Sharing Beliefs about Actions: A Parallel Composition Operator for Epistemic Programs

Penelope Economou
Oxford University
Computing Laboratory
Oxford, OX1 3QD, UK
penelope.economou@comlab.ox.ac.uk

May 16, 2005

Abstract

We introduce a new operator on epistemic programs expressing the action of sharing a belief about the current action inside a subgroup. This is a sort of "epistemic parallel composition" of programs. This can be used to define many useful epistemic programs and operators encountered in the related literature. We propose an equational calculus for this operator and we give two examples of its usage.

1 Introduction

One of the major problems in the field of distributed systems has to do with *modelling and reasoning about information flow and information exchange between 'agents'*. There are many issues connected to this problem which has many applications in *communication protocols* where one has to check for secrecy and authentication, or in *Artificial Intelligence* where agents need to be equipped with the necessary tools to reason about the changes in the environment and about each other, etc.

Traditionally there have been *two standard approaches* to information flow in multi-agent systems: *modal logic approaches*, that is, combinations of *dynamic and epistemic* logic approaches, and *equational* approaches based on process algebras (CCS, CSP, ACP etc.). Dynamic Epistemic Logic (**DEL**) is a **PDL**- style logic in which it is possible to describe both the effect of epistemic actions on the knowledge of an agent in a multi-agent system, and the effect that knowledge has on the epistemic actions performed by an agent in such a system. A general notion of epistemic programs for (**DEL**) was introduced in [1,3,4], where the ideas and techniques presented are related to the work by Plazza [8], Groeneveld

[7], and Gerbrandy [6]. All authors deal with subjects arising from the work of [5].

We built on the ideas presented in [1,2,3,4]. More concretely, we introduce a new operator for epistemic programs expressing the action of sharing a belief about the current action inside a subgroup. This is a sort of "epistemic parallel composition" of programs, which can be used to define many useful epistemic programs and operators encountered in the related literature. We propose an equational calculus for this operator and we give both a simple and a more complicated example of its usage by modeling a variant of the well-known cryptographic attack Man-in-the-middle (MITM).

2 An Epistemic Scenario

Perhaps the best way to enter to enter our overall subject is by considering an epistemic scenario. First, we introduce some preliminary notions.
Preliminary Notions. Given a set Φ of atomic propositions and a set of agents $\mathcal{A}$, an *(epistemic) state model* is a Kripke model $\mathbf{S} = (S, \xrightarrow{A}, \| \cdot \|)_{A \in \mathcal{A}}$, where the members of the universe represent *states* and the atomic sentences non-epistemic, 'objective' *facts* of the world, which can be thought of as properties of states. The valuation $\| \cdot \|$ tells us which facts hold at which states. Finally, the accessibility relations model the agents' epistemic uncertainty about the current state.

Epistemic relevant properties of states in (epistemic) state models are given by *epistemic propositions.* Let **SMod** be the collection of all state models. An *epistemic proposition* is an operation $\boldsymbol{\varphi}$ defined on **SMod** such that for all $\mathbf{S} \in \mathbf{SMod}$, $\boldsymbol{\varphi}_{\mathbf{S}} \subseteq S$. If $s \in \boldsymbol{\varphi}_{\mathbf{S}}$, we say that *state* $s \in \mathbf{S}$ *satisfies proposition* $\boldsymbol{\varphi}$, or that $\boldsymbol{\varphi}$ is *true* at state s in model $\mathbf{S}$.

Epistemic actions (or programs) are (possibly complex) actions preformed by agents in order to *change* their information states. We explicitly formalize the underlying epistemic structure of an epistemic program by representing it as an epistemic action model.

An *(epistemic) action model* is a Kripke model $\mathbf{\Sigma} = (\Sigma, \xrightarrow{A}, \mathsf{pre})_{A \in \mathcal{A}}$ where Σ is a set of *simple actions*, $\xrightarrow{A}$ is an A-indexed family of relations on Σ, and $\mathsf{pre} : \Sigma \to \mathcal{P}(\mathbf{\Phi})$. Simple actions are meant to represent particularly simple kinds of actions, that is, *deterministic* actions. Simple actions have *preconditions* each of whom is an epistemic proposition, thus associating to each action $\sigma \in \Sigma$ a set $\mathsf{pre}(\sigma)$ of epistemic propositions. An action can happen only in states where all the preconditions in $\mathsf{pre}(\sigma)$ are satisfied.

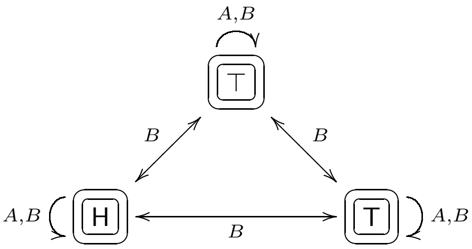
An *epistemic program* is defined as a pair $\pi = (\mathbf{\Sigma}, \Gamma)$ consisting of an action model $\mathbf{\Sigma}$ and a set $\Gamma = |\pi|$ of *designated simple actions.* Each of the simple actions $\gamma \in |\pi|$ can be thought as being a possible 'deterministic resolution' of the non-deterministic program π. Epistemic programs can alternatively be presented by having maps: $\pi_A : S \to \mathcal{P}(S)$, $\sigma \mapsto (\sigma)_A \subseteq S$, for every agent $A \in \mathcal{A}$, instead of the arrows. If we are given the arrows $\overset{A}{\to}$, we can define the *appearance*[1]: $\sigma_A = \{\sigma' : \sigma \xrightarrow{A} \sigma'\}$ to each agent A. ⊣

The Epistemic Scenario. Traditional epistemic scenarios often deal with *changes of knowledge* that come about in various ways. The standard treatment of these matters attempts to formalize the informal notions of *knowledge* and *common knowledge*, i.e. *justified true belief.* Nonetheless, in many contexts agents are deceived by certain actions without loosing their rationality; therefore, we can consider more generally *justifiable beliefs* weather they happen to be true or not. Our new operation-to be introduced shortly-claims to formalize some of these beliefs and the justifiable changes affecting them.

In what follows we consider an epistemic scenario. This will give the reader some idea of the motivation for introducing a new operator on epistemic programs.

Scenario 1: Pick a card. A and B enter a large room containing a remote-control mechanical coin flipper. One presses the button and the coin spins through the air and then lands in a small box on a table. A card is shown to A, in the presence of B, which either says heads (H), tails (T), or is blank. In the first two cases the card describes truly the state of the coin in the box, and in the last case the intention is that no information is given.

The epistemic program describing the above scenario has the following representation



where we draw simple actions labeled with their preconditions, and epistemic uncertainty relations as arrows; we use double circles for the designated ('currently possible') action(s). We call the top arc μ, the one to the left σ, and the one to the right ρ.

Here is a justification of our model: either of σ, ρ, and μ can *really* happen: σ, and ρ are the actions where A learns the *true* side of the coin, while B makes a public announcement he believes A either learns the side of the coin or she learns nothing, and μ is the action where A learns nothing while B makes a similar announcement. Both A and B are aware of the possibility of these actions. In fact, A has no uncertainty about the current action taking place: in each case, she considers the current action (σ, ρ, or μ) as the *only* alternative. On the other hand, B has uncertainty about the current action taking place: he cannot distinguish between σ, ρ, or μ taking place.

Our goal is to propose a theoretical understanding of the above representation. In the following sections we attempt to give an answer to the question:

[1] The appearance sets express the agents beliefs about the very action that is taking place. Thus, $\sigma' \in \sigma_A$ represents the actions that an agent considers as 'alternatives' of the 'real' action.

‘How are we going to model this epistemic action?’ In the following section we introduce a new operation that models the beliefs of the agents about the current action taking place and the justifiable changes affecting these beliefs. Then, we introduce and discuss our logic after which we revisit the example scenario and show how it can be modeled. We further demonstrate the usage of our new operation by examining a more complicated epistemic scenario.

3 A New Operation on Epistemic Programs

A New Operation. Sharing Beliefs: While π, $\mathcal{B}$ announces π' to $\mathcal{C}$. Given a group of agents $\mathcal{B}$, $\mathcal{C} \subseteq \mathcal{A} = \{A, B \dots\}$, and epistemic programs π, and π', we introduce a new operation $\pi \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi'$, to be read as : *while* π, $\mathcal{B}$ *announces* π' *to* $\mathcal{C}$. This operation represents $\mathcal{B}$'s (sincere) *belief* about what is happening. In other words: while π is happening (the members of group) $\mathcal{B}$ *share* with $\mathcal{C}$ their *belief* that π' is happening. It represents a kind of *parallel composition*: program π is happening in parallel with a *sincere, private*, (but *not necessarily truthful*) announcement (from $\mathcal{B}$ to $\mathcal{C}$) that the program π' is happening. In other words, while π is happening (all members of) $\mathcal{B}$ believe that π' is happening instead, and moreover, they *communicate* this belief to (all members) of $\mathcal{C}$ by sending (them) a message over a fully private, secure and reliable channel. And as $\mathcal{B}$'s belief that π' is taking place is *shared* with $\mathcal{C}$, after is announced it becomes *common knowledge* among (the agents of) $\mathcal{B} \cup \mathcal{C}$ that $\mathcal{B}$ believes π' has been taking place. However, the beliefs of the agents in $\mathcal{C}$ are not changed by $\mathcal{B}$'s announcement, i.e. they don't assume that $\mathcal{B}$'s beliefs are truthful, they are simply aware of them while at the same time they have their own beliefs about which actions is currently happening.

Towards giving an abstract definition of the new operation we discuss the intuitive meaning of the preconditions and the appearance maps. We begin with the preconditions: $\pi \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi'$ can happen only when π can happen; after all, the ‘real’ program taking place is π. We move on to the appearance maps: $\pi \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi'$ appears to each member of $\mathcal{B}$ as if either of the designated actions of π' is taking place, while it announces to $\mathcal{C}$ that π'. As far as the members of $\mathcal{C}$, $\pi \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi'$ appears to (each of) them as if either of the actions they consider as alternatives to the current deterministic resolution of π is happening, while they are also aware of $\mathcal{B}$'s belief that π' is happening. Finally, what the rest of the agents $C \notin \{\mathcal{B}, \mathcal{C}\}$ believe to be happening is: either of the actions they consider as alternatives to the current deterministic resolution of π. After all, $\mathcal{B}$'s announcement to $\mathcal{C}$ is *private*; hence, all agents outside $\{\mathcal{B}, \mathcal{C}\}$ are unaware of it happening. Formally,

$$\begin{array}{rcc}
|\,\pi \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi'\,| & = & \{\sigma \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi' : \sigma \in |\pi|\} \\
\mathsf{pre}(\sigma \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi') & =: & \mathsf{pre}(\sigma) \\
(\sigma \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi')_B & =: & \{\sigma' \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi' : \sigma' \in |\pi'|\},\ \text{for all}\, B \in \mathcal{B} \\
(\sigma \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi')_C & =: & \{\lambda \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi' : \lambda \in \sigma_C\},\ \text{for all}\, C \in \mathcal{C} \setminus \mathcal{B} \\
(\sigma \,!^{\mathcal{B}}_{\mathcal{C}}\, \pi')_D & =: & \sigma_D,\ \text{for all}\, D \notin \{\mathcal{B}, \mathcal{C}\}
\end{array}$$

4 Dynamic Epistemic Calculus (DEC)

4.1 The language of DEC

Definition 4.1.1 (The language of DEC). Given a set of *atomic propositions* $\mathbf{\Phi}_0$ whose elements are usually denoted by $\boldsymbol{p}$, $\boldsymbol{q}$, $\boldsymbol{r}$ and so on, the top and bottom element $\top$, and $\bot$, and a set of agents $\mathcal{B}, \mathcal{C} \subseteq \mathcal{A} = \{A, B \ldots\}$, the formal definition of the well-formed *basic programs* of the *language* of **DEC** is given by

$$\pi \quad ::= \quad ?\boldsymbol{p} \quad | \quad \pi + \pi' \quad | \quad \pi \cdot \pi' \quad | \quad \pi \,!^{\mathcal{B}}_{\mathcal{C}} \pi'$$

where $\boldsymbol{p}$ ranges over $\mathbf{\Phi}_0$.

Hence, a basic program of **DEC** is either a test of an atomic proposition as in PDL ($?\boldsymbol{p}$): a proposition $\boldsymbol{\varphi}$ is tested, without anybody noticing, or a non-deterministic sum of programs[2] ($\pi + \pi'$), or a sequential composition of programs[3] ($\pi \cdot \pi'$), or a sort of parallel composition of programs to be read as: *while* π*,* $\mathcal{B}$ *announces* π' *to* $\mathcal{C}$ ($\pi \,!^{\mathcal{B}}_{\mathcal{C}} \pi'$).

Remark 4.1.2. As already noticed, in our proposed syntax we have restricted the scope of the argument of tests to *atomic propositions.* Even though we can easily extend it to cover *epistemic propositions* as well, we chose to restrict it as to obtain a simpler axiomatic system for the purposed of this paper.

We chose to have the particular operations in the language of **DEC** based on the intuition that these are the natural operations on epistemic actions and most others can be defined in terms of them.

For example, one of the advantages of choosing $?\boldsymbol{p}$ as a basic program is that we can express the well-known programs *skip*, and *crash* in our language: *skip* $\stackrel{\text{def}}{=} ?\top$, and *crash* $\stackrel{\text{def}}{=} ?\bot$. We can also alternatively define the testing of an atomic proposition $\boldsymbol{p}$ by: $?\boldsymbol{p} \stackrel{\text{def}}{=} ?\boldsymbol{p} \,!^{\mathcal{B}}_{\mathcal{C}} ?\top$.

As far as $\pi \,!^{\mathcal{B}}_{\mathcal{C}} \pi'$ and according to the choice of π, π', $\mathcal{B}$, and $\mathcal{C}$, we get a variety of interesting programs.

Special case 1: $\pi \,!^{\mathcal{B}}_{\mathcal{A}} \pi' \stackrel{\text{def}}{=} \pi \,!^{\mathcal{B}} \pi'$, obtained by taking $\mathcal{C} = \mathcal{A}$. We call this: *public announcement of a program by subgroups.* The difference here has to do with the nature of $\mathcal{B}$'s announcement: by announcing to the set of *all* agents ($\mathcal{A}$) their beliefs, the members of (group) $\mathcal{B}$ are no longer making a *private* announcement but a *public* one.

Special case 2: $\pi \,!^{\mathcal{B}}_{\mathcal{B}} \pi' \stackrel{\text{def}}{=} \pi \,|^{\mathcal{B}} \pi'$, obtained by taking $\mathcal{B} = \mathcal{C}$. We read this: *while* π*,* $\mathcal{B}$ *thinks* π'. The intended meaning is that: while π is happening, (all members of) $\mathcal{B}$ *believe* that π' is happening, in which case it intuitively represents the generalized version (to a group of agents) of an operation defined by Baltag[4] [2].

[2] From now on, we will drop the word *basic* and simply speak of programs.

[3] It should be noted that $?\boldsymbol{p}$, $\pi + \pi'$, and $\pi \cdot \pi'$, can all be defined similarly to $\pi \,!^{\mathcal{B}}_{\mathcal{C}} \pi'$, by using appearance maps and factual contents, and by specifying their designated actions.

[4] The binary operation operation $\pi|^{A}\pi'$ which is a sort of parallel composition of program π with an updating action done by A, is understood as follows: *while* π*,* A *thinks* π'.

The following are all special cases of the above:

Special case 2a: ${}^{\mathcal{B}}\pi \stackrel{\text{def}}{=} \mathit{skip}\,|^{\mathcal{B}}\pi$. We call this: *gratuitous (i.e. mistaken) group updating with a program.* It is the operation where nothing happens except that all members of (group) $\mathcal{B}$ *simultaneously* start to believe that π is happening. It represents the generalized version of an operation Baltag [2] calls *gratuitous (i.e. mistaken) updating* with a program: ${}^{A}\pi \stackrel{\text{def}}{=} \mathit{skip}\,|^{A}\,\pi$.

Special case 2b: $\pi/\mathcal{B} \stackrel{\text{def}}{=} \pi\,|^{\mathcal{B}}\mathit{skip}$. We call this: *group hiding of a program.* The intended meaning of the operation is that: while π is happening all members of (group) $\mathcal{B}$ *simultaneously* start to believe that *nothing* is happening. It represents the generalized version of an operation Baltag [2] calls *hiding* of a program from an agent: $\pi/A \stackrel{\text{def}}{=} \pi\,|^{A}\,\mathit{skip}$.

Special case 2c: $\mathcal{L}^{\mathcal{B}}\pi \stackrel{\text{def}}{=} \pi\,|^{\mathcal{B}}\pi$. We call this: *group learning of a program.* In other words, while π is happening all members of (group) $\mathcal{B}$ think (i.e. *learn*) that π is happening. This operation can be seen as a generalized version of an operation Baltag [2] calls *learning of a program*: $\mathcal{L}^{A}\pi \stackrel{\text{def}}{=} \pi\,|^{A}\pi$.

Finally, by choosing $\pi\,!^{\mathcal{B}}_{\mathcal{C}}\pi'$ as a basic program we can express many well-known programs. First of all, we have $?\boldsymbol{p}\,!^{\mathcal{B}}_{\mathcal{B}}\,?\boldsymbol{p} \stackrel{\text{def}}{=} \boldsymbol{p}!_{\mathcal{B}}$, where $\boldsymbol{p}!_{\mathcal{B}}$ is the *totally private, truthful announcement* of an atomic proposition $\boldsymbol{p}$ to *only* to a subgroup $\mathcal{B} \subseteq \mathcal{A}$, while it is assumed that none of the 'outsiders' suspect anything. Alternatively, we can define the private announcement of an atomic proposition to subgroups as the group learning of the testing of such a proposition i.e. $\boldsymbol{p}!_{\mathcal{B}} \stackrel{\text{def}}{=} \mathcal{L}^{\mathcal{B}}(?\boldsymbol{p})$. Also, by taking $\mathcal{B}$ to be the set of all agents $\mathcal{A}$, we also get $?\boldsymbol{p}\,!^{\mathcal{A}}_{\mathcal{A}}\,?\boldsymbol{p} \stackrel{\text{def}}{=} \boldsymbol{p}!_{\mathcal{A}} \stackrel{\text{def}}{=} \boldsymbol{p}!$, where $\boldsymbol{p}!$ is the *public, truthful announcement* of an atomic proposition $\boldsymbol{p}$ to the set of all agents $\mathcal{A}$. Alternatively, we can define public announcements via the learning operator: $\boldsymbol{p}! \stackrel{\text{def}}{=} \mathcal{L}^{\mathcal{A}}(?\boldsymbol{p})$.

4.2 Axioms of DEC

We now formalize our intuitive understanding of the programs of the basic syntax of **DEC** by proposing some axioms for each of our operators. All axioms are easily checked to be sound.

As noticed in subsection 3.1, many useful programs can be defined in our calculus like *skip*, *crash*, *private announcement* (of an atomic proposition) *to subgroups* and *public announcement* of an atomic proposition, i.e. $\mathit{skip} \stackrel{\text{def}}{=} ?\top$, $\mathit{crash} \stackrel{\text{def}}{=} ?\bot$, $?\boldsymbol{p}\,!^{\mathcal{B}}_{\mathcal{B}}\,?\boldsymbol{p} \stackrel{\text{def}}{=} \boldsymbol{p}!_{\mathcal{B}}$ and $?\boldsymbol{p}\,!^{\mathcal{A}}_{\mathcal{A}}\,?\boldsymbol{p} \stackrel{\text{def}}{=} \boldsymbol{p}!_{\mathcal{A}} \stackrel{\text{def}}{=} \boldsymbol{p}!$. Moreover, we also defined the *group learning of a program*, the *gratuitous (i.e. mistaken) updating of a group with a program* and the *hiding of a program from a group of agents*, i.e. $\mathcal{L}^{\mathcal{B}}\pi \stackrel{\text{def}}{=} \pi\,|^{\mathcal{B}}\pi$, ${}^{\mathcal{B}}\pi \stackrel{\text{def}}{=} \mathit{skip}\,|^{\mathcal{B}}\pi$, and $\pi/\mathcal{B} \stackrel{\text{def}}{=} \pi\,|^{\mathcal{B}}\mathit{skip}$, where (from now on) $|^{\mathcal{B}} \stackrel{\text{def}}{=} !^{\mathcal{B}}_{\mathcal{B}}$.

We propose some natural axioms for each of our operators: $(+)$ is associative, commutative, idempotent, and and has *crash* as a neutral element; $(\cdot)$ is associative as well as right and left distributive over $(+)$. Also, a natural axiom for $(\cdot)$ is: $\mathit{skip} \cdot x = x \cdot \mathit{skip} = x$.

We continue with some natural axioms for ?

$$?\boldsymbol{p} \cdot ?\boldsymbol{p} = ?\boldsymbol{p} \tag{t1}$$

$$?\boldsymbol{p} \cdot x = x \cdot ?\boldsymbol{p} \tag{t2}$$

$$?\boldsymbol{p} = ?\boldsymbol{p} \,!^{\mathcal{B}}_{\mathcal{C}}\, \mathit{skip} \tag{t3}$$

$$?\boldsymbol{p} \cdot (?\boldsymbol{p} \,!^{\mathcal{B}}_{\mathcal{C}}\, x) = ?\boldsymbol{p} \,!^{\mathcal{B}}_{\mathcal{C}}\, x \tag{t4}$$

Finally, here are some natural axioms for $!^{\mathcal{B}}_{\mathcal{C}}$

$$(x + y) \,!^{\mathcal{B}}_{\mathcal{C}}\, z = x \,!^{\mathcal{B}}_{\mathcal{C}}\, z + y \,!^{\mathcal{B}}_{\mathcal{C}}\, z \tag{a1}$$

$$(x \,!^{\mathcal{B}}_{\mathcal{C}}\, y) \cdot (z \,!^{\mathcal{B}}_{\mathcal{C}}\, w) = ((x \,!^{\mathcal{B}}_{\mathcal{C}}\, y) \cdot z) \,!^{\mathcal{B}}_{\mathcal{C}}\, ((y \,!^{\mathcal{B}}_{\mathcal{C}}\, y) \cdot w) \tag{a2}$$

$$(x \,!^{\mathcal{B}}_{\mathcal{C}}\, y) \,!^{\mathcal{B}}_{\mathcal{C}}\, (z \,!^{\mathcal{B}}_{\mathcal{C}}\, w) = x \,!^{\mathcal{B}}_{\mathcal{C}}\, z \tag{a3}$$

$$\mathit{skip} \,!^{\mathcal{B}}_{\mathcal{C}}\, \mathit{skip} = \mathit{skip} \tag{a4}$$

5 The epistemic scenario revisited

Now that we have introduced our new operation we are ready to give the total action describing the epistemic scenario of Section 2: $\pi = \sigma + \rho + \mu$, where $\sigma = (\mathcal{L}^A ?\mathsf{H}) \,!^B (\mathcal{L}^A ?\mathsf{H} + \mathcal{L}^A ?\mathsf{T} + \mathit{skip})$, $\rho = (\mathcal{L}^A ?\mathsf{T}) \,!^B (\mathcal{L}^A ?\mathsf{H} + \mathcal{L}^A ?\mathsf{T} + \mathit{skip})$, and $\mu = \mathit{skip} \,!^B (\mathcal{L}^A ?\mathsf{H} + \mathcal{L}^A ?\mathsf{T} + \mathit{skip})$.

As π is non-deterministic, it can be resolved in any way σ, ρ, or μ are resolved: $|\pi| = \{\sigma\} \cup \{\rho\} \cup \{\mu\}$. Simple actions σ, ρ, and μ can happen if the card is showing heads, tails or nothing, respectively. Hence, it can be shown that $\mathsf{pre}(\sigma) = \mathsf{H}$, $\mathsf{pre}(\rho) = \mathsf{T}$, and $\mathsf{pre}(\mu) = \top$. As far as the appearance of σ, ρ, and μ to the agents it can be shown that for A: $(\sigma)_A = \sigma$, $(\rho)_A = \rho$, and $(\mu)_A = \mu$, and for B: $(\sigma)_B = (\rho)_B = (\mu)_B = \{\sigma\} \cup \{\rho\} \cup \{\mu\}$.

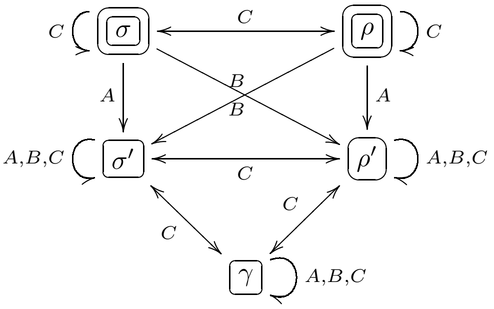
6 A cryptographic attack

We further demonstrate the usage of our new operator by examining a more complicated scenario which is a variant of the well-known cryptographic attack Man-in-the-middle (MITM). In cryptography, a man in the middle attack (MITM) is an attack in which an attacker is able to read, insert and modify at will, messages between two parties without either party knowing that the link between them has been compromised. The attacker must be able to observe and intercept messages going between the two victims.

Deciphering encrypted communications without knowing the proper keys is one of the major problems in cryptography today. Hence, the modeling of scenarios expressing such problems is both an interesting and a useful task. As it turns out epistemic operators introduced by Baltag, Gerbrandy, and van Ditmarsh are too simple or not enough to handle the MITH scenario. We claim that the epistemic program describing the variant of the MITM attack (to be described shortly) can be expressed in (the language of) **DEC** using the parallel composition operator introduced earlier.

Description of the attack. Two agents, A and B, share a secret key, so that they can send each other encrypted messages over some channel. But the channel is not secure: some outsider C may intercept the messages or prevent them from being delivered (although he cannot read them, or send around instead his own encrypted messages, since he doesn't have the key). Suppose also the *encryption method* is publicly known (although the key is secret). It is also known that A is the only one who knows some important *secret* (say, whether some *fact* $\boldsymbol{p}$ *holds or not*). Suppose now that A sends an encrypted message to B, communicating the secret (whether $\boldsymbol{p}$ or $\neg\boldsymbol{p}$). B gets the message, and he's convinced it must be authentic, since it has been encrypted with the secret key. To make sure B got the message, the protocol requires him to to broadcast over a public channel (but impossible to block or falsify) a message saying 'Yes, I got the encrypted message'. So both of them will be convinced that they now share the secret, and that C doesn't know the secret, although he may *suspect* they know it. (But they think C can't be sure of that either, since for all he knows the message might have been just junk). However, suppose that agent C is the only one to notice two features of the encryption method: first, that the shape of the encrypted message can show whether it contains a secret ($\boldsymbol{p}$ or $\neg\boldsymbol{p}$) or it's just junk; second, that without knowing the key or reading the content, he can modify the encrypted message in a trivial way, so that the encoded bit is changed to its opposite: the message will read $\boldsymbol{p}$ if it was $\neg\boldsymbol{p}$, and vice-versa. So the outsider C will secretly intercept the message, change it appropriately and send it to B. Without knowing the secret C has managed to manipulate his opponents' beliefs: A and B will *mistakenly believe that they now share the secret*; while in fact B got the 'wrong secret' instead!

Modeling of the attack. The epistemic program describing the above *cryptographic attack* (including the simultaneous sending of the secret by A, interception and manipulation by C, and receiving and acknowledgement by B) has the following representation:



where we draw simple actions labeled with their preconditions, and epistemic uncertainty relations as arrows; we use double circles for the designated ('currently possible') action(s).

We now give a justification for our action model: only one of the two actions on top (σ, and ρ) can *really* happen: these are actions in which the "secret" (either $\boldsymbol{p}$ or $\neg\boldsymbol{p}$) is intercepted, modified and resent to B. Only C is aware of the possibility of these actions, but he doesn't know which of them is happening: so there are C-arrows between these top nodes. The two nodes in the middle row (σ', and ρ') are possible actions that A or B may think to be happening: they represent what *would have* happened if the encryption method was safer. A and B are completely deceived: A knows what message she sent, but she wrongly thinks that B has got it; while B is even wrong about the secret: his arrows point to actions with the wrong preconditions. The bottom node corresponds to sending a 'junk' (or empty) message.

A representation of the epistemic program in DEC. We claim that the epistemic program describing the above *cryptographic attack* can be expressed in (the language of) **DEC**. The total action where the secret ($\boldsymbol{p}$, or $\neg\boldsymbol{p}$) is intercepted, modified and resent to B is $\sigma+\rho$. We claim that: $\sigma = \sigma'' \,|^{C}\, (\sigma''+\rho'')$ and $\rho = \rho'' \,|^{C}\, (\sigma'' + \rho'')$, where

$$\begin{aligned}\sigma'' &= (?\boldsymbol{p} \cdot ({}^{A}(\boldsymbol{p}!_{A,B}) \cdot {}^{B}((\neg\boldsymbol{p})!_{A,B}))\,!^{C}\,(\boldsymbol{p}!_{A,B} + (\neg\boldsymbol{p})!_{A,B} + \textit{skip}),\\ \rho'' &= (?(\neg\boldsymbol{p}) \cdot ({}^{A}((\neg\boldsymbol{p})!_{A,B}) \cdot {}^{B}(\boldsymbol{p}!_{A,B}))\,!^{C}\,(\boldsymbol{p}!_{A,B} + (\neg\boldsymbol{p})!_{A,B} + \textit{skip}).\end{aligned}$$

We also claim that

$$\sigma' = (\boldsymbol{p}!_{A,B})\,!^{C}(\boldsymbol{p}!_{A,B} + (\neg\boldsymbol{p})!_{A,B} + \textit{skip}),$$

$$\rho' = ((\neg\boldsymbol{p})!_{A,B})\,!^{C}(\boldsymbol{p}!_{A,B} + (\neg\boldsymbol{p})!_{A,B} + \textit{skip}),$$

and

$$\gamma = \textit{skip}\,!^{C}\,(\boldsymbol{p}!_{A,B} + (\neg\boldsymbol{p})!_{A,B} + \textit{skip}).$$

According to the action model above, we should have[5]:
$\sigma_C = \rho_C = \{\sigma\} \cup \{\rho\}$, $\sigma_A = \rho_B = \sigma'$, $\rho_A = \sigma_B = \rho'$, $\gamma_A = \gamma_B = \gamma$,
$\sigma'_C = \rho'_C = \gamma_{=}\{\sigma'\} \cup \{\rho'\} \cup \{\gamma\}$, $\sigma'_A = \sigma'_B = \sigma'$, $\rho'_A = \rho'_B = \rho'$.

Remark. The difficulty and the main point of interest in modelling the above cryptographic attack was to find a way to simultaneously express the act of sending the secret by A, intercepting and manipulating it by C, receiving and acknowledging it by B, while-most importantly and at the same time-expressing A's and B's *false beliefs* (that C doesn't know the secret, although he may *suspect* they know it, and also that he can't be sure of that either as the message might have been junk). We also had to find a way to express the fact that A's and B's beliefs are *common knowledge* among all three agents. The latter was solved by making C *publicly announce* his beliefs that either A, and B believe they share the correct secret (i.e. whichever $\boldsymbol{p}$ or $\neg\boldsymbol{p}$ holds) or that nothing is happening, an announcement which was modeled by the *sum* of the (private) group announcement to A and B that $\boldsymbol{p}$, the group announcement to A and B that $\neg\boldsymbol{p}$ and *skip.* But the problem still remained that this has to express what is going on in A's and B's mind about C's beliefs and not what is *truly*

[5] Due to space limitations we leave out the proofs of our claims.

happening. This was tackled in two steps: first, we introduced the two simple actions σ'' and ρ'' which represent the sending, intercepting and receiving of the message *in parallel* with a *public* announcement done by C just like the above, i.e. that he believes $\boldsymbol{p}!_{A,B} + (\neg \boldsymbol{p})!_{A,B} + \mathit{skip}$. In more detail, the act of the message being sent by A, intercepted and manipulated, received and acknowledged by B (without A or B suspecting the interception) was modeled by the sequential composition of the testing the *correct* secret, followed by A's updating with the private announcement between her and B of the *same* secret, followed by B's updating with the private announcement between him and A of the *wrong* secret.

However, we still had to find a way to express that A's and B's beliefs about the original message and C are indeed *false* and that *only* C is aware of that fact, plus the interception and manipulation of the message (without knowing the correct one). In effect, this was achieved my making C *learn all* of the above *while* they are taking place. For that we used the learning operator $\mathcal{L}^{\mathcal{B}}\pi \overset{\text{def}}{=} \pi \,|^{\mathcal{B}}\pi$ introduced earlier. The over all action of the attack was expressed by the sum of simple actions σ and ρ, where in each of them the *real* action taking place is σ'' and ρ'' respectfully, while C *updates* with $\sigma'' + \rho''$. Notice the multiple effect of C's updating: first, C no longer believes the content of his announcement. After all, in any epistemic program of the form $\pi\,!^{C}\pi'$, C may believe that π' is happening, but the real program being executed is in fact π. Second, C's updating with $\sigma'' + \rho''$ is a *totally private* action: both A, and B are *unaware* of it happening, the real program taking place is either σ'' and ρ''. In effect both A and B are deceived. Finally, C's knowledge of the current action taking place $(\sigma'' + \rho'')$ is *correct but incomplete* as he doesn't know which of the simple actions σ'' or ρ'' is currently taking place.

7 Conclusions and Future Work

We introduced a new operation with epistemic programs i.e. $\pi\,!^{\mathcal{B}}_{\mathcal{C}}\,\pi'$ expressing the action of sharing a belief about the current action inside a subgroup. We showed that many useful programs can be defined by $\pi\,!^{\mathcal{B}}_{\mathcal{C}}\,\pi'$. We proposed an equational calculus for this operation and we gave an example of its usage.

Following the investigations described in the paper, there are a number of projects to be taken up. First of all, we would like to prove completeness of the equational calculus. Moreover, we would like to explore the expressive power of the calculus by finding semantic properties that can be expressed in the calculus. For example, a natural question would be: "Which actions and action operations can be defined by terms in the equational calculi"? We would also like to demonstrate the use of the calculus in the analysis of distributed systems. Finally, after extending the range of $\boldsymbol{p}$ (in our proposed syntax) from atomic to epistemic propositions, we want to try and develop a complete calculus for epistemic programs and epistemic propositions.

References

[1] Alexandru Baltag, A logic of epistemic actions, *(Electronic) Proceedings of the FACAS workshop, held at ESSLLI'99*, Utrecht University, Utrecht 1999.

[2] Alexandru Baltag. Logics for Communication: reasoning about information in dialogue games. Lecture notes of a course at *NASSLLI'03.* Available at http://www.indiana.edu/ nasslli

[3] Alexandru Baltag, and L.S. Moss, Logics for epistemic programs, *Synthese* 139:165-224, 2004.

[4] Alexandru Baltag, Lawrence S. Moss, and Sławomir Solecki, The logic of common knowledge, public announcements, and private suspicions, in I. Gilboa (ed), *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'98)*, 1998, 43–56.

[5] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi, **Reasoning About Knowledge**, MIT Press, 1996.

[6] Jelle Gerbrandy, *Bisimulations on Planet Kripke*, Ph.D. dissertation, University of Amsterdam, 1999.

[7] Jelle Gerbrandy and Willem Groeneveld, Reasoning about information change, *Journal of Logic, Language, and Information* **6** (1997) 147–169.

[8] Jan Plaza, Logics of public communications, *Proceedings,* 4^{th} *International Symposium on Methodologies for Intelligent Systems*, 1989.

[9] Penelope Economou, A Parallel Composition Operator for Epistemic Programs, submitted for the *European Summer School on Logic, Language and Information*, 2005.