

The analysis below proves the equation. Note that the protocol is deterministic.

1. Consider the following processes.

   $P' \overset{\text{def}}{=} public(y).\mathsf{decrypt}\, y \looparrowright^{k} w \,\mathsf{in}\, R$, and
   $Q' \overset{\text{def}}{=} (\nu m)\mathsf{encrypt}\, m \looparrowright^{k} x \,\mathsf{in}\, \overline{public}\langle x\rangle\, .S\{k/z\}$.

   The first step is the transmission of the key on channel sec:

   $(\nu sec)(P \mid Q) \xrightarrow{\tau} (\nu sec, k)(P' \mid Q')$.

2. Consider now the following processes.

   $P'' \overset{\text{def}}{=} \mathsf{decrypt}\, x \looparrowright^{k} w \,\mathsf{in}\, R\{x/y\}$, and
   $Q'' \overset{\text{def}}{=} (\nu m)(!\overline{x \cdot k}\langle m\rangle \mid S\{k/z\})$.

   The next step is the transmission of the encrypted message:

   $(\nu sec, k)(P' \mid Q') \xrightarrow{\tau} (\nu sec, k, x)(P'' \mid Q'')$.

3. Finally, the encrypted message is decrypted:

   $(\nu sec, k, x)(P'' \mid Q'') \xrightarrow{\tau}$
   $(\nu sec, k, x, m)(R\{x/y\}\{m/w\} \mid (!\overline{x \cdot k}\langle m\rangle \mid S\{k/z\}))$

Since $x \notin \mathsf{fn}(R)$ and $k, m \notin \mathsf{fn}(S)$, then $R\{x/y\} = R$ and $S\{k/z\} = S$. Moreover, $sec \notin \mathsf{fn}(R) \cup \mathsf{fn}(S)$. Thus, using the laws presented above, one concludes the proof by transitivity.

$$\begin{aligned}
&(\nu sec, k, x, m)(R\{x/y\}\{m/w\} \mid (!\overline{x \cdot k}\langle m\rangle \mid S\{k/z\}))\\
&= (\nu sec, k, x, m)(R\{m/w\} \mid (!\overline{x \cdot k}\langle m\rangle \mid S))\\
&\simeq_e (\nu k, m)(R\{m/w\} \mid (\nu x)(!\overline{x \cdot k}\langle m\rangle \mid S))\\
&\simeq_e (\nu k, m)(R\{m/w\} \mid \mathsf{encrypt}\, m \looparrowright^{k} x \,\mathsf{in}\, S)
\end{aligned}$$

4.3 A Fully Abstract Encoding

In order to prove the soundness and completeness of the encoding with respect to barbed congruence, which we proved in Corollary 3.22 to coincide with early congruence, we build on successive auxiliary results.

Note that we will consider as a target language a sub-calculus of epi without summation (as we just saw cryptographic protocols do not necessarily make use of this construct). We shall consider this calculus with and without cryptographic primitives (encoding the first, crypto-epi, in the second, epi) because the proof is lighter. Henceforth, whenever we write P we refer to a process of the cryptographic π-calculus with polyadic synchronization.

LEMMA 4.3. *Substitution Lemma*
$[|\, P\sigma \,|] = [|\, P \,|]\sigma$, *for any substitution* σ.

The following lemma shows a strong operational correspondence between the actions of a process and the actions of its encoding.

LEMMA 4.4. *Operational Correspondence*

1. *If* $[\![P]\!] \xrightarrow{\alpha} Q$ *then there is a* P' *such that* $P \xrightarrow{\alpha} P'$ *and* $[|P'|] = Q$.
2. *If* $P \xrightarrow{\alpha} P'$ *then* $[|P|] \xrightarrow{\alpha} [|P'|]$.

The following lemma prepares the ground for proving the soundness and the completeness of the encoding.

LEMMA 4.5.

1. *If* $[|\, P \,|] \sim_e [|\, Q \,|]$ *then* $P \sim_e Q$.
2. *If* $[|\, P \,|] \sim_{beq} [|\, Q \,|]$ *then* $P \sim_{beq} Q$.

PROOF.

1. We prove that $\mathcal{R} = \{(P, Q) : [|\, P \,|] \sim_e [|\, Q \,|]\}$ is an early bisimulation (*cf.* Definition 3.12 in page 5).

   Case $P \xrightarrow{\alpha} P'$ (the case $Q \xrightarrow{\alpha} Q'$ is similar, and we omit its analysis).

   Then, by Lemma 4.4.2 we have that $[\![P]\!] \xrightarrow{\alpha} [\![P']\!]$. Since by hypothesis $[|\, P \,|] \sim_e [|\, Q \,|]$ then there is a Q' such that $[|\, Q \,|] \xrightarrow{\alpha} Q'$, and by Lemma 4.4.1 we have that there is a Q'' such that $Q \xrightarrow{\alpha} Q''$, where $[|\, Q'' \,|] = Q'$.

   We split the proof according to the possible transitions of $[|\, P \,|]$.

   Case $\alpha \in \{\tau, \overline{u}y, \overline{u}(y)\}$, where $\mathsf{bn}(\alpha) \cap \mathsf{fn}(P, Q) = \emptyset$.

   By definition of $\sim_e$ we have that $[|\, P' \,|] \sim_e [|\, Q'' \,|]$ and therefore $P'\mathcal{R}Q''$.

   Case $\alpha = u(y)$ where $y \notin \mathsf{fn}(P, Q)$. The reasoning is similar to the one above.

   By definition of $\sim_e$ we have $[|\, P' \,|]\{w/y\} \sim_e [|\, Q'' \,|]\{w/y\}$, and by application of Lemma 4.3 we know that $[|\, P'\{w/y\}|] \sim_e [|\, Q''\{w/y\}|]$ holds as well. Therefore, we conclude that $P'\{w/y\}\mathcal{R}Q''\{w/y\}$.

2. Follows directly from Lemma 4.5.1 and Theorem 3.21, where it was proved that early bisimulation coincides with barbed equivalence.

□

We are now in a position to prove a main result: there is a fully abstract encoding of the cryptographic primitives in epi.

THEOREM 4.6. *Soundness*
If $[|\, P \,|] \simeq_b [|\, Q \,|]$ *then* $P \simeq_b Q$

PROOF. If $[|\, P \,|] \simeq_b [|\, Q \,|]$ then for any substitution σ we have that $[|\, P \,|]\sigma \sim_{beq} [|\, Q \,|]\sigma$. By Lemma 4.3 we then know that $[|\, P\sigma \,|] \sim_{beq} [|\, Q\sigma \,|]$, and by Lemma 4.5.2 we have, as required, $P\sigma \sim_{beq} Q\sigma$. □

LEMMA 4.7.

1. *If* $P \sim_e Q$ *then* $[|\, P \,|] \sim_e [|\, Q \,|]$.
2. *If* $P \sim_{beq} Q$ *then* $[|\, P \,|] \sim_{beq} [|\, Q \,|]$.

PROOF. Similar to the one of the previous lemma. □

THEOREM 4.8. *Completeness*
If $P \simeq_b Q$ *then* $[|\, P \,|] \simeq_b [|\, Q \,|]$.

PROOF. If $P \simeq_b Q$ then for any substitution σ we have that $P\sigma \sim_{beq} Q\sigma$. By Lemma 4.7.2 then $[|\, P\sigma \,|] \sim_{beq} [|\, Q\sigma \,|]$ and by Lemma 4.3 we know that $[|\, P\sigma \,|] = [|\, P \,|]\sigma$, thus we conclude that $[|\, P \,|]\sigma \sim_{beq} [|\, Q \,|]\sigma$, *i.e.*, $[|\, P \,|] \simeq_b [|\, Q \,|]$. □

5. Conclusions and Future Work

The various variants of π-calculus possess a very rich behavioral theory, with contextual equivalences characterized by bisimulations, and with axiomatic laws for reasoning about programs. However, the extra structure for data handling in cryptographic calculi like the Applied π-calculus or Spi, severely complicates equational reasoning: naïve adaptation of bisimulations are not adequate; new notions developed are “heavy”, and difficult to automate [AF01, AG97, BAF07, BNP02, BN05].

Our contribution is this: we provide standard behavioral theory for a mobile calculus with nondeterministic, symmetrical key encryption primitives. This work may be used not only to directly analyze security protocols (possibly defining other cryptographic primitives), but also to study the relationship with the other calculi, comparing the observational equivalences and trying to define encodings. Moreover, adapting analysis tools like the Mobility Workbench [Vic94, VM94] should be straightforward.

Aim and achievements. One aim of this work is to show that the π-calculus with polyadic synchronization, epi, is expressive enough to provide behavioral theory for the study of cryptographic protocols. In particular, we show that, in epi, explicit encryption and decryption primitives (handy for specifying protocols, but a burden when developing behavioral theory) are not needed because they may be fully abstractly encoded. Thus, they may be simply defined as programming constructs, what simplifies the development of the behavioral theory and of analysis tools.

To attain this aim, we study in detail the behavioral semantics of epi. We first define a contextual equivalence—barbed congruence—and look for a co-inductive congruence relation which characterizes it. To obtain such a result, we define in epi the usual notions of bisimilarities proposed for the π-calculus, and comparing them, establishing a lattice of inter-relations (similar to that of the π-calculus). We establish that, in epi, barbed congruence, the "natural" contextual equivalence, coincides with early bisimilarity. Moreover, we extend epi with nondeterministic, symmetrical, cryptographic primitives, defining the syntax and operational semantics of this new calculus. The behavioral theory also extends naturally to this setting. Following Carbone and Maffeis [CM03] we define an encoding of the new constructs for encryption and decryption of messages into the original epi. Furthermore, we prove that such an encoding is sound and complete with respect to barbed congruence. This fully abstract encoding allows to import to crypto-epi all the behavioral theory of epi. We therefore conclude that the π-calculus with polyadic synchronization (epi) is potentially expressive enough to provide behavioral theory for, to analyze and to verify, security protocols. To illustrate the use of the theory developed, we prove the correctness of a protocol of secure message exchange. This work strengthens the hypothesis that a fully abstract encoding of a crypto calculus like the Spi-calculus into epi is possible. Notice that Baldamus *et al.* already proposed an encoding of Spi into the pi-calculus, but only preserving may testing [BPV04].

Future work. We plan to study if and how epi can express properties of cryptographic protocols such as authenticity and secrecy. In particular, we shall address the following issues: (1) adapt the Mobility Workbench to work with this setting; (2) deal with other crypto primitives; (3) develop equational (axiomatic) theory; (4) test with larger examples / known protocols; (5) look for a more general encoding (the one presented is ad-hoc); and (6) study an encoding of Spi and/or of Applied Pi into epi.

Acknowledgements

This research has initially been carried out in the context of Joana Martinho's Master in Software Systems Engineering at Aalborg University, supervised by Luca Aceto and António Ravara. Special thanks to Luca for all the support and guidance on this research. We also thank Michele Boreale, Marco Carbone, Sergio Maffeis, and the anonymous referees for useful comments on some of the matters discussed herein. António Ravara was partially supported by the EU FEDER and FCT, via the Center for Logic and Computation, and the EU IST proactive initiative FET-Global Computing (project Sensoria, IST–2005–16004).

References

[AF01] Martín Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In *Proceedings of the 28th ACM Symposium on Principles of Programming Languages (POPL'01)*, pages 104–115. ACM Press, 2001.

[AG97] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 36–47. ACM Press, 1997.

[BAF07] Bruno Blanchet, Martín Abadi, and Cédric Fournet. Automated verification of selected equivalences for security protocols. *Journal of Logic and Algebraic Programming*, 2007. To appear.

[BN05] Johannes Borgström and Uwe Nestmann. On bisimulations for the spi calculus. *Mathematical Structures in Computer Science*, 15(3):487–552, 2005.

[BNP02] Michele Boreale, Rocco De Nicola, and Rosario Pugliese. Proof techniques for cryptographic processes. *SIAM Journal on Computing*, 31(3):947–986, 2002.

[BPV04] Michael Baldamus, Joachim Parrow, and Björn Victor. Spi calculus translated to π-calculus preserving may-testing. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS '04)*, pages 21–31. i3ecsp, 2004.

[CM03] Marco Carbone and Sergio Maffeis. On the expressive power of polyadic synchronization in π-calculus. *Nordic Journal of Computing*, 10(2):70–98, 2003.

[HHK95] Martin Hansen, Hans Hüttel, and Josva Kleist. Bisimulations for asynchronous mobile processes. In *Proceedings of the Tiblisi Symposium on Language, Logic, and Computation*. Research paper HCRC/RP-72, Human Communication Research Centre, University of Edinburgh, 1995.

[MPW92] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, part I/II. *Journal of Information and Computation*, 100:1–77, 1992.

[MR07] Joana Martinho and António Ravara. Encoding cryptographic primitives in a calculus with polyadic synchronisation. Technical report, Department of Mathematics, Instituto Superior Técnico, Technical University of Lisbon, Portugal, 2007. URL: www.math.ist.utl.pt/∼amar/papers/cepi-long.pdf.

[MS92] Robin Milner and Davide Sangiorgi. Barbed bisimulation. In *Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP '92)*, volume 623 of *Lecture Notes in Computer Science*, pages 685–695. Springer-Verlag, 1992.

[Qua99] Paola Quaglia. The pi-calculus: Notes on labelled semantics. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 68:104–114, 1999.

[San92] Davide Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms*. PhD thesis CST–99–93, Department of Computer Science, University of Edinburgh, U. K., 1992.

[San96] Davide Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Informatica*, 33:69–97, 1996. An extract appeared in *Proceedings of the 4th International Conference on Concurrency Theory (CONCUR '93)*, Lecture Notes in Computer Science 715, Springer-Verlag.

[San00] Davide Sangiorgi. Lazy functions and mobile processes. In Gordon Plotkin, Colin Stirling, and Mads Tofte, editors, *Proof, Language and Interaction: Essays in Honour of Robin Milner*. M. I. T. Press, 2000.

[SW01] Davide Sangiorgi and David Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.

[Vic94] Björn Victor. *A Verification Tool for the Polyadic π-Calculus*. Licentiate thesis, Department of Computer Systems, Uppsala University, Sweden, 1994. Available as report DoCS 94/50.

[VM94] Björn Victor and Faron Moller. The Mobility Workbench — a tool for the π-calculus. In *Proceedings of the 6th International Conference on Computer Aided Verification (CAV '94)*, volume 818 of *Lecture Notes in Computer Science*, pages 428–440. Springer-Verlag, 1994.