Journal of Universal Computer Science, vol. 19, no. 10 (2013), 1396-1432
submitted: 19/10/12, accepted: 27/5/13, appeared: 28/5/13

An Algebraic Theory of Epistemic Processes

Hamid Reza Mahrooghi
(Department of Computer Engineering, Sharif University of Technology
Tehran, Iran
mahrooghi@ce.sharif.edu)

Rasool Jalili
(Department of Computer Engineering, Sharif University of Technology
Tehran, Iran
jalili@sharif.edu)

Abstract: In the past few years, several process-algebraic frameworks have been proposed that incorporate the notion of epistemic knowledge. These frameworks allow for reasoning about knowledge-related properties, such as anonymity, secrecy and authentication, in the operational specifications given in process-algebraic languages. Hitherto, no sound and (ground-)complete axiomatization has been given for the abovementioned process-algebraic frameworks. In this paper, we define notions of bisimulation that are suitable for such process algebras with histories and give a sound and ground-complete axiomatization for the theory of *CryptoPAi*, which is a process algebra based on Milner's Calculus of Communicating Systems (CCS) extended with cryptographic terms and identities. Moreover, we show that one of our defined notions of bisimulation is precisely characterized by the extension of the Hennessy-Milner logic with epistemic constructs.

Key Words: Process Algebra, Epistemic Logic, Axiomatization, Security Protocols, Cryptography

Category: F.4, F.4.1, F.4.3

1 Introduction

Equational reasoning is a cornerstone of algebraic process theory and allows for reasoning on processes at the level of syntax, without generating their (huge and often infinite) state space. Several process algebras have been proposed in the past four decades [Milner 1980, Hoare 1985, Baeten et al. 2009] and many of them have been applied to academic and industrial case studies. More recently, several process theories have been equipped with sufficient syntactic and semantic constructs to allow for reasoning about epistemic properties of processes and agents, e.g., to express properties such as the anonymity (lack of knowledge) of certain actions, or the authentication (having true belief about the identity) of certain agents. Examples of such process algebras include the Agent Communication Programming Language (ACPL, a CSP-based process algebra) of [de Boer et al. 2003, Eijk et al. 2003], the Calculus of Cryptographic Communication (C^3) of [Borgström et al. 2006] and its timed extension [Borgström et al. 2006], and the

Calculus of Communicating Systems with identities (CCSi) of [Dechesne et al. 2007] and its extension with cryptographic constructs *CryptoPAi* [Mahrooghi and Mousavi 2011].

We are not aware of a (ground-)complete axiomatization for any of the above-mentioned process theories (or any other process theory extended with epistemic constructs). In this paper, we fill this gap by first defining some notions of bisimulation that are suitable for epistemic processes. Subsequently, we use one of our defined notions of bisimulation to provide a sound and complete axiomatization for the process algebra *CryptoPAi*. We further show that the chosen notion of bisimulation is precisely characterized by the extension of the modal mu-calculus with epistemic knowledge constructs.

1.1 Related Work

Axiomatizing process languages has been a challenging task and the first finite ground-complete axiomatization of a process algebras dates back to the seminal work of Bergstra and Klop in [Bergstra and Klop 1984]. Ever since, many papers have been published on axiomatization and unaxiomatizability of process algebras, of which [Baeten 2005] and [Aceto and Ingólfsdóttir 2007] provide interesting historical accounts. Our approach to axiomatizing *CryptoPAi* builds upon these earlier approaches, but is complicated by a few issues: the existence of histories recording the past events (a necessary ingredient for epistemic reasoning), makes it non-trivial to define an appropriate notion of bisimulation. To solve this issue, we rely on earlier research on bisimulation for process algebras with data [Baeten and Bergstra 1997, Mousavi et al. 2004, Mousavi et al. 2005]. Moreover, the introduction of cryptographic terms as data call for some additional work in equating atomic actions and their parameters. To this end, we adapt some ideas from earlier work on symbolic cryptography, for example, in [Abadi and Rogaway 2002].

Our axiomatization and its proofs can be readily used for a ground axiomatization of *PAi* [Dechesne et al. 2007], since *PAi* is a special case of *CryptoPAi*, where the set of parameters is taken to be empty. Also, we expect our results to be applicable to other process algebras with epistemic constructs, such as those proposed in [de Boer et al. 2003, Eijk et al. 2003, Borgström et al. 2006, Borgström et al. 2006].

1.2 Structure of the Paper

The rest of this paper is structured as follows. In Section 2, we recall the syntax and the semantics of *CryptoPAi*. In Section 3, we explore different possibilities for defining a notion of bisimulation for processes with epistemic constructs and propose a notion of bisimulation that is a congruence, i.e., compositional, for

CryptoPAi. In Sections 4 and 5, we provide a sound and ground completeness of, respectively, the sequential subset, and the full *CryptoPAi* language, for the proposed notion of bisimulation. In Section 6, we introduce the extension of the Hennessy-Milner logic with an epistemic knowledge construct, called EHML, and show that our notion of bisimilarity is the notion of equivalence characterized by the introduced modal logic. We conclude the paper in Section 7.

2 The *CryptoPAi* Calculus

The syntax of *CryptoPAi* calculus extends the syntax of Milner's Calculus of Communicating Systems (CCS) with two essential ingredients: identities and cryptographic terms. Identities are used to annotate actions in order to designate the agents that can observe the action (the agents that cannot observe a certain action, will observe its *public appearance*). Cryptographic terms are used to denote the data that is communicated in a basic action and possession of keys by agents will provide another aspect of observability (and thus, indistinguishability) to the calculus. In the remainder of this section, we first define our notion of cryptographic terms. Subsequently, we define the syntax of the *CryptoPAi* calculus and present its operational semantics.

2.1 Cryptographic Terms

Our signature for cryptographic terms features basic constructs such as plain text, nonces, symmetric keys and encryption, given in the following definition. This language can be extended, as demonstrated in [Mahrooghi and Mousavi 2011], with more constructs such as asymmetric keys, signing, and blinding. However, the addition of these constructs does not have any influence for the axiomatization presented in the remainder of this paper and hence, will not be considered henceforth.

Definition 1 Symbolic Terms. We assume a syntactic class $\mathcal{K}ey$ of *symmetric keys*, typically denoted by k, k_I,...; a syntactic class $\mathcal{I}d$ of *principal identities*, typically represented by A, B, 1, 2, ...; a class $\mathcal{N}once$ of *nonces*, denoted by n, n_I, ...; a class $\mathcal{M}sg$ of *plain-text messages*, denoted by m, m_I, ...; a class $\mathcal{R}nd$ of *formal random-numbers*, denoted by r, r_i, ... Plain text messages, keys, nonces, and random-numbers are different syntactic classes and are assumed to be disjoint.

The set of crypto-terms, denoted by $\mathcal{T}erm$, is defined by the following grammar:

$$m, m' ::= k \mid id \mid t \mid n \mid \{m\}_k^r \mid (m, m')$$

where $k \in \mathcal{K}ey$, $id \in \mathcal{I}d$, $t \in \mathcal{M}sg$, $r \in \mathcal{R}nd$, and $n \in \mathcal{N}once$. The term $\{m\}_k^r$ is the result of encrypting m with the symmetric key k and random-number r. The term (m, m') represents the pairing of m and m'.

$$(\in_{\mathbf{T}})\frac{}{T \vdash m} m \in T$$

$$(\mathbf{pair})\frac{T \vdash m \quad T \vdash m'}{T \vdash (m, m')} \quad (\mathbf{fst})\frac{T \vdash (m, m')}{T \vdash m} \quad (\mathbf{snd})\frac{T \vdash (m, m')}{T \vdash m'}$$

$$(\mathbf{dec})\frac{T \vdash \{m\}_k \quad T \vdash k}{T \vdash m} \quad (\mathbf{enc})\frac{T \vdash k \quad T \vdash m}{T \vdash \{m\}_k}$$

Figure 1: Term Deduction Rules

Since some of the cryptographic constructors (such as encryption) represent probabilistic operations, we need to use $\mathcal{R}nd$ in order to keep track of different invocations of the encryption primitive (as defined in [Abadi and Jürjens 2001] and [Laud and Corin 2003]). We reason under the assumption of perfect cryptography and assume that the results of two invocations of encryption will be different, even with the same arguments. Thus, to keep track of invocation of that operation, the elements of $\mathcal{R}nd$ are used. As the labels r are identifiers of invocations of the encryption primitive, whenever we consider two expressions $\{m\}_k^r$ and $\{m'\}_{k'}^r$ with the same label r, then it should also hold that $m = m'$ and $k = k'$.

Informally, we may just write $\{m\}_k$ instead of $\{m\}_k^r$, when r is irrelevant or clear from the context. To avoid cluttering the notation, we assume that pairing associates to the right and denote nested pairs of the form $(m_0, (m_1, , \ldots, (m_{n-1}, m_n)))$ by $(m_0, m_1, \ldots, m_{n-1}, m_n)$.

2.2 Term Deduction

Using cryptographic operations, terms can be derived (i.e., (de)constructed) from others. We have captured this concept by the deduction rules in Figure 1. A judgement of the form $T \vdash m$ reads *term m is derivable from the set T of terms.* The deduction rules are self-explanatory; it should only be noted that m is an arbitrary crypto-term in these rules, while k is a key.

Example 1. Below, for instance, we assume $\mathbf{T}$ as a set of terms that has been encountered in a run of a protocol.

$$\mathbf{T} = \boldsymbol{\{}(A, B, n_A), \{(k_{AB}, B, \{(k_{AB}, A)\}_{k_{BS}})\}_{k_{AS}}, \{(k_{AB}, A)\}_{k_{BS}}, \{n_B\}_{k_{AB}}, k_{AS}\boldsymbol{\}}.$$

The derivation depicted in Figure 2, shows that the term n_B is derivable from T, using the deduction rules of Figure 1.

$$
(\mathbf{dec})\dfrac{(\mathbf{fst})\dfrac{(\mathbf{dec})\dfrac{(\in_{\mathbf{T}})\dfrac{}{T \vdash \{(k_{AB}, B, \{k_{AB}, A\}_{k_{BS}})\}_{k_{AS}}} \quad (\in_{\mathbf{T}})\dfrac{}{T \vdash k_{AS}}}{T \vdash (k_{AB}, B, \{k_{AB}, A\}_{k_{BS}})}}{T \vdash k_{AB}} \quad (\in_{\mathbf{T}})\dfrac{}{T \vdash \{n_B\}_{k_{AB}}}}{T \vdash n_B}
$$

Figure 2: Deduction Rules for $T \vdash n_B$

2.3 *CryptoPAi* Syntax

In this section we present the syntax of *CryptoPAi*, which is a process algebraic language with value-passing of crypto-terms. Let $\mathcal{A}ct$ be a finite set of *action names* ranged over by $a, !a, ?a, b, \ldots$, and let $\mathcal{I}d$ be a finite set of *identities* typically denoted by $i, j, \ldots$. We use the exclamation and the question mark to designate send and receive actions, respectively. Actions that result from a synchronization as well as internal actions are denoted without any annotation. This is a slight deviation from Milner's CCS, where the result of a synchronization is necessarily denoted by an internal action. In our calculus, agents may be able to observe and derive useful information from synchronizations. Action $\tau \in \mathcal{A}ct$ denotes the silent action which also represents a message that offers no new information to any observer.

$P, Q ::=$	*Processes*
0	*inaction*
$D; P$	*action prefixing*
$P + Q$	*nondeterministic choice*
$P \mid\mid Q$	*parallel composition*
$\partial(P)$	*encapsulation*

$$D ::= (\mathtt{J})\alpha(\overrightarrow{M})$$

Constant 0 denotes inaction (termination). $D; P$ denotes action prefixing; an action may contain zero or more crypto-terms as parameters. Nondeterministic choice among P and Q is denoted by $P + Q$ and means that the first action taken by either of the two processes determines the choice. $P||Q$ denotes parallel composition; it allows for interleaving of internal actions and the results of earlier communications as well as hand-shaking synchronization between input and output actions. Hand-shaking synchronization results in a value-passing by replacing the variables of the receiving party with closed terms from the sending party. Our language also includes an encapsulation operator, ∂, which prevents unsuccessful communication attempts and turns them into deadlock.

$(\mathtt{J})\alpha(\overrightarrow{M}) \in D$ denotes a parametric decorated action, which has the following intuition: action $\alpha \in \mathcal{A}ct$ with parameters $\overrightarrow{M}$ is visible to principal $i \in \mathtt{J} \subseteq \mathcal{I}d$. Other principals ($j \notin \mathtt{J}$) observe $\rho(\alpha(\overrightarrow{M}))$ being taken, where $\rho : \mathcal{A}ct \times \mathcal{T}erm^* \to \mathcal{A}ct \times \mathcal{T}erm^*$ is a global renaming function, which assigns a "public appearance" to every parameterized action and should be defined by the specifier of a protocol. We assume that $\rho(\tau)$ is always defined to be τ, reflecting the fact that τ is invisible to all principals. The combination of identity decoration on actions, action renaming (public appearance), and action parameters provides different views on the behavior of protocols, according to different (participating or observing) principals.

To avoid cluttering the syntax, we assume that action prefixing binds stronger than non-deterministic choice and non-deterministic choice binds stronger than parallel composition. Also, we omit the trailing 0 and write, for example, D for $D; 0$.

2.4 *CryptoPAi* Semantics: Indistinguishability

The operational state of *CryptoPAi* is typically denoted by (p, Γ), which comprises two components: a process p and a computation Γ. The first component of the operational semantics determines the possible future behavior of the specified system, while the second component records the history of its past behavior, i.e., which processes executed which decorated actions leading to the current state. The process p has the syntax described in the previous section. The computation Γ, formally defined below, is a sequence of pairs of processes and decorated actions.

Definition 2 Computation. A computation $\Gamma \in \mathcal{C}omp$ is a sequence of pairs of the form (p, d), where p is a process and d is a parameterized decorated action with term parameter (in the syntax given in Section 2.3). We denote concatenating a pair (p, d) with a computation Γ by $(p, d) \frown \Gamma$. The empty computation is denoted by ε.

The operational semantics defines two types of relations among states: a transition relation, defining how a state may evolve by performing actions, and an indistinguishability relation (labeled by principal identities), defining all states that are deemed possible, given the current computation observed by each principal. This combination, called an Epistemic Labeled Transition System (ELTS). In order to define the indistinguishability, we need a number of auxiliary definitions, which are presented in the remainder of this section.

In order to reason about the behavior of processes, we need to attach computations to them. However, not all computations are consistent with a particular process, in that the computation in the past may not lead to the process at hand.

Hence, it is necessary to make sure that the past (computation) is consistent with the present (process), as defined below.

Definition 3 Consistency of Computations and Processes. Let $(p, \mathit{\Gamma})$ be an operation state where $p \in \mathcal{P}roc$ and $\mathit{\Gamma} \in \mathcal{C}omp$. Assume that the computation $\mathit{\Gamma}$, which is a finite computation recording the history of the process executed so far, is of the form $(p_0, d_0), \ldots, (p_n, d_n)$. $\mathit{\Gamma}$ is consistent with p when: $(p_i, (p_0, d_0), \ldots, (p_{i-1}, d_{i-1})) \xrightarrow{d_i} (p_{i+1}, (p_0, d_0), \ldots, (p_i, d_i))$ for each $i \leq n$ and moreover, $p_{n+1} = p$.

Two computations $\mathit{\Sigma}$ and $\mathit{\Sigma}'$ are called consistent, if the last process of $\mathit{\Sigma}$ and the first process of $\mathit{\Sigma}'$ are identical.

For example, $((\mathtt{J})\alpha(\overrightarrow{M}) + (\mathtt{J})\beta(\overrightarrow{M'}), \epsilon)$ is consistent with both $(\mathtt{J})\alpha(\overrightarrow{M})$ and $(\mathtt{J})\beta(\overrightarrow{M'})$. Also, $((\mathtt{J})\alpha(\overrightarrow{M}); (\mathtt{J})\alpha(\overrightarrow{M}), \epsilon)$ is consistent with $(\mathtt{J})\alpha(\overrightarrow{M})$, but not with $(\mathtt{J})\beta(\overrightarrow{M'})$.

Definition 4 Patterns. A pattern is a crypto-term with (possibly multiple) occurrences of a specific symbol $\Box^r$ (indexed by random-numbers), capturing those encrypted messages that cannot be decrypted. Hence, the set $\mathcal{P}Term$ of patterns is defined with the following grammar.

$$m_P, m'_P ::= k \mid id \mid t \mid n \mid \{m_P\}_k^r \mid (m_P, m'_P) \mid \Box^r$$

The below-given function $Pattern : \mathcal{T}erm \rightarrow \mathcal{P}Term$ defines the pattern of a message given a set of keys and is defined using the auxiliary function $pat : \mathcal{T}erm \times \mathcal{K}ey \rightarrow \mathcal{P}Term$.

$$Pattern(M) = pat(M, \{k \in \mathcal{K}ey \mid M \vdash k\}).$$

Intuitively, the pattern of a term is what can be learned and understood from the term from the viewpoint of a principal having a number of keys at its disposal.

$$\begin{array}{lll}
pat(k, K) & = k & (k \in \mathcal{K}ey), \\
pat(id, K) & = id & (id \in \mathcal{I}d), \\
pat(t, K) & = t & (t \in \mathcal{M}sg), \\
pat(n, K) & = n & (n \in \mathcal{N}once), \\
pat((m, m'), K) & = (pat(m, K), pat(m', K)), &
\end{array}$$

$$pat(\{m\}_k^r, K) = \begin{cases} \{pat(m, K)\}_k^r & if\ k \in K, \\ \Box^r & \text{otherwise.} \end{cases}$$

We re-use the same notation and write $pat(m)$ for $pat(m, K)$, where K is the set of all keys k such that $\{m\} \vdash k$.

Example 2. The following examples illustrate the notion of pattern:

$$\begin{array}{lll} pat(\{\{(k_{AB},A)\}^{r}_{k_{BS}}\}^{r'}_{k_{AS}},\emptyset) & = & \Box^{r'} \\ pat(\{\{(k_{AB},A)\}^{r}_{k_{BS}}\}^{r'}_{k_{AS}},\{k_{AS}\}) & = & \{\Box^{r}\}^{r'}_{k_{AS}} \qquad \text{if } k_{AS} \neq k_{BS} \\ pat(\{\{(k_{AB},A)\}^{r}_{k_{BS}}\}^{r'}_{k_{AS}},\{k_{AS},k_{BS}\}) & = & \{\{(k_{AB},A)\}^{r}_{k_{BS}}\}^{r'}_{k_{AS}} \\ pat(\{\{(k_{AB},A)\}^{r}_{k_{BS}}\}^{r'}_{k_{AS}}) & = & \Box^{r'} \\ pat((\{A\}^{r}_{k},\{A\}^{r'}_{k})) & = & (\Box^{r},\Box^{r'}) \end{array}$$

Using the notion of patterns and incorporating the randomness of keys, nonces and random-numbers, we define the following notion of pattern equivalence, denoted by $\equiv_p$.

Definition 5 Pattern Equivalence. Two terms m and $m' \in \mathcal{T}erm$ are pattern equivalent, denoted by $m \equiv_p m'$, if and only if there exists a type preserving bijection σ of $\mathcal{K}ey \cup \mathcal{N}once \cup \mathcal{R}nd$, such that $pat(m) = pat(m'\sigma)$, where $m'\sigma$ denotes the application of σ to m'.

Example 3. The following examples illustrate the notion of pattern equivalence:

$$\begin{array}{llll} k & \equiv_p & k' & \\ (k_0,k_0) & \not\equiv_p & (k_0,k_1) & \text{if } k_0 \neq k_1 \\ \{A\}^{r}_{k} & \equiv_p & \{B\}^{r'}_{k'} & \\ (\{A\}^{r}_{k},k) & \not\equiv_p & (\{B\}^{r'}_{k'},k') & \text{if } A \neq B \\ (\{A\}^{r}_{k},\{A\}^{r}_{k}) & \not\equiv_p & (\{A\}^{r}_{k},\{A\}^{r'}_{k}) & \text{if } r \neq r' \end{array}$$

For instance, we have $A^{r}_{k} \equiv_p B^{r'}_{k'}$. This is because the keys k and k' are not known and hence, the observer will see an obscure bit-stream that cannot be deciphered in either case. Consequently, $\{A\}^{r}_{k}$ and $\{B\}^{r'}_{k'}$ are pattern equivalent, because there exists a type-preserving bijection σ of $\mathcal{K}ey \cup \mathcal{R}nd$ which maps k' to k, r' to r and hence, it holds that $pat(\{A\}^{r}_{k}) = pat(\{B\}^{r'}_{k'}\sigma) = \Box^{r}$.

Example 4. Assume two terms M and N as follows.

$$\begin{array}{ll} M = ((\{0\}^{r_1}_{k_1},\{1\}^{r_2}_{k_2}),(\{1\}^{r_3}_{k_3},k_4)), & pat(M) = ((\Box^{r_1},\Box^{r_2}),(\Box^{r_3},k_4)) \\ N = ((\{1\}^{r_5}_{k_5},\{0\}^{r_6}_{k_6}),(\{0\}^{r_7}_{k_7},k_8)), & pat(N) = ((\Box^{r_5},\Box^{r_6}),(\Box^{r_7},k_8)) \end{array}$$

Then, if we replace $k_5 \to k_1$, $k_6 \to k_2$, $k_7 \to k_3$, $k_8 \to k_4$, $r_5 \to r_1$, $r_6 \to r_2$, and $r_7 \to r_3$ in N, the pattern of N turns into the pattern of M. i.e., the two terms M and N are pattern equivalent, denoted by $M \equiv_p N$, because there exists a type preserving bijection σ of $\mathcal{K}ey \cup \mathcal{R}nd$ such that maps k_5 to k_1, k_6 to k_2 etc and consequently, we have $pat(M) = pat(N\sigma) = ((\Box^{r_1},\Box^{r_2}),(\Box^{r_3},k_4))$.

The following lemma illustrates that the information that is abstracted away by the notion of pattern, can always be recovered by putting terms into context.

Lemma 6. *Consider two terms m and m' such that $m \equiv_p m'$; if it holds that for each term m'', $(m'', m) \equiv_p (m'', m')$, then m and m' are syntactically equal.*

Proof. By induction on the size of m plus the size of m'. We make a case distinction based on the structure of m:

- Assume that m is a plain text message or an identifier; then it follows from Definition 4 that m' has to be of the same form and also syntactically equal to m; hence, the lemma follows trivially.
- Assume that m is a nonce and take m'' to be the same nonce as m; then, it should hold that $(m, m) \equiv_p (m, m')$. Then, there exists a bijection σ such that $pat((m, m)) = pat((m, m')\sigma)$, or, following the definition of pat, (m, m) has to be syntactically identical to $(m\sigma, m'\sigma)$; however, since σ is a bijection, this can only hold if m and m' are syntactically equivalent.
- Assume that m' is a key; following a similar reasoning as above (by taking m'' to be the same key as m), it follows that m' should also be the same key.
- Assume that m is a pair of the form (m_0, m_1); then it follows from Definition 4 that m' should also be a pair of the form (m'_0, m'_1) such that $pat(m_0) = pat(m'_0\sigma)$ and $pat(m_1) = pat(m'_1\sigma)$. It then follows from the induction hypothesis that m_0 should be syntactically equal to m'_0 and m_1 has to be syntactically equal to m'_1. Hence, m and m' are syntactically equal.
- Assume that m is an encrypted message of the form $\{m_0\}^r_k$; then m' should also be of the form $\{m'_0\}^{r'}_{k'}$. Take m'' to be $(k, \{m_0\}^r_k, k', \{m'_0\}^{r'}_{k'})$ and it follows from the induction hypothesis, the definition of pat, and Definition 4 that m_0 and m'_0 are syntactically identical and so are, respectively, r and r', and k and k'. □

A straightforward generalization of Lemma 6 concerns its lifting to sequences of terms; in fact, we use this generalization in some of our proofs in the remainder of this paper.

Lemma 7. *$\equiv_p$ is an equivalence relation.*

Proof. Equality on patterns is an equivalence relation and hence $\equiv_p$ is an equivalence relation. □

In order to lift the notion of pattern equivalence to computations, we need the notions of appearance for actions and visible terms, defined below.

Definition 8 Appearance of Actions. Given a decorated action d, an identity i, and a fixed global renaming function ρ, the appearance of d to i, denoted by $Appear^i(d)$, is defined as follows.

$$Appear^i((J)a(\overrightarrow{M})) = \begin{cases} a(\overrightarrow{M}) & \text{if } i \in J, \\ \\ b(\overrightarrow{M'}) & \text{if } i \notin J \text{ and } \rho(a(\overrightarrow{M})) = b(\overrightarrow{M'}) \end{cases}$$

In the above definition it is assumed that $\overrightarrow{M}$ and $\overrightarrow{M'}$ can be empty vectors of terms, denoting actions with no parameter (including τ).

Definition 9 Visible Terms. Given a computation Γ and an identity i, the set of visible terms in Γ according to i is denoted by $TermSet^i(\Gamma)$, and is defined below (ε and $\emptyset$ denote an empty computation and empty set, respectively).

$$TermSet^i(\Gamma) = \begin{cases} \emptyset & \text{if } \Gamma = \varepsilon, \\ \\ \{\overrightarrow{M}\} \cup TermSet^i(\Gamma') & \text{if } \Gamma = (p,d)^\frown\Gamma' \text{ and} \\ & \quad Appear^i(d) = a(\overrightarrow{M}), \\ \\ TermSet^i(\Gamma') & \text{if } \Gamma = (p,d)^\frown\Gamma' \text{ and} \\ & \quad Appear^i(d) = a(). \end{cases}$$

Definition 10 Indistinguishability of Computations. Let Γ be a computation, then the pattern of computation Γ from the viewpoint of the principal i is denoted by $CPattern^i(\Gamma)$, as defined below.

$$CPattern^i(\Gamma) = \begin{cases} \varepsilon & \text{if } \Gamma = \varepsilon, \\ \\ CPattern^i(\Gamma') & \text{if } \Gamma = (p,d)^\frown\Gamma' \text{ and} \\ & \quad Appear^i(d) = \tau, \\ \\ a(pat(M,T^i))^\frown CPattern^i(\Gamma') & \text{if } \Gamma = (p,d)^\frown\Gamma' \text{ and} \\ & \quad Appear^i(d) = a(M), \\ \\ a()^\frown\ CPattern^i(\Gamma') & \text{if } \Gamma = (p,d)^\frown\Gamma' \text{ and} \\ & \quad Appear^i(d) = a() \text{ and } a \neq \tau, \end{cases}$$

where $T^i = \{t | t \in \mathcal{T}erm \ \wedge \ TermSet^i(\Gamma) \vdash \ t\}$.

$$(\mathbf{0})\frac{}{(0,\Gamma)\checkmark} \qquad (\mathbf{d})\frac{}{(d;p,\Gamma)\stackrel{d}{\Rightarrow}(p,\Gamma^\frown(d;p,d))}$$

$$(\mathbf{n0})\frac{(p_0,\Gamma)\stackrel{d}{\Rightarrow}(q_0,\Gamma^\frown(p_0,d))}{(p_0+p_1,\Gamma)\stackrel{d}{\Rightarrow}(q_0,\Gamma^\frown(p_0+p_1,d))} \qquad (\mathbf{n2})\frac{(p_0,\Gamma)\checkmark}{(p_0+p_1,\Gamma')\checkmark}$$

$$(\mathbf{p0})\frac{(p_0,\Gamma)\stackrel{d}{\Rightarrow}(q_0,\Gamma^\frown(p_0,d))}{(p_0\,||\,p_1,\Gamma)\stackrel{d}{\Rightarrow}(q_0\,||\,p_1,\Gamma^\frown(p_0\,||\,p_1,d))} \qquad (\mathbf{p2})\frac{(p_0,\Gamma)\checkmark \quad (p_1,\Gamma')\checkmark}{(p_0\,||\,p_1,\Gamma'')\checkmark}$$

$$(\mathbf{p3})\frac{(p_0,\Gamma)\stackrel{(\mathtt{J})?a(\overrightarrow{M})}{\Rightarrow}(q_0,\Gamma') \quad (p_1,\Gamma)\stackrel{(\mathtt{J}')!a(\overrightarrow{M})}{\Rightarrow}(q_1,\Gamma'')}{(p_0\,||\,p_1,\Gamma)\stackrel{(\mathtt{J}\cup\mathtt{J}')a(\overrightarrow{M})}{\Rightarrow} q_0\,||\,q_1,\Gamma^\frown(p_0\,||\,p_1,(\mathtt{J}\cup\mathtt{J}')a(\overrightarrow{M})))}$$

$$(\mathbf{encap})\frac{(p_0,\Gamma)\stackrel{d}{\Rightarrow}(q_0,\Gamma^\frown(p_0,d))}{(\partial(p_0),\Gamma)\stackrel{d}{\Rightarrow}(\partial(q_0),\Gamma^\frown(\partial(p_0),d))}$$

$$(\mathbf{strip})\frac{(p,\Gamma)\stackrel{(\mathtt{J})a(\overrightarrow{M})}{\Rightarrow}(q,\Gamma')}{(p,\Gamma)\stackrel{a(\overrightarrow{M})}{\rightarrow}(q,\Gamma')} \qquad (\mathbf{I_C})\frac{\Gamma_0\stackrel{i}{=}\Gamma_1}{(p_0,\Gamma_0)\stackrel{i}{\cdots}(p_1,\Gamma_1)}$$

Figure 3: SOS Rules of *CryptoPAi*

Next, we define the indistinguishability relation, denoted by $\stackrel{i}{=}$, of computations as follows.

$$\Gamma \stackrel{i}{=} \Gamma' \qquad \textit{iff} \qquad CPattern^i(\Gamma) = CPattern^i(\Gamma'\sigma),$$
$$\text{for a type-preserving bijection } \sigma \text{ on } \mathcal{K}ey \cup \mathcal{N}once \cup \mathcal{R}nd.$$

Two computations are called indistinguishable when they are indistinguishable with respect to every identity.

2.5 *CryptoPAi* Semantics: Deduction Rules

Plotkin-style deduction rules for the structural operational semantics [Plotkin 2004] of *CryptoPAi* are given in Figure 3.

The transition relation $\rightarrow$ has exactly the same role and meaning as in the standard notion of Labeled Transition system (LTS). The formula $s\checkmark$ shows the possibility of termination in state s. The indistinguishability between s_0 and s_1 according to principal i is denoted by expression $s_0 \stackrel{i}{\cdots} s_1$. Observing of actions depends on the visibility range of actions. The relation $\stackrel{d}{\Rightarrow} \subseteq St \times St$ has been defined for each parametric decorated action $d \in D$ where St is the set of operational states. The deduction rules for $\overset{d}{\Rightarrow}$ are mostly self-explanatory and standard to most process algebras. In the deduction rule (**strip**), the extra information on the labels (concerning the visibility range) are stripped off, i.e., we block individual send and receive actions and thereby obtain the transition relation $\rightarrow$. The deduction rule ($\mathbf{I_C}$) moves the indistinguishability relation of computations up to operational states. Γ_0 and Γ_1 are consistent with p_0 and p_1, respectively. Because $\overset{i}{\cdots}$ only concerns histories and not processes, it specifies when two past histories are indistinguishable, and does not make any statement about their possible future developments. The symmetric rules (**n1**), (**n3**), (**p1**), and (**p4**) are omitted for brevity. Termination of a process is orthogonal to its past history, so different meta-variables are used for the histories in the premises and the conclusion of rules (**n2**) and (**p2**). The transition relation $\Rightarrow$ and indistinguishability equivalence relation $\cdots$ are the sets of all closed statements provable using the deduction rules (plus their symmetric versions) from Figure 3. These define an Epistemic Labeled Transition System (ELTS) for each process p, as defined below.

Definition 11 Semantics of Processes. Given the sets $\mathcal{A}ct$ and $\mathcal{T}erm$, an ELTS is a 5-tuple $\langle St, \rightarrow, \checkmark, I_C, s_0\rangle$, where $\rightarrow \subseteq St \times \mathcal{A}ct \times \mathcal{T}erm^* \times St$ is the transition relation, $\checkmark \subseteq St$ is the termination predicate, $I_C \subseteq St \times \mathcal{I}d \times St$ is the indistinguishability relation, and s_0 is the initial state.

The semantics of process p has been defined by the ELTS with pairs of processes and computation as the states, $\rightarrow$ as the transition relation, $\checkmark$ as the termination relation, $\cdots$ as the indistinguishability equivalence relation, and (p, ε) as the initial state, where ε denotes the empty computation.

An operational state is a pair (p, Γ), where $p \in \mathcal{P}roc$ is a *CryptoPAi* process and Γ is a finite computation recording the history of the process executed so far.

3 Notions of Bisimulation for Epistemic Processes

We aim to provide an equational theory for our *CryptoPAi* calculus to equationally reason about security protocols. First, we need to present a notion of our bisimilarity relation between processes on the ELTS. We define this notion based on the notion of strong bisimilarity.

3.1 Notions of Bisimilarity

We need to adapt strong bisimilarity to cater for the computations attached to processes in our operational semantics. Our approach is inspired by [Mousavi et al. 2004], but makes some modifications in order to support the issues regarding patterns, consistency of computations, and applying substitutions. The following notion of bisimilarity is the first attempt to enrich strong bisimilarity with computations.

Definition 12 State-based Pattern Bisimulation Relation. A binary symmetric relation $R^{\sigma}_{sb} \subseteq (\mathcal{P}roc \times \mathcal{C}omp)^2$ over the set of states of an ELTS is a state-based pattern bisimulation if and only if whenever $(p, \Gamma_p)\ R^{\sigma}_{sb}\ (q, \Gamma_q)$ then Γ_p and Γ_q are, respectively, consistent with p and q and moreover, the following statements hold:

- $\Gamma_p \overset{i}{=} \Gamma_q\sigma$, for each $i \in \mathcal{I}d$,
- if $(p, \Gamma_p) \xrightarrow{d} (p', \Gamma_{p'})$ then there exists a transition $(q, \Gamma_q) \xrightarrow{d'} (q', \Gamma_{q'})$ such that $(p', \Gamma_{p'})\ R^{\sigma}_{sb}\ (q', \Gamma_{q'})$,
- if $(p, \Gamma_p)\checkmark$ then $(q, \Gamma_q)\checkmark$, and
- if $(p, \Gamma_p) \overset{i}{\cdots} (p', \Gamma_{p'})$ then there exists a relation $(q, \Gamma_q) \overset{i}{\cdots} (q', \Gamma_{q'})$ such that $(p', \Gamma_{p'})\ R^{\sigma}_{sb}\ (q', \Gamma_{q'})$.

Two states (p, Γ_p) and (q, Γ_q) are called state-based pattern bisimilar, denoted by $(p, \Gamma_p) \sim_{sb} (q, \Gamma_q)$, *iff* there exits R^{σ}_{sb} s.t. $(p, \Gamma_p)\ R^{\sigma}_{sb}\ (q, \Gamma_q)$.

Note that in the second condition in Definition 12, the target computations ($\Gamma_{p'}$ and $\Gamma_{q'}$) have to be indistinguishable due to the first condition in the same definition and hence, the labels d and d' are implicitly related.

Theorem 13. *The state-based pattern bisimularity is an equivalence relation.*

Proof. Equivalence of the state-based pattern bisimularity follows trivially from the equivalence property of the state-based equivalence bisimilarity [Mousavi et al. 2004] and that of our indistinguishability equivalence relation. □

Our state-based pattern bisimilarity relates pairs of processes and computations and as such does not allow for any interference from the context and hence is not robust (compositional) with respect to process composition. In other words, if the common initial history state is not known (e.g., if the components have to start their execution on the result of an unknown or non-deterministic process), then this notion of bisimilarity is not useful. The example below, illustrates this fact, which is also depicted in Figure 4. Note that for simplicity, we have left out the principal identities which do not play a role in our example.

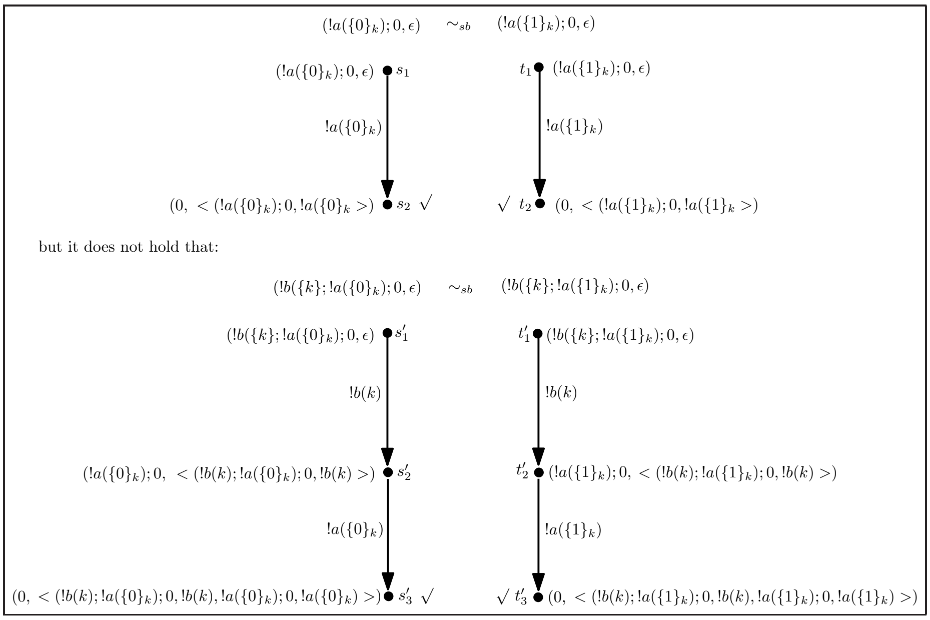

Figure 4: Lack of compositionality for state-based pattern bisimilarity.

Example 5. Let $s = (!a(\{0\}_k); 0, \epsilon)$ and $t = (!a(\{1\}_k); 0, \epsilon) \in St$. We have $s \sim_{sb} t$, but it does not hold that $(!b(k); a(\{0\}_k); 0, \epsilon) \sim_{sb} (!b(k); a(\{1\}_k); 0, \epsilon)$. This is due to the computation of the corresponding operational-states are indistinguishable in the first case while it is violated in the second one. As shown in Figure 4, according to the form of states $(p_{st}, \mathit{\Gamma}_{st})$: the process part and the computation part, the following relations can be extracted for each $i \in \mathcal{I}d$:

$$CPattern^i(\mathit{\Gamma}_{s_1}) = CPattern^i(\mathit{\Gamma}_{t_1}) = \varepsilon,$$
$$CPattern^i(\mathit{\Gamma}_{s_2}) = CPattern^i(\mathit{\Gamma}_{t_2}) = < a(\Box) >,$$

but,

$$CPattern^i(\mathit{\Gamma}_{s'_1}) = CPattern^i(\mathit{\Gamma}_{t'_1}) = \varepsilon,$$
$$CPattern^i(\mathit{\Gamma}_{s'_2}) = CPattern^i(\mathit{\Gamma}_{t'_2}) = < b(k) >,$$
$$CPattern^i(\mathit{\Gamma}_{s'_3}) = < b(k), a(\{0\}_k) > \neq \ CPattern^i(\mathit{\Gamma}_{t'_3}) = < b(k), a(\{1\}_k) >,$$

and so there is no bijection σ to hold that $CPattern^i(\mathit{\Gamma}_{s'_3}) = CPattern^i(\mathit{\Gamma}_{t'_3}\sigma)$.

To overcome this problem, we define below the notion of initially stateless pattern bisimilarity.

Definition 14 Initially Stateless Pattern Bisimulation. Let $p, q \in \mathcal{P}roc$; $\Gamma_p, \Gamma_q \in \mathcal{C}omp$; and σ be a bijection. We define p and q are initially stateless pattern bisimilar, denoted by $p \sim_{isl} q$, if and only if there exists a state-based pattern bisimulation relation R^σ_{sb} s.t. $(p, \Gamma_p)\ R^\sigma_{sb}\ (q, \Gamma_q)$ for each two indistinguishable computations Γ_p and Γ_q that are consistent with p and q, respectively.

Theorem 15. *The initially stateless pattern bisimularity is an equivalence relation.*

Proof. Equivalence of the initially stateless pattern bisimularity follows trivially from the equivalence property of the initially stateless equivalence bisimilarity [Mousavi et al. 2004] and that of our indistinguishability relation. □

Congruence, which is an important property of equivalence relations in process algebra, requires an equivalence relation to be closed under composition using algebraic operators. For analyzing protocols, a congruence behavioral equivalence comes very handy since it allows for breaking the verification task into smaller parts and concluding the result from the results of the sub-problems.

Theorem 16. *The initially stateless pattern bisimularity is a congruence with respect to the sequential subset of CryptoPAi operators.*

Proof. Assume that $p, p', q, q' \in \mathcal{P}roc$, $p \sim_{isl} p'$, and $q \sim_{isl} q'$. By definition, this means there exist state-based pattern bisimulations $R^{\sigma_1}_{sb}$ and $R^{\sigma_2}_{sb}$ such that $((p, \Gamma), (p', \Gamma')) \in R^{\sigma_1}_{sb}$ and $((q, \Gamma), (q', \Gamma')) \in R^{\sigma_2}_{sb}$. Before we proceed with the rest of the proof, we unify the two substitutions by means of the following lemma.

Lemma 17. *Given two terms p and p', if there exists a state-based pattern bisimulation relation R^σ_{sb} such that $((p, \Gamma), (p', \Gamma')) \in R^\sigma_{sb}$ for each two indistinguishable computations Γ and Γ' (consistent with p and p', respectively), then, R^{id}_{sb} is also a state-based pattern bisimulation relation where id is the identity function on nonces, keys, and random numbers.*

Proof. We show that R^σ_{sb} is a state-based pattern bisimulation relation under substitution id. Take a pair $((p, \Gamma), (q, \Gamma'))$ in R^σ_{sb}. Transition $(p, \Gamma) \xrightarrow{d} (p', \Gamma^\frown d)$ can be mimicked by q using a transition of the form $(q, \Gamma') \xrightarrow{d'} (q', \Gamma'^\frown d')$, such that $\Gamma^\frown d$ and $\Gamma'^\frown d$ are indistinguishable. Since Γ and Γ' can be extended with arbitrary prefixes, a generalization of the proof of Lemma 6 implies that for each identity i, the sequence of visible terms of $\Gamma^\frown d$ and $\Gamma'^\frown d'$ should be syntactically identical. Hence, the identity function can be used to relate these two sequences, which concludes the proof. □

Thus, henceforth we shall use R_{sb} instead of $R^{\sigma_i}_{sb}$. To prove congruence, we prove the following items:

(1) $d;p \sim_{isl} d';p'$, where $Appear^i(d) = Appear^i(d')$ for each identity i. This is a stronger claim than congruence, i.e., by taking $d = d'$, congruence follows,

(2) $p + q \sim_{isl} p' + q'$, and

(3) $\partial(p) \sim_{isl} \partial(p')$.

Let R_p be the minimum bisimulation relation relating p and p', R_q be the one relating q and q', I_d be the identity relation, and $\Gamma, \Gamma' \in \mathcal{C}omp$ be indistinguishable computations s.t. $\Gamma \stackrel{i}{=} \Gamma'$ for all $i \in \mathcal{I}d$.

1. We construct the relation $R \triangleq I_d \cup R_p \cup \{((d;p,\Gamma),(d';p',\Gamma')) \mid ((p,\Gamma),(p',\Gamma')) \in R_p \wedge\ (\forall_{i\in\mathcal{I}d}\ (\Gamma \stackrel{i}{=} \Gamma'\ \wedge\ \ Appear^i(d) = Appear^i(d')))\}$. Then;

- We have the transition $(d;p,\Gamma) \stackrel{d}{\Rightarrow} (u,\Gamma_u)$ due to the rule (**d**) where $u = p$ and $\Gamma_u = \Gamma^\frown(d;p,d)$. By applying the same deduction rule, we obtain that $(d';p',\Gamma') \stackrel{d'}{\Rightarrow} (u',\Gamma'_u)$ where $u' = p'$ and $\Gamma'_u = \Gamma'\ ^\frown(d';p',d')$. But since for all identity i, $\Gamma \stackrel{i}{=} \Gamma'$ and $Appear^i(d) = Appear^i(d')$, we have $\Gamma_u \stackrel{i}{=} \Gamma'_u$. It follows that $((u,\Gamma_u),(u',\Gamma'_u)) \in R$ and hence, this condition is satisfied.

- If $(d;p,\Gamma) \stackrel{i}{\cdots} (w,\Gamma_w)$, for some $i \in \mathcal{I}d$, this is only derivable from the rule ($\mathbf{I_C}$) and hence, it holds that $\Gamma \stackrel{i}{=} \Gamma_w$. Since Γ and Γ' are indistinguishable and indistinguishability is an equivalence relation, we have $\Gamma' \stackrel{i}{=} \Gamma_w$ that follows $(d';p',\Gamma') \stackrel{i}{\cdots} (w,\Gamma_w)$, for all identity i. The fact that $(w,w) \in I_d$ proves that this condition is also satisfied.

2. We construct the relation $R \triangleq I_d \cup R_p \cup R_q \cup \{((p+q,\Gamma),(p'+q',\Gamma'))\ \mid\ ((p,\Gamma),(p',\Gamma')) \in R_p \wedge\ ((q,\Gamma),(q',\Gamma')) \in R_q \wedge\ (\forall_{i\in\mathcal{I}d}\ \Gamma \stackrel{i}{=} \Gamma')\}$. The pairs of process terms in R that are elements of I_d, R_p, or R_q obviously satisfy the transfer conditions. Thus, it remains to prove that the pairs of process terms $((p+q,\Gamma),(p'+q',\Gamma'))$ satisfy the transfer conditions. Then;

- If $(p+q,\Gamma) \stackrel{d}{\Rightarrow} (u,\Gamma_u)$, it is due to the rule (**n0**) where $u = p_1$ and $\Gamma_u = \Gamma^\frown(p+q,d)$. Since $p \sim_{isl} p'$, we have $(p',\Gamma') \stackrel{d'}{\Rightarrow} (p'_1,\Gamma'_p)$ where $\Gamma'_p = \Gamma'\ ^\frown(p',d')$, $\Gamma \stackrel{i}{=} \Gamma'$, and $\Gamma_p \stackrel{i}{=} \Gamma'_p$, for each $i \in \mathcal{I}d$. The application of the same rule (**n0**) concludes that $(p'+q',\Gamma') \stackrel{d'}{\Rightarrow} (u',\Gamma'_u)$ s.t. $u' = p'_1$ and $\Gamma'_u = \Gamma'\ ^\frown(p'+q',d')$. Because $\Gamma_u \stackrel{i}{=} \Gamma'_u$ and $((u,\Gamma_u),(u',\Gamma'_u)) \in R$, this condition is satisfied.

- If $(p+q,\Gamma) \stackrel{i}{\cdots} (w,\Gamma_w)$, for some $i \in \mathcal{I}d$, this case is identical to indistinguishability for action prefixing and hence, it is not repeated.

- If $(p{+}q, \Gamma)\checkmark$, this transition is due to the rule (**n2**) that follows $(p, \Gamma_p)\checkmark$ (or $(q, \Gamma_q)\checkmark$). Since $p \sim_{isl} p'$ (and $q \sim_{isl} q'$), it implies that $(p', \Gamma'_p)\checkmark$ (or $(q', \Gamma'_q)\checkmark$). By applying the same deduction rule, we obtain $(p'{+}q', \Gamma')\checkmark$.

3. We construct the relation $R \triangleq I_d \cup \{((\partial(p), \Gamma), (\partial(p'), \Gamma')) \mid ((p, \Gamma), (p', \Gamma')) \in R_p \wedge (\forall_{i \in \mathcal{I}d}\ \Gamma \stackrel{i}{=} \Gamma')\}$. Then;

- If $(\partial(p), \Gamma) \stackrel{(\mathtt{J})a(\overrightarrow{M})}{\Rightarrow} (u, \Gamma_u)$, it is due to the rule (**encap**) where $u = \partial(p_1)$ and $\Gamma_u = \Gamma \frown (\partial(p), (\mathtt{J})a(\overrightarrow{M}))$. It implies $(p, \Gamma) \stackrel{(\mathtt{J})a(\overrightarrow{M})}{\Rightarrow} (p_1, \Gamma_p)$ where $\Gamma_p = \Gamma \frown (p, (\mathtt{J})a(\overrightarrow{M}))$. Since $p \sim_{isl} p'$, we have $(p', \Gamma') \stackrel{(\mathtt{J})a(\overrightarrow{M'})}{\Rightarrow} (p'_1, \Gamma'_p)$ where $\Gamma'_p = \Gamma' \frown (p', (\mathtt{J})a(\overrightarrow{M'}))$, $\overrightarrow{M} \equiv_p \overrightarrow{M'}$, $\Gamma \stackrel{i}{=} \Gamma'$, and $\Gamma_p \stackrel{i}{=} \Gamma'_p$, for all $i \in \mathcal{I}d$. Applying the rule (**encap**) concludes that $(\partial(p'), \Gamma') \stackrel{(\mathtt{J})a(\overrightarrow{M'})}{\Rightarrow} (u', \Gamma'_u)$ where $u' = \partial(p'_1)$ and $\Gamma'_u = \Gamma' \frown (\partial(p'), (\mathtt{J})a(\overrightarrow{M'}))$. Because $\Gamma_u \stackrel{i}{=} \Gamma'_u$ and $((u, \Gamma_u), (u', \Gamma'_u)) \in R$, this condition is satisfied.

- If $(\partial(p), \Gamma) \stackrel{i}{\cdots} (w, \Gamma_w)$, for some $i \in \mathcal{I}d$, it is due to using the rule $(\mathbf{I_C})$ that implies $(p, \Gamma) \stackrel{i}{\cdots} (w, \Gamma_w)$ where $\Gamma \stackrel{i}{=} \Gamma_w$. But since $p \sim_{isl} p'$, then there exits a state (w', Γ'_w) s.t. $(p', \Gamma') \stackrel{i}{\cdots} (w', \Gamma'_w)$ where $\Gamma \stackrel{i}{=} \Gamma'$ and $\Gamma' \stackrel{i}{=} \Gamma'_w$, for all $i \in \mathcal{I}d$. It implies that $(\partial(p'), \Gamma') \stackrel{i}{\cdots} (w', \Gamma'_w)$ for all $i \in \mathcal{I}d$. The facts that $\Gamma_w \stackrel{i}{=} \Gamma'_w$ and $((w, \Gamma_w), (w', \Gamma'_w)) \in R$ prove the satisfaction of this condition.

□

However, putting two initially stateless pattern bisimilar processes in a parallel context may break their bisimilarity, as the intermediate computations may be modified by the context. In the following example, this issue is illustrated.

Example 6. Let $p = !b(\{0\}_k); 0$ and $q = !b(\{1\}_k); 0 \in St$. According to Theorem 16, we have $?c(x_k); p \ \sim_{isl}\ ?c(x_k); q$. As depicted in Figure 5, it does not hold that $!c(k) \,||\, ?c(x_k); p \ \sim_{isl}\ !c(k) \,||\, ?c(x_k); q$. For example, for each $i \in \mathcal{I}d$, we have:

$$CPattern^i(\Gamma_{s_1}) = CPattern^i(\Gamma_{t_1}) = \varepsilon,$$
$$CPattern^i(\Gamma_{s_2}) = CPattern^i(\Gamma_{t_2}) =< c(x_k) >,$$
$$CPattern^i(\Gamma_{s_3}) = CPattern^i(\Gamma_{t_3}) =< c(x_k), b(\Box) >,$$

but,

$$CPattern^i(\Gamma_{s'_1}) = CPattern^i(\Gamma_{t'_1}) = \varepsilon,$$
$$CPattern^i(\Gamma_{s'_2}) = CPattern^i(\Gamma_{t'_2}) =< c(k) >,$$
$$CPattern^i(\Gamma_{s'_3}) =< c(k), b(\{0\}_k) > \neq\ CPattern^i(\Gamma_{t'_3}) =< c(k), b(\{1\}_k) >,$$

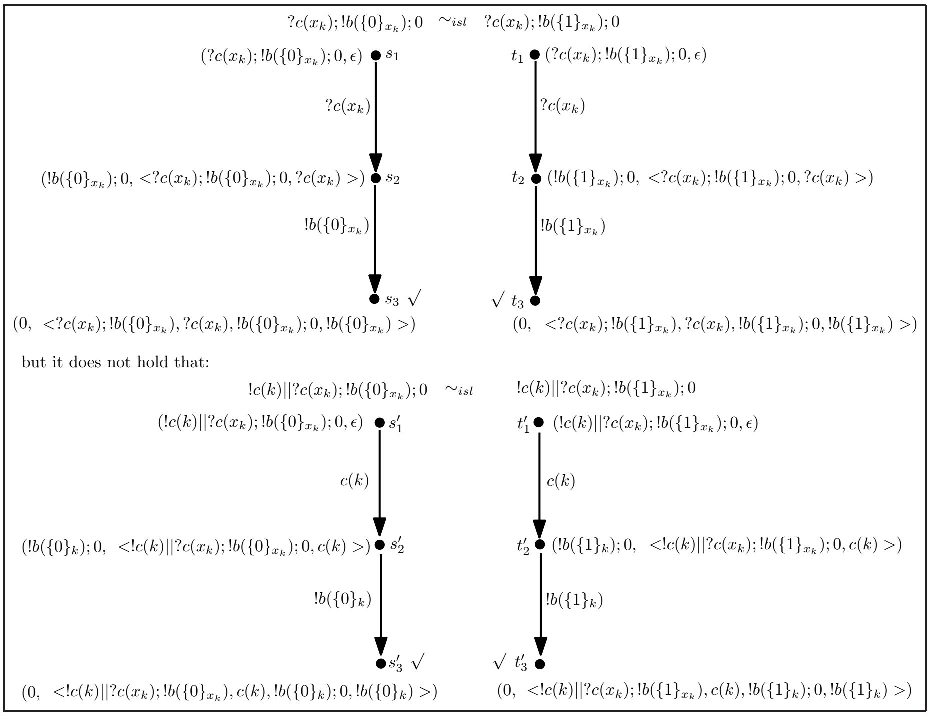

Figure 5: Example of breaking up the congruency property of the initially stateless pattern bisimilarity.

and so there is no bijection σ to hold that $CPattern^i(\Gamma_{s'_3}) = CPattern^i(\Gamma_{t'_3}\sigma)$.

To remedy this problem, the notion of stateless bisimilarity has been introduced, as in [Mousavi et al. 2003, Groote and Ponse 1994, Mousavi et al. 2004]. We adapt this notion to our setting in the following definition.

Definition 18 Stateless Pattern Bisimulation Relation. Let $\Gamma_p, \Gamma_{p'}, \Gamma_q, \Gamma_{q'} \in \mathcal{C}omp$; $p, p', q, q' \in \mathcal{P}roc$; $\alpha \in \mathcal{A}ct$; $M, M' \in \mathcal{T}erm$; and σ as a bijection. The symmetric relation R^σ_{sl} on processes is a stateless pattern bisimulation relation if and only if for all processes p and q, if $p\ R^\sigma_{sl}\ q$ then the following statements hold for each two indistinguishable computations Γ_p and Γ_q that are consistent with p and q, respectively:

- $\forall_{d,p',\Gamma_{p'}}$ if $(p,\Gamma_p) \xrightarrow{d} (p',\Gamma_{p'})$ then:
  - there exists a transition $(q,\Gamma_q) \xrightarrow{d'} (q',\Gamma_{q'})$, and
  - $\Gamma_{p'} \stackrel{i}{=} \Gamma_{q'}\sigma$ for each $i \in \mathcal{I}d$, and
  - $p'\ R^\sigma_{sl}\ q'$,

- if $(p, \Gamma_p)\checkmark$ then $(q, \Gamma_q)\checkmark$, and
- if $(p, \Gamma_p) \overset{i}{\cdots} (p', \Gamma_{p'})$ then there exists a relation $(q, \Gamma_q) \overset{i}{\cdots} (q', \Gamma_{q'})$ such that $p' \; R^{\sigma}_{sl} \; q'$.

p and q are stateless pattern bisimilar, denoted by $p \sim_{sl} q$, if and only if there exists a relation R^{σ}_{sl} s.t. $(p, q) \in R^{\sigma}_{sl}$.

Theorem 19. *Stateless pattern bisimularity is an equivalence relation.*

Proof. Equivalence of the stateless pattern bisimilarity follows trivially from the equivalence property of the original stateless equivalence bisimilarity [Mousavi et al. 2004] and that of our indistinguishability relation [Mahrooghi and Mousavi 2011]. □

Theorem 20. *The stateless pattern bisimularity is a congruence with respect to CryptoPAi operators.*

Proof. Common to the proof of Theorem 16 and using a similar line of reasoning as in Lemma 17, we can drop the substitutions and tacitly assume that all substitutions can be replaced by the identity substitution. To prove the theorem, the following propositions should be proven. For all decorated parametric actions $d = (I)\alpha(\overrightarrow{M})$ and $d' = (I)\alpha(\overrightarrow{M'})$ where $\overrightarrow{M} \equiv_p \overrightarrow{M'}$, and for all processes p, p', q, and q', if $p \sim_{sl} p'$ and $q \sim_{sl} q'$ then we have:

(1) $d; p \sim_{sl} d'; p'$, where $Appear^i(d) = Appear^i(d')$ for each identity i,

(2) $p + q \sim_{sl} p' + q'$,

(3) $\partial(p) \sim_{sl} \partial(p')$, and

(4) $p \,||\, q \sim_{sl} p' \,||\, q'$.

Let R_p be the minimum bisimulation relation relating p and p', R_q be the one relating q and q', I_d be the identity relation, and $\Gamma, \Gamma' \in \mathcal{C}omp$ be indistinguishable computations s.t. $\Gamma \overset{i}{=} \Gamma'$ for all $i \in \mathcal{I}d$.

1. We construct the relation $R \triangleq I_d \cup R_p \cup \{(d; p, d'; p') \mid (p, p') \in R_p \wedge (\forall_{i \in \mathcal{I}d}\ Appear^i(d) = Appear^i(d'))\}$. Then;

- If $(d; p, \Gamma) \overset{d}{\Rightarrow} (u, \Gamma_u)$, it is due to the rule (**d**) where $\Gamma_u = \Gamma ^\frown (d; p, d)$. Applying the same deduction rule concludes that $(d'; p', \Gamma') \overset{d'}{\Rightarrow} (u', \Gamma'_u)$ where $u' = p'$, $\Gamma'_u = \Gamma' {}^\frown (d'; p', d')$. Since $Appear^i(d) = Appear^i(d')$ and $\Gamma \overset{i}{=} \Gamma'$, we have $\Gamma_u \overset{i}{=} \Gamma'_u$, for each identity i. It follows that $(u, u') \in R_p$ and hence, this condition is satisfied.

- If $(d; p, \Gamma) \overset{i}{\cdots} (w, \Gamma_w)$, for some $i \in \mathcal{I}d$, it is only derivable from the rule $(\mathbf{I_C})$ and hence, it holds that $\Gamma \overset{i}{=} \Gamma_w$. Since Γ and Γ' are indistinguishable and indistinguishability is an equivalence relation, we have $\Gamma' \overset{i}{=} \Gamma_w$ that follows $(d'; p', \Gamma') \overset{i}{\cdots} (w, \Gamma_w)$, for all identity i. $(w, w) \in I_d$ proves that this condition is also satisfied.

2. We construct the relation $R \triangleq I_d \cup R_p \cup R_q \cup \{(p+q, p'+q') \mid (p, p') \in R_p \wedge (q, q') \in R_q\}$. It suffices to prove the pairs of process terms $(p+q, p'+q')$ satisfy the transfer conditions. Then;

- If $(p+q, \Gamma) \overset{d}{\Rightarrow} (u, \Gamma_u)$, this transition is due to the rule $(\mathbf{n0})$ where $u = p_1$ and $\Gamma_u = \Gamma^\frown(p+q, d)$. It follows that necessarily $(p, \Gamma) \overset{d}{\Rightarrow} (p_1, \Gamma_p)$ where $\Gamma_p = \Gamma^\frown(p, d)$. But since $p \sim_{sl} p'$, we have $(p', \Gamma') \overset{d'}{\Rightarrow} (p'_1, \Gamma'_p)$ where $p_1 \sim_{sl} p'_1$, $\Gamma'_p = \Gamma'{}^\frown(p', d')$, $\Gamma \overset{i}{=} \Gamma'$, and $\Gamma_p \overset{i}{=} \Gamma'_p$, for each $i \in \mathcal{I}d$. The application of the rule $(\mathbf{n0})$ concludes that $(p'+q', \Gamma') \overset{d'}{\Rightarrow} (u', \Gamma'_u)$ where $u' = p'_1$ and $\Gamma'_u = \Gamma'{}^\frown(p'+q', d')$. Similar to the case 1, $\Gamma_u \overset{i}{=} \Gamma'_u$; and due to the fact that $(u, u') \in R$, the condition is satisfied.

- If $(p+q, \Gamma) \overset{i}{\cdots} (w, \Gamma_w)$ for some $i \in \mathcal{I}d$, this case is identical to indistinguishability for action prefixing and hence, it is not repeated.

- If $(p+q, \Gamma)\checkmark$, it is due to the rule $(\mathbf{n2})$ that follows $(p, \Gamma_p)\checkmark$ (or $(q, \Gamma_q)\checkmark$). Since $p \sim_{sl} p'$ (and $q \sim_{sl} q'$), we have $(p', \Gamma'_p)\checkmark$ (or $(q', \Gamma'_q)\checkmark$). By applying the same deduction rule, we obtain $(p'+q', \Gamma')\checkmark$.

3. Let the relation R to be $R \triangleq I_d \cup \{(\partial(p), \partial(p')) \mid (p, p') \in R_p\}$. Then;

- If $(\partial(p), \Gamma) \overset{(\mathtt{J})a(\overrightarrow{M})}{\Rightarrow} (u, \Gamma_u)$, this transition is due to the rule $(\mathbf{encap})$ where $u = \partial(p_1)$ and $\Gamma_u = \Gamma^\frown(\partial(p), (\mathtt{J})a(\overrightarrow{M}))$. It follows that necessarily $(p, \Gamma) \overset{(\mathtt{J})a(\overrightarrow{M})}{\Rightarrow} (p_1, \Gamma_P)$ where $\Gamma_p = \Gamma^\frown(p, (\mathtt{J})a(\overrightarrow{M}))$. But since $p \sim_{sl} p'$, we have $(p', \Gamma') \overset{(\mathtt{J})a(\overrightarrow{M'})}{\Rightarrow} (p'_1, \Gamma'_p)$ such that $\Gamma'_p = \Gamma'{}^\frown(p', (\mathtt{J})a(\overrightarrow{M'}))$, $\Gamma \overset{i}{=} \Gamma'$, and $\Gamma_p \overset{i}{=} \Gamma'_p$, for all $i \in \mathcal{I}d$. Applying the rule $(\mathbf{encap})$ concludes that $(\partial(p'), \Gamma') \overset{(\mathtt{J})a(\overrightarrow{M'})}{\Rightarrow} (u', \Gamma'_u)$ where $\Gamma'_u = \Gamma'{}^\frown(\partial(p'), (\mathtt{J})a(\overrightarrow{M'}))$ and $u' = \partial(p'_1)$. Because $(u, u') \in R$ and $\Gamma_u \overset{i}{=} \Gamma'_u$, this condition is satisfied.

- If $(\partial(p), \Gamma) \overset{i}{\cdots} (w, \Gamma_w)$, for some $i \in \mathcal{I}d$, it is due to the deduction rule $(\mathbf{I_C})$ and hence, $\Gamma \overset{i}{=} \Gamma_w$ that implies $(p, \Gamma) \overset{i}{\cdots} (w, \Gamma_w)$. Since $p \sim_{sl} p'$, there exits a state (w', Γ'_w) s.t. $w \sim_{sl} w'$, $(p', \Gamma') \overset{i}{\cdots} (w', \Gamma'_w)$, $\Gamma \overset{i}{=} \Gamma'$, and $\Gamma' \overset{i}{=} \Gamma'_w$, for each identity i. The application of the rule $(\mathbf{I_C})$

concludes that $(\partial(p'), \Gamma') \overset{i}{\cdots} (w', \Gamma'_w)$ for each $i \in \mathcal{I}d$. The facts that $(w, w') \in R$ and $\Gamma_w \overset{i}{=} \Gamma'_w$ prove the satisfaction of this condition.

4. We construct the relation $R \triangleq I_d \cup R_p \cup R_q \cup \{(p \,||\, q, p' \,||\, q') \mid (p, p') \in R_p \wedge (q, q') \in R_q\}$. Then;

- If $(p \,||\, q, \Gamma) \overset{d}{\Rightarrow} (u, \Gamma_u)$, this transition is due to the rule $(\mathbf{p0})$ where $u = p_1$ and $\Gamma_u = \Gamma^\frown(p \,||\, q, d)$. It follows that necessarily $(p, \Gamma) \overset{d}{\Rightarrow} (p_1, \Gamma_p)$ where $\Gamma_p = \Gamma^\frown(p, d)$. But since $p \sim_{sl} p'$, we have $(p', \Gamma') \overset{d'}{\Rightarrow} (p'_1, \Gamma'_p)$ where $p_1 \sim_{sl} p'_1$, $\Gamma'_p = \Gamma'{}^\frown(p', d')$, $\Gamma \overset{i}{=} \Gamma'$, and $\Gamma_p \overset{i}{=} \Gamma'_p$, for each identity i. The application of the rule $(\mathbf{p0})$ concludes that $(p' \,||\, q', \Gamma') \overset{d'}{\Rightarrow} (u', \Gamma'_u)$ where $u' = p'_1$ and $\Gamma'_u = \Gamma'{}^\frown(p' \,||\, q', d')$. Because $(u, u') \in R$ and $\Gamma_u \overset{i}{=} \Gamma'_u$, this condition is satisfied.

- If $(p \,||\, q, \Gamma) \overset{(\mathtt{J})a(\overrightarrow{M})}{\Rightarrow} (u, \Gamma_u)$, it is due to the rule $(\mathbf{p3})$ where $u = p_1 \,||\, q_1$ and $\Gamma_u = \Gamma^\frown(p \,||\, q, (\mathtt{J})a(\overrightarrow{M}))$. It follows that $(p, \Gamma) \overset{(\mathtt{J_p})?a(\overrightarrow{M'})}{\Rightarrow} (p_1, \Gamma_p)$ and $(q, \Gamma) \overset{(\mathtt{J_q})!a(\overrightarrow{M})}{\Rightarrow} (q_1, \Gamma_q)$ where $\mathtt{J} = \mathtt{J_p} \cup \mathtt{J_q}$, $\Gamma_p = \Gamma^\frown(p, (\mathtt{J_p})?a(\overrightarrow{M'}))$, and $\Gamma_q = \Gamma^\frown(q, (\mathtt{J_q})!a(\overrightarrow{M}))$. Since $p \sim_{sl} p'$ and $q \sim_{sl} q'$, we have $(p', \Gamma') \overset{(\mathtt{J_p})?a(\overrightarrow{M'_1})}{\Rightarrow} (p'_1, \Gamma'_p)$ and $(q', \Gamma') \overset{(\mathtt{J_q})!a(\overrightarrow{M_1})}{\Rightarrow} (q'_1, \Gamma'_p)$ where $p_1 \sim_{sl} p'_1$, $\Gamma'_p = \Gamma'{}^\frown(p', (\mathtt{J_p})?a(\overrightarrow{M'}))$, $q_1 \sim_{sl} q'_1$, and $\Gamma'_q = \Gamma'{}^\frown(q', (\mathtt{J_q})!a(\overrightarrow{M'}))$, $M \equiv_p M_1$, $M' \equiv_p M'_1$, $\Gamma \overset{i}{=} \Gamma'$, $\Gamma_p \overset{i}{=} \Gamma'_p$, and $\Gamma_q \overset{i}{=} \Gamma'_q$, for each identity i. Applying the rule $(\mathbf{p3})$ concludes that $(p' \,||\, q', \Gamma') \overset{(\mathtt{J})a(\overrightarrow{M_1})}{\Rightarrow} (u', \Gamma'_u)$ where $u' = p'_1 \,||\, q'_1$ and $\Gamma'_u = \Gamma'{}^\frown(p' \,||\, q', (\mathtt{J})a(\overrightarrow{M_1}))$. Because $M \equiv_p M_1$, $(u, u') \in R$, and $\Gamma_u \overset{i}{=} \Gamma'_u$, this condition is satisfied.

- If $(p \,||\, q, \Gamma) \overset{i}{\cdots} (w, \Gamma_w)$, for some $i \in \mathcal{I}d$, it is due to using the rule $(\mathbf{I_C})$ and hence, $\Gamma \overset{i}{=} \Gamma_w$ that implies $(p, \Gamma) \overset{i}{\cdots} (w, \Gamma_w)$. Since $p \sim_{sl} p'$, there exits a state (w', Γ'_w) s.t. $w \sim_{sl} w'$, $(p', \Gamma') \overset{i}{\cdots} (w', \Gamma'_w)$, $\Gamma \overset{i}{=} \Gamma'$, and $\Gamma' \overset{i}{=} \Gamma'_w$, for each identity i . The application of the rule $(\mathbf{I_C})$ concludes that $(p' \,||\, q', \Gamma') \overset{i}{\cdots} (w', \Gamma'_w)$ for each identity i. The facts that $(w, w') \in R$ and $\Gamma_w \overset{i}{=} \Gamma'_w$ prove the satisfaction of this condition.

- If$(p \,||\, q, \Gamma)\checkmark$, this transition is due to the rule $(\mathbf{p2})$ that follows $(p, \Gamma_p)\checkmark$ and $(q, \Gamma_q)\checkmark$. But since $p \sim_{sl} p'$ and $q \sim_{sl} q'$ this holds if and only if $(p', \Gamma'_p)\checkmark$ and $(q', \Gamma'_q)\checkmark$. By applying the same deduction rule, we obtain $(p' \,||\, q', \Gamma')\checkmark$.

$\square$

3.2 Comparing the Notions of Bisimilarity

Among the notions of pattern bisimilarity defined in this section, the state-based pattern bisimilarity is the least robust and also the weakest one. The reason, as witnessed by Example 5, is that the state-based pattern bisimulation attaches a fixed initial computation to the related processes and confines the changes of computation to the related processes. Hence, any change in the initial computation or interference with the intermediate ones will break the state-based pattern bisimilarity. The initially stateless pattern bisimilarity allows for arbitrary initial states and hence, is stronger and more robust than the state-based pattern bisimilarity, but still does not allow for intermediate interference with the computations. The stateless pattern bisimilarity is the strongest (finest) notion and the most robust one. Our comparison regarding the strength of these different notions is summarized in the following lemma.

Lemma 21. *For each two closed process terms p and q, and each two indistinguishable computations Γ and Γ', we have*
(1) $p \sim_{sl} q$ *implies* $p \sim_{isl} q$,
(2) $p \sim_{isl} q$ *implies* $(p, \Gamma) \sim_{sb} (q, \Gamma')$.

Proof. Below, we prove two implications separately:

1. Since p and q are stateless pattern bisimilar, they can mimic each other's transitions with respect to any arbitrary (indistinguishable and consistent) pairs of initial states and the resulting processes are again related. Construct a relation on pairs of processes and computations, that relates p and q on all (indistinguishable and consistent) initial states and the rest of the related processes (in the stateless pattern bisimulation relation) only on those computations, that are reachable from the pairs of initial states mentioned above. This is clearly a state-based pattern bisimulation relation as the transfer conditions are only subsets of the transfer conditions available from the stateless pattern bisimulation relation. Since p and q are related using all initial computations, p and q are initially stateless pattern bisimilar.

2. Immediate consequence of Definition 14.

□

Note that Examples 5 and 6 also provide witnesses that the inclusions indicated in Lemma 21 are indeed strict.

4 Axiomatizing Sequential *CryptoPAi*

In this section, we present a sound and ground-complete axiomatization of *CryptoPAi* modulo stateless pattern bisimilarity (Definition 18). Our choice for

$A1$	$p + 0 = p$	$A2$	$p + p = p$
$A3$	$p + q = q + p$	$A4$	$p + (q + r) = (p + q) + r$
$D1$	$\partial((\mathtt{I})!a(\overrightarrow{M}); p) = 0$	$D2$	$\partial((\mathtt{I})?a(\overrightarrow{M}); p) = 0$
$D3$	$\partial((\mathtt{I})a(\overrightarrow{M}); p) = (\mathtt{I})a(\overrightarrow{M}); \partial(p)$	$D4$	$\partial(p + q) = \partial(p) + \partial(q)$

Figure 6: Axiomatization of *CryptoPAi* Terms

stateless pattern bisimilarity is motivated by its robustness and the fact that axioms proven sound for this notion also remain sound for the weaker notions introduced in Section 3.

We start with a set of standard axioms, as shown in Figure 6, by exploiting from [Bergstra and Klop 1987], where $p, q, r \in \mathcal{P}roc$ and $d, d' \in D$. To take care of transitions that feature syntactically different, yet observationally equivalent labels, we extend our axiomatization system with the axioms given in Figure 7. We write $CryptoPAi \vdash p = q$ to denote that the equation $p = q$ is derivable from the axiom system given in Figures 6 and 7.

4.1 Soundness

In this section, we prove that our axiomatization is sound for the term algebra of *CryptoPAi* modulo stateless pattern bisimularity.

Theorem 22. *The process theory CryptoPAi is a sound axiomatization of the term algebra of CryptoPAi modulo* $\sim_{sl}$.

Proof. We need to prove that our axiomatization is sound for the term algebra of processes modulo $\sim_{sl}$. It must be shown that, for each axiom $p = q$ of *CryptoPAi*, $\mathbb{P}(CryptoPAi)/_{\sim_{sl}} \models p = q$. Let Γ, Γ' be indistinguishable computations s.t. $\Gamma \stackrel{i}{=} \Gamma'$ for all $i \in \mathcal{I}d$. The proofs are given below.

- **A1.** Let the minimum bisimulation relation relating p and p be the identity relation denoted by I_d. It suffices to give a bisimulation on process terms that contain all pairs $(p + 0, p)$. We construct the relation $R \triangleq I_d \cup \{(p + 0, p) \mid (p, p) \in I_d\}$. According to Definition 18, it should be proven that R is a stateless pattern bisimulation relation. To this aim, we show that all elements of R satisfy Definition 18. The definition trivially holds for all pairs (p, p). Hence, we consider the elements of R which have the form $(p + 0, p)$. Let $(p + 0, p)$ be such an element. Then;
  - If $(p + 0, \Gamma) \stackrel{d}{\Rightarrow} (p', \Gamma_u)$, this transition is due to the rule (**n0**). It follows that $(p, \Gamma) \stackrel{d}{\Rightarrow} (p', \Gamma'_u)$. The fact that $(p', p') \in R$ proves this case.

- If $(p+0,\Gamma) \overset{i}{\cdots} (w,\Gamma_w)$ for some $i \in \mathcal{I}d$, this is only derivable from the rule $(\mathbf{I_C})$ and hence, it holds that $\Gamma \overset{i}{=} \Gamma_w$ and consequently, we have $(p,\Gamma') \overset{i}{\cdots} (w,\Gamma_w)$. The fact that $(w,w) \in R$ indicates the satisfaction of this condition.

- **A2.** Let the minimum bisimulation relation relating p and p be the identity relation denoted by I_d. Similarly to the case $A1$, it suffices to give a bisimulation on process terms that contain all pairs $(p+p,p)$. We construct the relation $R \triangleq I_d \cup \{(p+p,p) \mid (p,p) \in I_d\}$. The definition of stateless pattern bisimulation relation trivially holds for all pairs (p,p) and so we consider only the elements of R which have the form $(p+p,p)$. Let $(p+p,p)$ be such an element. Then;

  - If $(p+p,\Gamma) \overset{d}{\Rightarrow} (p',\Gamma_u)$, this transition is due to the rule $(\mathbf{n0})$ (or $(\mathbf{n1})$). It follows that necessarily $(p,\Gamma) \overset{d}{\Rightarrow} (p',\Gamma'_u)$. The fact that $(p',p') \in R$ proves that this transfer condition is satisfied.

  - If $(p+p,\Gamma) \overset{i}{\cdots} (w,\Gamma_w)$, for some $i \in \mathcal{I}d$, this is only derivable from the rule $(\mathbf{I_C})$ and hence, it holds that $\Gamma \overset{i}{=} \Gamma_w$. Consequently, we have $(p,\Gamma') \overset{i}{\cdots} (w,\Gamma_w)$. The fact that $(w,w) \in R$ indicates the satisfaction of this condition.

- **A3.** Let the minimum bisimulation relation relating p and p, and also q and q, be the identity relation denoted by I_d. We construct the relation $R \triangleq I_d \cup \{(p+q,q+p) \mid (p,p) \text{ and } (q,q) \in I_d\}$. The definition trivially holds for all pairs in I_d and so we consider only the elements of R which have the form $(p+q,q+p)$. Let $(p+q,q+p)$ be such an element. Then;

  - If $(p+q,\Gamma) \overset{d}{\Rightarrow} (u,\Gamma_u)$, this transition is due to the rule $(\mathbf{n0})$ where $u=p_1$ and $\Gamma_u = \Gamma^\frown(p+q,d)$. It follows that $(p,\Gamma) \overset{d}{\Rightarrow} (p_1,\Gamma_p)$ where $\Gamma_p = \Gamma^\frown(p,d)$. The application of the rule $(\mathbf{n1})$ (symmetric version of $(\mathbf{n0})$) yields $(q+p,\Gamma) \overset{d}{\Rightarrow} (p_1,\Gamma'_u)$ that follows $\Gamma'_u = \Gamma^\frown(q+p,d)$. The facts Γ_u and Γ'_u are indistinguishable and $(p_1,p_1) \in R$ prove the satisfaction of this condition.

  - If $(p+q,\Gamma) \overset{i}{\cdots} (w,\Gamma_w)$, for some $i \in \mathcal{I}d$, it is due to using the rule $(\mathbf{I_C})$ and hence, it holds that $\Gamma \overset{i}{=} \Gamma_w$. It follows that $(q+p,\Gamma') \overset{i}{\cdots} (w,\Gamma_w)$. The fact that $(w,w') \in R$ proves the satisfaction of this condition.

  - If $(p+q,\Gamma)\checkmark$, this transition is due to the rule $(\mathbf{n2})$ that follows $(p,\Gamma_p)\checkmark$ (or $(q,\Gamma_q)\checkmark$). By applying the same rule $(\mathbf{n2})$, we obtain $(q+p,\Gamma')\checkmark$.

- **A4.** Let the minimum bisimulation relation relating p and p, q and q, and also r and r, be the identity relation denoted by I_d. We construct the relation $R \triangleq I_d \cup \{(p+(q+r), (p+q)+r) \mid (p,p), (q,q) \text{ and } (r,r) \in I_d\}$. The definition trivially holds for all pairs in I_d and so we consider only the elements of R which have the form $(p+(q+r), (p+q)+r)$. Let $(p+(q+r), (p+q)+r)$ be such an element. Then;

  - If $(p+(q+r), \Gamma) \overset{d}{\Rightarrow} (u, \Gamma_u)$, this transition is due to the rule (**n0**) where $\Gamma_u = \Gamma \frown (p+(q+r), d)$ and $u = p_1$. It follows that $(p, \Gamma) \overset{d}{\Rightarrow} (u, \Gamma_p)$ where $\Gamma_p = \Gamma \frown (p, d)$. Two times application of the rule (**n0**) concludes that $(p+q, \Gamma) \overset{d}{\Rightarrow} (u, \Gamma_{pq})$ and $((p+q)+r, \Gamma) \overset{d}{\Rightarrow} (u, \Gamma'_u)$ s.t. $\Gamma_{pq} = \Gamma \frown (p+q, d)$ and $\Gamma'_u = \Gamma \frown ((p+q)+r, d)$. The facts Γ_u and Γ'_u are indistinguishable and $(u, u) \in R$ prove this case.

  - If $(p+(q+r), \Gamma) \overset{i}{\cdots} (w, \Gamma_w)$, for some $i \in \mathcal{I}d$, this is only derivable from the rule ($\mathbf{I_C}$) and hence, it holds that $\Gamma \overset{i}{=} \Gamma_w$. Consequently, we have $((p+q)+r, \Gamma') \overset{i}{\cdots} (w, \Gamma_w)$. This transfer condition is satisfied due to $\Gamma_w \overset{i}{=} \Gamma_w$ and $(w, w) \in R$.

- **D1-D2.** The proofs of these cases are straightforward because there is no possible transition for $(\partial((\mathtt{I})!a(\overrightarrow{M}); p), \Gamma)$ or $(\partial((\mathtt{I})?a(\overrightarrow{M}); p), \Gamma)$.

- **D3.** We construct the relation $R \triangleq I_d \cup \{(\partial((\mathtt{I})a(\overrightarrow{M}); p), (\mathtt{I})a(\overrightarrow{M}); \partial(p))\}$. The definition of stateless pattern bisimulation relation trivially holds for all pairs $(\partial(p), \partial(p))$ and hence, we consider only the elements of R which have the form $(\partial((\mathtt{I})a(\overrightarrow{M}); p), (\mathtt{I})a(\overrightarrow{M}); \partial(p))$. Let $(\partial((\mathtt{I})a(\overrightarrow{M}); p), (\mathtt{I})a(\overrightarrow{M}); \partial(p))$ be such an element. Then;

  - If $(\partial((\mathtt{I})a(\overrightarrow{M}); p), \Gamma) \overset{(\mathtt{I})a(\overrightarrow{M})}{\Rightarrow} (u, \Gamma_u)$, this transition is due to the rule (**encap**) where $u = \partial(p)$ and $\Gamma_u = \Gamma \frown (\partial((\mathtt{I})a(\overrightarrow{M}); p), (\mathtt{I})a(\overrightarrow{M}))$. By applying the rule (**d**), we obtain $((\mathtt{I})a(\overrightarrow{M'}); \partial(p), \Gamma') \overset{(\mathtt{I})a(\overrightarrow{M'})}{\Rightarrow} (u', \Gamma'_u)$ s.t. $u' = \partial(p)$ and $\Gamma'_u = \Gamma' \frown ((\mathtt{I})a(\overrightarrow{M'}); \partial(p), (\mathtt{I})a(\overrightarrow{M'}))$. Consequently, we have $M \equiv_p M'$ and $\Gamma \overset{i}{=} \Gamma'$ for all $i \in \mathcal{I}d$. Because $(u, u') \in R$, this transfer condition is satisfied.

  - If $(\partial((\mathtt{I})a(\overrightarrow{M}); p), \Gamma) \overset{i}{\cdots} (w, \Gamma_w)$, for some $i \in \mathcal{I}d$, it is due to using the rule ($\mathbf{I_C}$) and hence, it holds that $\Gamma \overset{i}{=} \Gamma_w$. Consequently, we have $((\mathtt{I})a(\overrightarrow{M}); \partial(p), \Gamma') \overset{i}{\cdots} (w, \Gamma_w)$. The fact that $(w, w) \in R$ proves the satisfaction of this condition.

- **D4.** Let the minimum bisimulation relation relating $\partial(p)$ and $\partial(p)$ be the identity relation denoted by I_d. We construct the relation $R \triangleq I_d \cup \{(\partial(p+q), \partial(p)+\partial(q))\}$. The definition of stateless pattern bisimulation relation trivially holds for all pairs $(\partial(p), \partial(p))$ and so we consider only the elements of R which have the form $((\partial(p+q), \partial(p)+\partial(q))$. Let $((\partial(p+q), \partial(p)+\partial(q))$ be such an element. Then;

  - If $(\partial(p+q), \Gamma) \overset{(\mathtt{I})a(\overrightarrow{M})}{\Rightarrow} (u, \Gamma_u)$, this transition is due to the rule (**encap**) where $u = \partial(u_0)$ and $\Gamma_u = \Gamma^\frown(\partial(p+q), (\mathtt{I})a(\overrightarrow{M}))$. It follows that necessarily $(p+q, \Gamma) \overset{(\mathtt{I})a(\overrightarrow{M})}{\Rightarrow} (u_0, \Gamma_{u_1})$. By applying the rule (**n0**) (or (**n1**)), we obtain $(p, \Gamma) \overset{(\mathtt{I})a(\overrightarrow{M})}{\Rightarrow} (u_0, \Gamma_{u_2})$ (or $(q, \Gamma) \overset{(\mathtt{I})a(\overrightarrow{M})}{\Rightarrow} (u_0, \Gamma_{u_3})$). It follows that $(\partial(p), \Gamma') \overset{(\mathtt{I})a(\overrightarrow{M})}{\Rightarrow} (\partial(u_0), \Gamma'_{u_1})$ (or $(\partial(q), \Gamma) \overset{(\mathtt{I})a(\overrightarrow{M})}{\Rightarrow} (\partial(u_0), \Gamma'_{u_2})$) by re-applying the rule (**encap**). The application of the rule (**n0**) concludes $(\partial(p) + \partial(q), \Gamma') \overset{(\mathtt{I})a(\overrightarrow{M})}{\Rightarrow} (\partial(u_0), \Gamma'_u)$. The fact that $(\partial(u_0), \partial(u_0)) \in R$ proves this case.

  - If $(\partial(p+q), \Gamma) \overset{i}{\cdots} (w, \Gamma_w)$, for some $i \in \mathcal{I}d$, it is due to using the rule ($\mathbf{I_C}$) and hence, it holds that $\Gamma \overset{i}{=} \Gamma_w$. Consequently, we have $(\partial(p) + \partial(q), \Gamma') \overset{i}{\cdots} (w, \Gamma_w)$. Because $(w, w) \in R$, this transfer condition is satisfied.

- **D5.** Let the minimum bisimulation relation relating p and p be the identity relation denoted by I_d. It suffices to give a bisimulation on process terms that contains all pairs $((\mathtt{I})\alpha(\overrightarrow{M}); p, (\mathtt{J})\alpha(\overrightarrow{M}); p)$ s.t. $\rho(\alpha(\overrightarrow{M})) = \alpha(\overrightarrow{M})$. We construct the relation $R \triangleq I_d \cup \{((\mathtt{I})\alpha(\overrightarrow{M}); p, (\mathtt{J})\alpha(\overrightarrow{M}); p) \mid \rho(\alpha(\overrightarrow{M})) = \alpha(\overrightarrow{M})\}$. The definition of stateless pattern bisimulation relation trivially holds for all pairs (p, p) and so we consider only the elements of R which have the form $((\mathtt{I})\alpha(\overrightarrow{M}); p, (\mathtt{J})\alpha(\overrightarrow{M}); p)$. Let $((\mathtt{I})\alpha(\overrightarrow{M}); p, (\mathtt{J})\alpha(\overrightarrow{M}); p)$ be such an element. Then;

  - If $((\mathtt{I})\alpha(\overrightarrow{M}); p, \Gamma) \overset{(\mathtt{I})\alpha(\overrightarrow{M})}{\Rightarrow} (p, \Gamma_u)$, this transition is due to the rule (**d**) where $\Gamma_u = \Gamma^\frown((\mathtt{I})\alpha(\overrightarrow{M}); p, (\mathtt{I})\alpha(\overrightarrow{M}))$. Applying the same rule (**d**) yields $((\mathtt{J})\alpha(\overrightarrow{M}); p, \Gamma') \overset{(\mathtt{J})\alpha(\overrightarrow{M})}{\Rightarrow} (p, \Gamma'_u)$ s.t. $\Gamma'_u = \Gamma'^\frown((\mathtt{J})\alpha(\overrightarrow{M}); p, (\mathtt{J})\alpha(\overrightarrow{M}))$. Definition 10 and the condition of Axiom $D5$ follow that $\rho(\alpha(\overrightarrow{M})) = \alpha(\overrightarrow{M})$ and $\Gamma_u \overset{i}{=} \Gamma'_u$. The fact that $(p, p) \in R$ proves this case.

  - If $((\mathtt{I})\alpha(\overrightarrow{M}); p, \Gamma) \overset{i}{\cdots} (w, \Gamma_w)$, for some $i \in \mathcal{I}d$, it is due to using the rule ($\mathbf{I_C}$) and hence, it holds that $\Gamma \overset{i}{=} \Gamma_w$. Consequently, we have $((\mathtt{J})\alpha(\overrightarrow{M}); p, \Gamma') \overset{i}{\cdots} (w, \Gamma_w)$. Because $(w, w) \in R$, this transfer condition is satisfied.

$D5$	$(\mathtt{I})\alpha(\overrightarrow{M});p$	$=$	$(\mathtt{J})\alpha(\overrightarrow{M});p$	if	$\rho(\alpha(\overrightarrow{M})) = \alpha(\overrightarrow{M})$
$D6$	$(\mathtt{I})\alpha(\overrightarrow{M});p$	$=$	$(\mathtt{I})\alpha(\overrightarrow{M'});p$	if	$\rho(\alpha(\overrightarrow{M})) = \rho(\alpha(\overrightarrow{M'}))$ and $M \equiv_p M'$
$D7$	$(\mathtt{I})\alpha(\overrightarrow{M});p$	$=$	$(\mathtt{I})\alpha(\overrightarrow{M'});p$	if	$I = \emptyset$ and $\rho(\alpha(\overrightarrow{M})) = \rho(\alpha(\overrightarrow{M'}))$
$D8$	$(\mathtt{I})\alpha(\overrightarrow{M});p$	$=$	$(\overline{\mathtt{I}})\alpha(\overrightarrow{M'});p$	if	$\rho(\alpha(\overrightarrow{M})) = \alpha(\overrightarrow{M'})$ and $\rho(\alpha(\overrightarrow{M'})) = \alpha(\overrightarrow{M})$

Figure 7: Extension of the Axiomatization of *CryptoPAi* Terms

- **D6.** Let the minimum bisimulation relation relating p and p be the identity relation denoted by I_d. It suffices to give a bisimulation on process terms that contains all pairs $((\mathtt{I})\alpha(\overrightarrow{M});p, (\mathtt{I})\alpha(\overrightarrow{M'});p)$ s.t. $\rho(\alpha(\overrightarrow{M})) = \rho(\alpha(\overrightarrow{M'}))$ and $M \equiv_p M'$. We construct the relation $R^\sigma \triangleq I_d \cup \{((\mathtt{I})\alpha(\overrightarrow{M});p, (\mathtt{I})\alpha(\overrightarrow{M'});p) \mid \rho(\alpha(\overrightarrow{M})) = \rho(\alpha(\overrightarrow{M'}))$ and $M \equiv_p M'\sigma\}$. Similar to the case $D5$, our bisimulation relation trivially holds for all pairs (p,p) and hence, we consider only the elements of R which have the form $((\mathtt{I})\alpha(\overrightarrow{M});p, (\mathtt{I})\alpha(\overrightarrow{M'});p)$. Let $((\mathtt{I})\alpha(\overrightarrow{M});p, (\mathtt{I})\alpha(\overrightarrow{M'});p)$ be such an element. Then;
  - If $((\mathtt{I})\alpha(\overrightarrow{M});p,\Gamma) \overset{(\mathtt{I})\alpha(\overrightarrow{M})}{\Rightarrow} (p,\Gamma_u)$, this transition is due to the rule (**d**) where $\Gamma_u = \Gamma \frown ((\mathtt{I})\alpha(\overrightarrow{M});p,(\mathtt{I})\alpha(\overrightarrow{M}))$. The application of this rule yields $((\mathtt{I})\alpha(\overrightarrow{M'});p,\Gamma') \overset{(\mathtt{I})\alpha(\overrightarrow{M'})}{\Rightarrow} (p,\Gamma'_u)$ where $\Gamma'_u = \Gamma' \frown ((\mathtt{I})\alpha(\overrightarrow{M'});p,(\mathtt{I})\alpha(\overrightarrow{M'}))$. Definition 10 and the conditions of Axiom $D6$ follow that necessarily $\Gamma_u \overset{i}{=} \Gamma'_u$. It implies $(p,p) \in R$ that proves this case.
  - If $((\mathtt{I})\alpha(\overrightarrow{M});p,\Gamma) \overset{i}{\cdots} (w,\Gamma_w)$, for some $i \in \mathcal{I}d$, it is due to using the rule ($\mathbf{I_C}$) and hence, it holds that $\Gamma \overset{i}{=} \Gamma_w$. Consequently, we have $((\mathtt{I})\alpha(\overrightarrow{M'});p,\Gamma') \overset{i}{\cdots} (w,\Gamma_w)$. Because $(w,w) \in R$, this transfer condition is satisfied.
- **D7-D8.** The proofs are similar to the case D6.

□

The soundness of our axiomatization of *CryptoPAi* guarantees that using the equational theory for deriving equalities cannot lead to mistakes. Conversely, the ground-completeness property assures that our axiomatization is strong enough to derive all stateless pattern bisimilarities between closed terms (i.e., assuming two closed *CryptoPAi*-terms p and q those are bisimilar, the equality of p and q is also derivable).

We recall the notion of summands as introduced in [Baeten et al. 2009]. For any *CryptoPAi*-terms p and q, it is said that p is a summand of q if and only if *CryptoPAi* $\vdash p + q = q$ (i.e., the summand p describes a subset of the alternatives of term q, and consequently can be added to q without essentially changing the process). In fact, the idea behind the ground-completeness proof is to show $CryptoPAi \vdash p = q$ via the intermediate results $CryptoPAi \vdash p = p+q$ and $CryptoPAi \vdash p+q = q$. These intermediate results imply $CryptoPAi \vdash p = p+q = q$. The proof of ground-completeness property shows that if p and q are stateless pattern bisimilar, it must be the case that every summand of p is a summand of q and vice versa. Corresponding to the above intermediate results, it implies that p as a whole can be seen as a summand of q and vice versa.

We need the following two lemmas in the ground-completeness proof.

Lemma 23. *Let p,q, and r be closed CryptoPAi-terms. If* $(p+q)+r \sim_{sl} r$*,then* $p+r \sim_{sl} r$ *and* $q+r \sim_{sl} r$.

Proof. We assume $d = (\mathtt{J})\alpha(\overrightarrow{M})$ and $d' = (\mathtt{J})\alpha(\overrightarrow{M'})$ where $\alpha \in \mathcal{A}ct$, $\mathtt{J} \subseteq \mathcal{I}d$, and $M, M' \in \mathcal{T}erm$ are pattern-equivalent terms (i.e., $M \equiv_p M'$). We construct the relation $R \triangleq I_d \cup \{(p+r, r) \mid (p+q)+r \sim_{sl} r\}$. The definition trivially holds for all pairs in I_d and so we consider only the elements of R which have the form $(p+r, r)$. Let $(p+r, r)$ be such an element.
If $((p+q)+r, \Gamma) \overset{d}{\Rightarrow} (u, \Gamma_u)$, this transition is due to one of the following deduction rules:

- (**n0**) that follows $\Gamma_u = \Gamma^\frown((p+q)+r, d)$, and then, by repeated application of the same rule, we obtain $(p, \Gamma) \overset{d}{\Rightarrow} (u, \Gamma'_u)$ and $(p+r, \Gamma) \overset{d}{\Rightarrow} (u, \Gamma''_u)$ where $\Gamma'_u = \Gamma^\frown(p, d)$ and $\Gamma''_u = \Gamma^\frown(p+r, d)$. It is clear that $\Gamma''_u \overset{i}{=} \Gamma_u$ for each $i \in \mathcal{I}d$.
- (**n1**) that follows $\Gamma_u = \Gamma^\frown((p+q)+r, d)$, and then, $(r, \Gamma_r) \overset{d'}{\Rightarrow} (r', \Gamma'_r)$ where $\Gamma'_r = \Gamma_r^\frown(r, d')$, $u \sim_{sl} r'$, $\Gamma \overset{i}{=} \Gamma_r$ and $\Gamma_u \overset{i}{=} \Gamma'_r$ for each $i \in \mathcal{I}d$.

Consequently, we have $\Gamma''_u \overset{i}{=} \Gamma'_r$ for each $i \in \mathcal{I}d$. The facts that $u \sim_{sl} r'$, $\Gamma \overset{i}{=} \Gamma_r$, and $\Gamma''_u \overset{i}{=} \Gamma'_r$ (for each $i \in \mathcal{I}d$) satisfy the conditions of $\sim_{sl}$ definition (see Definition 18). It means that $p+r \sim_{sl} r$. Similarly, it can be proven that $q+r \sim_{sl} r$. □

Lemma 24. *Let p be a closed CryptoPAi-term. If* $(p, \Gamma_p) \overset{(\mathtt{J})\alpha(\overrightarrow{M})}{\Rightarrow} (p', \Gamma_{p'})$ *for some closed term* p', $\alpha \in \mathcal{A}ct$, $\mathtt{J} \subseteq \mathcal{I}d$, *and* $M \in \mathcal{T}erm$, *then* $CryptoPAi \vdash p = (\mathtt{J})\alpha(M).p' + p$.

Proof. The property is proven by induction on the structure of p.

1. Assume $p \equiv 0$. This case cannot occur as $(0, \Gamma_p) \overset{(\mathtt{J})\alpha(\overrightarrow{M})}{\not\Rightarrow}$.

2. Assume $p \equiv p_1 + p_2$ for some closed terms p_1 and p_2. $(p, \Gamma_p) \overset{(\mathtt{J})\alpha(\overrightarrow{M})}{\Rightarrow} (p', \Gamma_{p'})$ follows that (**i**) $(p_1, \Gamma_p) \overset{(\mathtt{J})\alpha(\overrightarrow{M})}{\Rightarrow} (p', \Gamma_{p'})$ or (**ii**) $(p_2, \Gamma_p) \overset{(\mathtt{J})\alpha(\overrightarrow{M})}{\Rightarrow} (p', \Gamma_{p'})$.

By induction, (**i**) implies $CryptoPAi \vdash p_1 = (\mathtt{J})\alpha(M).p' + p_1$ and then $CryptoPAi \vdash p = p_1+p_2 = ((\mathtt{J})\alpha(M).p'+p_1)+p_2 = (\mathtt{J})\alpha(M).p'+(p_1+p_2) = (\mathtt{J})\alpha(M).p'+p$. Also, (**ii**) implies $CryptoPAi \vdash p_2 = (\mathtt{J})\alpha(M).p'+p_2$ and then $CryptoPAi \vdash p = p_1+p_2 = p_1+((\mathtt{J})\alpha(M).p'+p_2) = (\mathtt{J})\alpha(M).p'+(p_1+p_2) = (\mathtt{J})\alpha(M).p' + p$.

3. Assume $p \equiv (\mathtt{I})b(\overrightarrow{M'}).p''$ for some closed term p'', $b \in \mathcal{A}ct$, $\mathtt{I} \subseteq \mathcal{I}d$, and $M' \in \mathcal{T}erm$. The assumption that $(p, \Gamma_p) \overset{(\mathtt{J})a(\overrightarrow{M})}{\Rightarrow} (p', \Gamma'_p)$ implies that $b \equiv a$, $J \equiv I$, and $M \equiv M'$. Thus, $CryptoPAi \vdash p+p = (\mathtt{I})b(\overrightarrow{M'}).p''+p = (\mathtt{J})a(\overrightarrow{M}).p'+p$.

4. Assume $p \equiv \partial(p_1)$. It follows directly from the axioms in Table 6 that:

- if $p_1 \equiv (\mathtt{J})!a(M); p_2$ or $p_1 \equiv (\mathtt{J})?a(M); p_2$ then we have $p \equiv \partial(p_1) \equiv 0$ due to Axiom $D1$ or Axiom $D2$. This is also not possible, since the latter term does not afford any transition.

- if $p_1 \equiv (\mathtt{J})a(M); p_2$ then we have $p \equiv \partial(p_1) \equiv (\mathtt{J})a(M); p'$ where $p' = \partial(p_2)$ (due to Axiom $D3$). The proof of this case is similar to the case 3.

- if $p_1 \equiv p_2 + p_3$ then we have $p \equiv \partial(p_1) \equiv p'_2 + p'_3$ where $p'_2 = \partial(p_2)$ and $p'_3 = \partial(p_3)$ (due to Axiom $D4$). Similar to the case 2, this case can be proven.

□

4.2 Ground-Completeness

We prove that our axiomatization is ground-complete for the term algebra of $CryptoPAi$ with finite-state models, modulo stateless pattern bisimularity.

Theorem 25. *The axiomatization of the process theory CryptoPAi is ground-complete for the term algebra of CryptoPAi modulo $\sim_{sl}$, i.e., for any closed CryptoPAi-terms p and q, $p \sim_{sl} q$ implies $CryptoPAi \vdash p = q$.*

Proof. Suppose that $\mathbb{P}(CryptoPAi)/_{sl} \models p = q$, i.e., $p \sim_{sl} q$. It must be shown that $CryptoPAi \vdash p = q$. It suffices to prove that, for all closed $CryptoPAi$-terms p and q:

(**i**) $p + q \sim_{sl} q$ implies $CryptoPAi \vdash p + q = q$,

(**ii**) $p \sim_{sl} p + q$ implies $CryptoPAi \vdash p = p + q$.

Assuming $p \sim_{sl} q$, from reflexivity and congruence of $\sim_{sl}$, we obtain $p+q \sim_{sl} p+p$ and $q + q \sim_{sl} p + q$. It follows from the soundness of Axiom $A2$ (idempotence of $+$) and the latter statements that $p + q \sim_{sl} p$ and $q \sim_{sl} p + q$. Then, from (**i**) and (**ii**), we have that $CryptoPAi \vdash p + q = p$ and $CryptoPAi \vdash q = p + q$.

From symmetry and transitivity of $=$, we obtain $CryptoPAi \vdash p = q$ and hence, ground completeness follows.

Property (**i**) is proven by induction on the total number of symbols (counting constants and action-prefix operators) in closed terms p and q as follows. The proof of property (**ii**) is similar and therefore skipped. The induction proof goes as follows. Assume $p + q \sim_{sl} q$ for some closed $CryptoPAi$-terms p and q. The base case of the induction corresponds to the case that p and q are both 0. $CryptoPAi \vdash 0 + 0 = 0$ is trivially followed using Axiom $A1$, proving the base case. The proof of the inductive step consists of a case analysis based on the structure of term p.

1. Assume $p \equiv 0$. It follows directly from the axioms in Table 6 that $CryptoPAi \vdash p + q = 0 + q = q + 0 = q$.

2. Assume $p \equiv p_1 + p_2$ for some closed terms p_1 and p_2. As $(p_1 + p_2) + q \sim_{sl} q$, by Lemma 23, $p_1 + q \sim_{sl} q$ and $p_2 + q \sim_{sl} q$. Thus, by induction, $CryptoPAi \vdash p_1 + q = q$ and $CryptoPAi \vdash p_2 + q = q$. Combining these results gives $CryptoPAi \vdash p + q = (p_1 + p_2) + q = p_1 + (p_2 + q) = p_1 + q = q$.

3. Assume $p \equiv (\mathtt{J})a(\overrightarrow{M}).p'$ for some $a \in \mathcal{A}ct$, closed term p', $\mathtt{J} \subseteq \mathcal{I}d$, and pattern-equivalent terms $M, M' \in \mathcal{T}erm$. Then, $(p, \Gamma_p) \overset{(\mathtt{J})a(\overrightarrow{M})}{\Rightarrow} (p', \Gamma'_p)$ and thus $(p + q, \Gamma_p) \overset{(\mathtt{J})a(\overrightarrow{M})}{\Rightarrow} (p', \Gamma')$ by applying the rule (**n0**). As $p + q \sim_{sl} q$, we have $(q, \Gamma_q) \overset{(\mathtt{J})a(\overrightarrow{M'})}{\Rightarrow} (q', \Gamma'_q)$ for some closed term q' such that $p' \sim_{sl} q'$, Γ' and Γ'_q are indistinguishable, and M and M' are pattern-equivalent. By Lemma 24, we have $CryptoPAi \vdash q = (\mathtt{J})a(\overrightarrow{M'}).q' + q$. From $p' \sim_{sl} q'$, as shown above, it follows that $p' + q' \sim_{sl} q'$ and $q' + p' \sim_{sl} p'$ and hence, by induction, $CryptoPAi \vdash p' + q' = q'$ and $CryptoPAi \vdash q' + p' = p'$. Combining these last two results gives $CryptoPAi \vdash p' = q' + p' = p' + q' = q'$ (i.e., $CryptoPAi \vdash p' = q'$). Similarly, it can be shown that $CryptoPAi \vdash (\mathtt{J})a(\overrightarrow{M})).p' = (\mathtt{J})a(\overrightarrow{M'})).q'$. Finally, $CryptoPAi \vdash p + q = (\mathtt{J})a(\overrightarrow{M})).p' + q = (\mathtt{J})a(\overrightarrow{M'})).q' + q = q$, which completes the proof of this case (i.e., $CryptoPAi \vdash p + q = q$).

4. Assume $p \equiv \partial(p_1)$. It follows directly from the axioms in Table 6 that:

- if $p_1 \equiv (\mathtt{J})!a(M); p_2$ or $p_1 \equiv (\mathtt{J})?a(M); p_2$ then we have $p \equiv \partial(p_1) \equiv 0$ due to Axiom $D1$ or Axiom $D2$. Consequently, the proof of this case is straightforward as the proof of 1.

- if $p_1 \equiv (\mathtt{J})a(M); p_2$ then we have $p \equiv \partial(p_1) \equiv (\mathtt{J})a(M); p'$ where $p' = \partial(p_2)$ (due to Axiom $D3$). The proof of this case is similar to the case 3.

$CM1$	$p \,\|\|\, q = p \lfloor\!\lfloor q + q \lfloor\!\lfloor p + p\|q$		$CM2$	$d \lfloor\!\lfloor p = d; p$	
$CM3$	$d; p \lfloor\!\lfloor q = d; (p \,\|\|\, q)$		$CM4$	$(p + q) \lfloor\!\lfloor r = p \lfloor\!\lfloor r + q \lfloor\!\lfloor r$	
$CM5$	$d; p\|d' = (d\|d'); p$	if $d\|d' \neq 0$	$CM6$	$d\|d'; p = (d\|d'); p$	if $d\|d' \neq 0$
$CM7$	$d; p\|d'; q = (d\|d'); (p \,\|\|\, q)$	if $d\|d' \neq 0$	$CM8$	$d; p\|d' = 0$	if $d\|d' = 0$
$CM9$	$d\|d'; p = 0$	if $d\|d' = 0$	$CM10$	$d; p\|d'; q = 0$	if $d\|d' = 0$
$CM11$	$(p + q)\|r = (p\|r) + (q\|r)$		$CM12$	$p\|(q + r) = (p\|q) + (p\|r)$	
$C1$	$p\|q = q\|p$		$C2$	$(p\|q)\|r = p\|(q\|r)$	
$C3$	$p\|0 = 0$				
$P1$	$(p \,\|\|\, q) \,\|\|\, r = p \,\|\|(q \,\|\|\, r)$		$P2$	$p \,\|\|\, q = q \,\|\|\, p$	
$P3$	$(p \lfloor\!\lfloor q) \lfloor\!\lfloor r = p \lfloor\!\lfloor (q \,\|\|\, r)$		$P4$	$(p\|q) \lfloor\!\lfloor r = p\|(q \lfloor\!\lfloor r)$	

Figure 8: Extension of the Axiomatization of *CryptoPAi* Terms with Parallel Activities

- if $p_1 \equiv p_2 + p_3$ then we have $p \equiv \partial(p_1) \equiv p'_2 + p'_3$ where $p'_2 = \partial(p_2)$ and $p'_3 = \partial(p_3)$ (due to Axiom $D4$). Similar to the case 2, this case can be proven.

□

5 Parallel and Communicating Processes

In this section, we extend our axiomatization to the full set of *CryptoPAi*-terms. Our approach is based on the auxiliary operators proposed first in [Bergstra and Klop 1984]. We extend our axiomatization system with the axioms given in Figure 8. The parallel composition of two processes p and q is denoted by $p \,||\, q$. To axiomatize parallel composition, we use two auxiliary operators, namely, $\lfloor\!\lfloor$ and $|$, respectively, denoting the left-merge and communication-merge operators. It is assumed that $\lfloor\!\lfloor$ and $|$ have the same binding priority as $||$ (i.e., they bind stronger than choice and weaker than action prefix). Also, since the operators $||$ and $|$ are commutative, the order in which the components are presented is irrelevant. The axioms $CM1-CM12$, $C1-C3$, and $P1-P4$ serve to rewrite each closed term involving parallel composition operators into a closed *CryptoPAi*-term. In these axioms, $d|d'$ is the result of the communication of a send and a receive, i.e., $(\mathtt{I})!a(\overrightarrow{M}) \mid (\mathtt{J})?a(\overrightarrow{M}) = (\mathtt{I} \cup \mathtt{J})a(\overrightarrow{M})$ and $(\mathtt{I})?a(\overrightarrow{M}) \mid (\mathtt{J})!a(\overrightarrow{M}) = (\mathtt{I} \cup \mathtt{J})a(\overrightarrow{M})$; otherwise, if d and d' are not matching send and receive actions, $d|d'$ is defined to be 0.

$$p \,||\, q = p \lfloor\!\lfloor q + q \lfloor\!\lfloor p + p|q.$$

The extension of the process theory *CryptoPAi* with the parallel operator has enlarged the set of closed terms that can be used for specifying processes. It is important to address the question whether this extension (called $CryptoPAi_{PA}$) has also enlarged the set of processes that can be specified in our processes theory (i.e., it must be investigated whether the expressiveness of our process theory has increased). We show this is not the case and the set of processes which can be specified is the same, but in more syntactically different ways than before. A common and standard proof technique is based on an elimination theorem [Baeten et al. 2009]. Such a theorem states that any process term, given in the extension of our process theory, can be rewritten into a basic term. The elimination theorem, stated below, expresses that every new closed-term is derivably equal to a basic closed $CryptoPAi$ term.

Theorem 26 Elimination. *For any closed $CryptoPAi_{PA}$ term p, there exists a basic term t s.t. $p = t$ can be derived from the axioms of $CryptoPAi_{PA}$ (i.e., $CryptoPAi_{PA} \vdash p = t$) by eliminating parallel composition using the axioms of Figure 8 as rewrite rules from left to right.*

Proof. Note that our axioms are special instances of generic axioms for concurrency stated in [Baeten et al. 2009] (by taking the communication function to be defined as specified above on matching send and receive actions and undefined otherwise). Hence, the normalization and confluence results of [Baeten et al. 2009] carry over to our axiomatization and elimination follows. □

6 Logical Aspects

In this section, we aim to discuss on logical aspects of our process theory. To this end, we first introduce an epistemic extension of the modal μ-calculus. Then, we study the issue of the logically equivalency and its correspondence with our bisimilarity notion. Then the characteristic theorem for our bisimulation notion in terms of EHML logic (see Section 6.1) is introduced and discussed.

6.1 The epistemic HM-Logic

EHML is an extension of Hennessy-Milner logic with an epistemic construct: K, representing epistemic knowledge. Epistemic knowledge refers to the facts that are inferred by principals using the combination of the (locally) observed computation and the knowledge of the protocol (i.e., the knowledge of other possible computations). The syntax of this logic has the following grammar.

$$\phi ::= \top \mid \phi \wedge \phi \mid \neg\phi \mid \langle a \rangle \phi \mid K_i \phi$$

where $i \in \mathcal{I}d$, and a is an action. $\langle a \rangle \phi$ means that "after some a transitions ϕ holds"; $K_i\phi$ indicates that "principal i knows that ϕ holds".

$\mathrm{E}, s \models \top$	iff	true
$\mathrm{E}, s \models \phi_0 \wedge \phi_1$	iff	$\mathrm{E}, s \models \phi_j$ for each $j \in \{0, 1\}$
$\mathrm{E}, s \models \neg\phi$	iff	$\mathrm{E}, s \models \phi$ is not true
$\mathrm{E}, s \models \langle d\rangle\phi$	iff	there is an $s' \in S$ and $d' \in D$ s.t. $\forall_{i\in\mathcal{I}d} Appear^i(d) = Appear^i(d')$ and $s \xrightarrow{d'} s'$ and $\mathrm{E}, s' \models \phi$
$\mathrm{E}, s \models K_i\phi$	iff	for all reachable $s' \in S$ such that $s \overset{i}{\cdots} s'$: $\mathrm{E}, s' \models \phi$

Figure 9: Satisfaction Relation of our Logic.

EHML-forms denotes the set of EHML formulae. In the following definition, the satisfaction of a formula $\phi \in$ EHML-forms in the ELTS E is interpreted.

Definition 27 Satisfaction. Let E be an ELTS as $\mathrm{E} = \langle S, \rightarrow, \checkmark, I_C, s_0\rangle$ and $s \in S$ be a state of E. The satisfaction relation $\models$ for formulae $\phi \in$ EHML-forms is defined inductively in Figure 9: E satisfies a formula ϕ, denoted $\mathrm{E} \models \phi$, if $s_0 \models \phi$.

Most of the definitions given in Figure 9 are straightforward. Of particular interest is the definition concerning the knowledge operator $K_i\phi$. It expresses the fact that i knows that ϕ if ϕ holds in all states reachable from s through the $\overset{i}{\cdots}$ relation. This relation was defined based on what i was allowed to observe and relating it to all computations that are indistinguishable to i.

Definition 28 Image finite process. A process p is image finite if and only if the collection $\{p' \mid p \xrightarrow{\alpha} p'\}$ is finite for each action α. An LTS is image finite if so in each of its states [Ben-Menachem 2010].

Note that the ELTS associated to a *CryptoPAi* process is by construction image finite, because we do not have any syntactic construct for generating infinitely many choices.

6.2 Logical Equivalence

In this section, we show that the EHML logics is the precise logical characterization of our stateless pattern bisimulation.

Theorem 29. *For each two processes $p, q \in \mathcal{P}roc$ and each two indistinguishable computations Γ and Γ' that are consistent with p and q, respectively, the following statement holds:*

$$p \sim_{sl} q \quad \textit{iff} \quad \forall_{\phi\in\mathrm{EHML}} E, (p, \Gamma) \models \phi \iff E, (q, \Gamma') \models \phi.$$

Proof. Below, we split the bi-implication into two implications and prove them separately:

1. Assume that $p \sim_{sl} q$ and $p \models \phi$ for some formulae $\phi \in$ EHML. Using structural induction on ϕ, we prove that $q \models \phi$. By symmetry, this is enough to establish that p and q satisfy the same formulae in EHML.

The proof proceeds by a case analysis on the form of ϕ.

- If ϕ is of the form $\top$, then the theorem holds vacuously because each process, including q, satisfies $\top$.

- If ϕ is of the form $\phi_0 \wedge \phi_1$ the theorem holds by the induction hypothesis: it follows from $E, (p, \Gamma) \models \phi$ and the semantics of EHML that $E, (p, \Gamma) \models \phi_0$ and $E, (p, \Gamma) \models \phi_1$ and from the induction hypothesis, we obtain $E, (q, \Gamma') \models \phi_0$ and $E, (q, \Gamma') \models \phi_1$. Using the semantics of EHML, we obtain $E, (q, \Gamma') \models \phi$.

- If ϕ is of the form $\neg\phi'$, then the theorem holds by the induction hypothesis, using a similar line of reasoning as above.

- If ϕ is of the form $\phi = \langle d \rangle \phi'$ for some action d and formula ϕ', then, there exists a state (p_0, Γ_0) and a decorated action d' s.t. $(p, \Gamma) \xrightarrow{d'} (p_0, \Gamma_0)$, $Appear^i(d) = Appear^i(d')$ and $(p_0, \Gamma_0) \models \phi'$. It follows from $p \sim_{sl} q$, that $(q, \Gamma') \xrightarrow{d''} (q_0, \Gamma_0')$ for some q_0', Γ_0' and d'' such that $\Gamma_0 \stackrel{i}{=} \Gamma_0'$ and $Appear^i(d') = Appear^i(d'')$, for each $i \in \mathcal{I}d$ and moreover, $p_0 \sim_{sl} q_0$. It follows from the latter bisimilarity $(p_0, \Gamma_0) \models \phi'$, $\Gamma_0 \stackrel{i}{=} \Gamma_0'$ and the induction hypothesis that $(q_0, \Gamma_0') \models \phi'$. $Appear^i(d) = Appear^i(d') = Appear^i(d'')$, $(q_0, \Gamma_0') \models \phi'$, and $(q, \Gamma') \xrightarrow{d''} (q_0, \Gamma_0')$ follow that necessarily $(q, \Gamma') \models \langle d \rangle \phi'$ or $(q, \Gamma') \models \phi$, which was to be shown.

- If ϕ is of the form $K_i\phi'$, for some $i \in \mathcal{I}d$ and $\phi' \in$ EHML, then the reasoning is similar to the above, using the induction hypothesis and bisimilarity of p and q.

2. Assume that p and $q \in \mathcal{P}roc$ satisfy the same formulae in EHML. We shall prove that $p \sim_{sl} q$. To this end, it is sufficient to show that the following relation is a stateless pattern bisimulation, for any $p, q \in \mathcal{P}roc$ and indistinguishable computations $\Gamma, \Gamma' \in \mathcal{C}omp$ that are consistent with p and q, respectively.

$$R = \{(p, q) \mid \ \forall_{\phi \in \text{EHML}} E, (p, \Gamma) \models \phi \Leftrightarrow E, (q, \Gamma') \models \phi\},$$

Take a pair $(p, q) \in R$; we aim at showing that p and q satisfy the transfer conditions of stateless pattern bisimilarity. We do so by induction on the length of the maximum trace afforded by p. Take two computations $\mathit{\Gamma}$ and $\mathit{\Gamma}'$, respectively, consistent with p and q.

The induction basis is when p does not afford any transition; then either q does not afford any transition, by which the theorem follows, or q does afford some d-labeled transition. In the latter case $E,(q,\epsilon) \models \langle d\rangle\top$, while the same formula is not satisfied by $E,(p,\epsilon)$. This, in turn, contradicts with the assumption that p and q satisfy the same set of EHML formulae.

For the induction step, assume that $(p,\mathit{\Gamma}) \xrightarrow{d} (p',\mathit{\Gamma}_0')$, and assume (towards a contradiction) that $(q,\mathit{\Gamma}')$ does not satisfy the transfer condition of stateless pattern bisimilarity regarding the transition labeled d, which can be due to one of the following two reasons:

- Either $(q,\mathit{\Gamma}')$ does not afford any d'-labeled transition, for any d' such that $Appear^i(d) = Appear^i(d')$. In this case, $(p,\mathit{\Gamma})$ satisfies the formula $\langle d\rangle\top$, while $(q,\mathit{\Gamma}')$ does not satisfy the same formula, which contradicts the assumption that $(p,\mathit{\Gamma})$ and $(q,\mathit{\Gamma}')$ satisfy the same set of EHML formulae.

- Or for each such d'-labeled transition of the form $(q,\mathit{\Gamma}') \xrightarrow{d_i} (q_i',\mathit{\Gamma}_i')$, p' is not stateless pattern bisimilar to q_i'. It follows from the induction hypothesis that for such q_i', there exists a formula ϕ_i such that ϕ_i is not satisfied by p, while it is satisfied by q (or vice versa, in which $\neg\phi_i$, has the property we seek). Since E is image finite, the number of such d'-labeled transitions is finite and so is the number of ϕ_i; let I be the finite set gathering all such i's. Hence $\wedge_{i\in I}\langle d_i\rangle\phi_i$ is not satisfied by p, but it is satisfied by q, which contradicts our assumption that p and q satisfy the same set of EHML formulae and hence, concludes the proof. □

7 Conclusions

In this paper, we defined a number of behavioral equivalences for an extension of process algebra with a notion of appearance for actions, which makes it suitable for reasoning about epistemic aspects of security protocols. We also presented some formal results regarding each of the defined notion and formulated and proved a congruence result for the most robust notion, called stateless pattern bisimilarity. Then, a sound and ground-complete axiomatization for *CryptoPAi* terms modulo stateless pattern bisimilarity was given. Finally, an extension of the Hennessy-Milner logic with an epistemic knowledge construct was presented and was shown that it characterizes our notion of stateless pattern bisimilarity. As demonstrated in [Mahrooghi and Mousavi 2011], a prototype model-checker for *CryptoPAi* has been provided in the Maude rewriting logic tool to model-check the specified properties on their corresponding models. We plan to further investigate the applicability of this prototype in the domain of e-voting protocols by applying it to more case studies. Extending *CryptoPAi* with probabilities is another avenue for future research.

Acknowledgements

We would like to express our sincerest thanks to Professor Mohammad Reza Mousavi, for his comments, and many insightful discussions. His contributions greatly improved this paper. We also want to thank Professor Mohammad Izadi, for reading and commenting on this paper.

References

[Abadi and Jürjens 2001] Abadi, M., Jürjens, J., 2001. Formal eavesdropping and its computational interpretation. In: Proceedings of the 4th International Symposium on Theoretical Aspects of Computer Software. TACS '01. pp. 82–94.

[Abadi and Rogaway 2002] Abadi, M., Rogaway, P., 2002. Reconciling two views of cryptography (the computational soundness of formal encryption). Journal of Cryptology 15 (2), 103–127.

[Aceto and Ingólfsdóttir 2007] Aceto, L., Ingólfsdóttir, A., 2007. The saga of the axiomatization of parallel composition. In: Caires, L., Vasconcelos, V. T. (Eds.), Proceedings of the 18th International Conference on Concurrency Theory (CONCUR 2007). Vol. 4703 of Lecture Notes in Computer Science. Springer, pp. 2–16.

[Baeten 2005] Baeten, J. C. M., 2005. A brief history of process algebra. Theor. Comput. Sci. 335 (2-3), 131–146.

[Baeten et al. 2009] Baeten, J. C. M., Basten, T., Reniers, M. A., Dec. 2009. Process Algebra: Equational Theories of Communicating Processes (Cambridge Tracts in Theoretical Computer Science), 1st Edition. Cambridge University Press.

[Baeten and Bergstra 1997] Baeten, J. C. M., Bergstra, J. A., 1997. Process algebra with propositional signals. Theor. Comput. Sci. 177 (2), 381–405.

[Basten and Hooman 1999] Basten, T., Hooman, J., 1999. Process algebra in pvs. In: Proc. of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS '99), volume 1579 of Lecture Notes in Computer Science. Springer-Verlag, pp. 270–284.

[Ben-Menachem 2010] Ben-Menachem, M., July 2010. Reactive systems: Modelling, specification and verification; is written by l. aceto, et al; and published by cambridge university press; distributed by cambridge university press; © 2007, (hardback), isbn 978-0-521-87546-2, pp. 300. SIGSOFT Softw. Eng. Notes 35, 34–35.

[Bergstra and Klop 1984] Bergstra, J. A., Klop, J. W., 1984. Process algebra for synchronous communication. Information and Control 60 (1-3), 109–137.

[Bergstra and Klop 1987] Bergstra, J. A., Klop, J. W., 1987. Acp - A Universal Axiom System For Process Specification Centre for Mathematics and Computer Science. Newsletter Nr: 15 pages: 3–23.

[Borgström et al. 2006] Borgström, J., Grinchtein, O., Kramer, S., 2006. Timed Calculus of Cryptographic Communication. In: Proceedings of the Workshop on Formal Aspects in Security and Trust. Vol. 4691 of Lecture Notes in Computer Science. pp. 16–30.

[Borgström et al. 2006] Borgström, J., Kramer, S., Nestmann, U., 2006. Calculus of cryptographic communication. In: Proceedings of FCS-ARSPA.

[de Boer et al. 2003] de Boer, F. S., van Eijk, R. M., van der Hoek, W., Meyer, J.-J. C., 2003. A fully abstract model for the exchange of information in multi-agent systems. Theor. Comput. Sci. 290 (3), 1753–1773.

[Dechesne et al. 2007] Dechesne, F., Mousavi, M., Orzan, S., 2007. Operational and epistemic approaches to protocol analysis: bridging the gap. In: Proceedings of the 14th international conference on Logic for programming, artificial intelligence and reasoning, LPAR'07. Springer-Verlag, pp. 226–241.

[Eijk et al. 2003] Eijk, R. M. v., Boer, F. d., Hoek, W. v., Meyer, J.-J. C., 2003. Process algebra for agent communication: A general semantic approach. In: Communication in Multiagent Systems - Agent Communication Languages and Conversation Policies. Vol. 2650 of Lecture Notes in Computer Science. Springer, pp. 113–128.

[Groote and Ponse 1994] Groote, J. F., Ponse, A., 1994. Process algebra with guards: Combining hoare logic with process algebra. Formal Asp. Comput. 6 (2), 115–164.

[Hennessy and Milner 1985] Hennessy, M., Milner, R., January 1985. Algebraic laws for nondeterminism and concurrency. J. ACM 32, 137–161.

[Hoare 1985] Hoare, C. A. R., 1985. Communicating Sequential Processes. Prentice-Hall.

[Laud and Corin 2003] Laud, P., Corin, R., 2003. Sound computational interpretation of formal encryption with composed keys. In: In Information Security and Cryptology - ICISC 2003, 6th International Conference, LNCS. Springer-Verlag, pp. 55–66.

[Mahrooghi and Mousavi 2011] Mahrooghi, H. R., Mousavi, M., 2011. Reconciling operational and epistemic approaches to the formal analysis of crypto-based security protocols. In: Proceedings of the 9th International Workshop on Security Issues in Concurrency (SecCo'11).

[Milner 1980] Milner, R., 1980. A Calculus of Communicating Systems. Vol. 92 of Lecture Notes in Computer Science. Springer.

[Mousavi et al. 2004] Mousavi, M. R., Reniers, M., Groote, J. F., 2004. Congruence for SOS with data. In: Proceedings of Nineteenth Annual IEEE Symposium on Logic in Computer Science (LICS'04). IEEE Computer Society Press, Turku, Finland, pp. 302–313.

[Mousavi et al. 2003] Mousavi, M. R., Reniers, M. A., Basten, T., Chaudron, M. R. V., 2003. Separation of concerns in the formal design of real-time shared data-space systems. In: ACSD. pp. 71–81.

[Mousavi et al. 2005] Mousavi, M. R., Reniers, M. A., Groote, J. F., 2005. Notions of bisimulation and congruence formats for sos with data. Inf. Comput. 200 (1), 107–147.

[Plotkin 2004] Plotkin, G. D., 2004. A structural approach to operational semantics. Journal of Logic and Algebraic Progamming 60, 17–139.