Verifying Epistemic Protocols under Common Knowledge

Yanjing Wang
Centrum Wiskunde en Informatica
PO Box 94079
1090 GB Amsterdam, NL
y.wang@cwi.nl

Lakshmanan Kuppusamy
Centrum Wiskunde en Informatica
PO Box 94079
1090 GB Amsterdam, NL
klachu@gmail.com

Jan van Eijck
Centrum Wiskunde en Informatica
PO Box 94079
1090 GB Amsterdam, NL
jve@cwi.nl

Abstract

Epistemic protocols are communication protocols aiming at transfer of knowledge in a controlled way. Typically, the preconditions or goals for protocol actions depend on the knowledge of agents, often in nested form. Informal epistemic protocol descriptions for muddy children, coordinated attack, dining cryptographers, Russian cards, secret key exchange are well known. The contribution of this paper is a formal study of a natural requirement on epistemic protocols, that the contents of the protocol can be assumed to be common knowledge. By formalizing this requirement we can prove that there can be no unbiased deterministic protocol for the Russian cards problem. For purposes of our formal analysis we introduce an epistemic protocol language, and we show that its model checking problem is decidable.

1 Introduction

Epistemic protocols have figured in puzzle books for quite some time; an early reference for the muddy children protocol is [9]; they also figure as standard examples in textbooks on epistemic logic [5, 18]. A formal study of epistemic protocols should investigate a number of natural properties of a protocol. More precisely, the protocol should prescribe what happens no matter what the initial situation is, and it should remain correct if the protocol itself is commonly known, including its goal and the precise preconditions for each action in the protocol.

Starting points for our investigation are the perspective on knowledge in *perfect cryptography* from [12], the analysis of Russian cards problems in [15, 2, 18, 16] and the analysis of multiparty secret key exchange in [7, 8, 4].

Compared to the flourishing field of formal verification of communication protocols that started from [3], one thing still lacking from the above accounts is a well-defined language for specifying epistemic protocols. As a consequence of this, formal verification of epistemic protocols is not easy.

Consider the case of Russian cards problems [15]. A Russian cards problem is a specification of a random card distribution among a set of three participants, together with a goal of communicating the hand of one participant to another participant by public announcements, in such a way that the third participant does not learn any card in the actual hands of the other two participants. Solutions to this should take the form of exhaustive lists of concrete distribution of cards with matching announcements, such that the protocol can be executed under an *arbitrary* initial distribution of cards, not just for specific ones. In this paper we will give formal specifications of such protocols. We then can make formal distinctions that have not been made before with informal descriptions of epistemic protocols, like that between deterministic and non-deterministic protocols, and analyze their properties. For example, the 7-hand direct exchange solutions to the *Russian cards problem* provided in [15] suggest unbiased (non-deterministic) protocols that may work on every initial distribution. However, we show that a deterministic protocol that is executable on arbitrary initial distributions exists for this, but that it is necessarily biased.

A notable feature of epistemic protocols, compared to more usual communication protocols, is that the correctness of the epistemic protocols heavily relies on the assumptions of the agents' *meta knowledge* about the protocol itself. It is reasonable to assume that the protocol and its goals are commonly known by all the agents including possible adversaries, if we want to apply the protocol repeatedly in real life cases. The following example of a tentative four hand solution

for Russian cards problem $RCP_{2.2.1}$[1] illustrates how such meta knowledge matters in the verification of the protocol. To check the correctness of protocols under the assumption that the protocol in commonly known, formalization of protocols is clearly imperative.

Example 1 (Guess My Cards) *There are 5 cards (0-4) and three agents $\{A, B, E\}$; agent A has two cards, B has two cards, and E has only one card. A wants to inform B of his hand by public announcement, without revealing his cards to E. A 'promising protocol' for this is that A announces the disjunction of his actual hand (say 01) with all the different combinations of the remaining cards, so he would announce "I have 01 or 23 or 24 or 34." Since B has one more card than E he can eliminate all of 23, 24 and 34, while E can only eliminate two of 23, 24 and 34. However, it does not work like this anymore if E knows that the protocol is meant to reveal A's hand to B. Assume that E has 3. Then E will know that A has either 01 or 24. Now suppose that A has 24 and B has 01. Then B could not have learnt A's hand from A's announcement. So E can infer that A has 01. Another way to see that the would-be protocol is wrong is as follows. The procedure to generate the announcement should also be commonly known. In the above case this procedure is a function from card hands to announcements $f(xy) =$ " I have xy or z_1z_2 or z_2z_3 or z_1z_3.", where z_1, z_2, z_3 are the remaining 3 cards other than x, y. This function is injective, so the announcement reveals the hand immediately.*

1.1 Contributions

The main contributions of this paper are:

- An expressive protocol specification and verification language whose model checking is decidable.
- Formal specifications and checks of epistemic protocols under common knowledge, from which it follows that:
  - the sequential muddy children protocol can be formally proved correct;
  - there is a correct, deterministic biased protocol for $RCP_{3.3.1}$;
  - there is no correct, deterministic and unbiased 2-step protocol for $RCP_{3.3.1}$;
  - the non-deterministic 1-bit secret key generation protocol can be formally proved correct.

[1] The parameters $n.n.k$ express that first agent and second agent each have n cards, and the third has k cards.

Structure of the paper We define the protocol specification language in Section 2. Section 3 talks about epistemic protocols in normal forms and their verification problem under common knowledge. Deterministic protocols for Russian cards problems are studied in Section 4. We also demonstrate non-deterministic protocol verification in Section 5 by looking at a simplified secret-key generation protocol.

2 Preliminaries

Informally, epistemic protocols are the communication patterns which make use of agents' epistemic reasoning power in executions, in order to guarantee the exchange of certain information without leaking undesired information to the possible adversaries. In this paper we focus on the ones which implement public announcements as the only communication methods, since public announcements are the simplest and best studied communication method in logic [18].

2.1 Language and Semantics

We define an *Epistemic Protocol Language* L_{EP} for specifying epistemic protocols and for reasoning about them. The language is kept as simple as possible. More realistic versions may have agent variable assignment, to express things like "for agent := 1 to n do ...". The protocols are meant to be general; there is no intrinsic link between agents and announcements. Such links can be established by restricting announcements to the form "agent a knows that ...".

Assume p ranges over Φ and a over an agent set Ag. The protocol language is a variation on dynamic epistemic languages as defined in [14, 13], with public announcement $[!\phi]$ as the basic communicative operation. The new twist in this paper is that public announcements are among the *epistemic programs* π as defined below. Further on, when we discuss the specification and analysis of unbiased protocols, we will extend the language with a graded modality.

$$\begin{array}{rcl} \phi & ::= & \top \mid p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid [\pi]\phi \\ \pi & ::= & a \mid ?\phi \mid !\phi \mid \pi_1;\pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^* \end{array}$$

Below, we will be more specific about basic propositions p, and may take them to be of the certain forms, e.g. $has_a x$ for $a \in Ag$, for certain applications. We employ the usual abbreviations: $\phi \vee \psi$, $\phi \rightarrow \psi$ and $\langle\pi\rangle\phi$ are shorthand for $\neg(\neg\phi \wedge \neg\psi)$, $\neg\phi \vee \psi$ and $\neg[\pi]\neg\phi$, respectively. The truth value of a L_{EP} formula ϕ in a state s of a multi-S5 Kripke model $\mathcal{M} = (S, \{\sim_i \mid i \in Ag\}, V)$, is defined by:

$\mathcal{M}, s \vDash p$	$\Leftrightarrow$	$p \in V(s)$
$\mathcal{M}, s \vDash \neg\phi$	$\Leftrightarrow$	$\mathcal{M}, s \nvDash \phi$
$\mathcal{M}, s \vDash \phi \wedge \psi$	$\Leftrightarrow$	$\mathcal{M}, s \vDash \phi$ and $\mathcal{M}, s \vDash \psi$
$\mathcal{M}, s \vDash [\pi]\phi$	$\Leftrightarrow$	for all $\mathcal{M}', s' : (\mathcal{M}, s)[\![\pi]\!](\mathcal{M}', s')$ implies $\mathcal{M}', s' \vDash \phi$

where π are epistemic programs functioning as *model changers*:

$(\mathcal{M}, s)[\![a]\!](\mathcal{M}', s')$	$\Leftrightarrow$	$\mathcal{M}' = \mathcal{M}$ and $s \sim_a s'$
$(\mathcal{M}, s)[\![?\psi]\!](\mathcal{M}', s')$	$\Leftrightarrow$	$(\mathcal{M}', s') = (\mathcal{M}, s)$ and $\mathcal{M}, s \vDash \psi$
$(\mathcal{M}, s)[\![!\psi]\!](\mathcal{M}', s')$	$\Leftrightarrow$	$(\mathcal{M}', s') = (\mathcal{M}\|_\psi, s)$ and $\mathcal{M}, s \vDash \psi$
$(\mathcal{M}, s)[\![\pi_1; \pi_2]\!](\mathcal{M}', s')$	$\Leftrightarrow$	$(\mathcal{M}, s)[\![\pi_1]\!] \circ [\![\pi_2]\!](\mathcal{M}', s')$
$(\mathcal{M}, s)[\![\pi_1 \cup \pi_2]\!](\mathcal{M}', s')$	$\Leftrightarrow$	$(\mathcal{M}, s)[\![\pi_1]\!] \cup [\![\pi_2]\!](\mathcal{M}', s')$
$(\mathcal{M}, s)[\![(\pi_1)^*]\!](\mathcal{M}', s')$	$\Leftrightarrow$	$(\mathcal{M}, s)[\![\pi_1]\!]^*(\mathcal{M}', s')$

where $\mathcal{M}|_\psi$ is the restriction of $\mathcal{M}$ to the states where ψ holds; $\circ, \cup$ and * at right-hand side express the usual composition, union and reflexive transitive closure on relations respectively.

As usual, in order to emphasise the intuitive epistemic meanings of some of our operators, we write $K_a\phi$ for $[a]\phi$ and we use $C\phi$ for $[(\bigcup_{i \in Ag} i)^*]\phi$ (the common knowledge operator).

A notable difference between our language and the PDL-style dynamic epistemic languages as in [14, 13] is that we treat atomic programs and announcements in an uniform way. Thus, we not only allow complicated program constructions on announcements like $(!\phi \cup !\psi)^*$ but also the interaction between atomic programs and announcements. For example, $[(!\psi; (a \cup b))^*]\phi$ expresses conditional common knowledge of ϕ among a, b w.r.t. announcements. When we interpret basic programs as arbitrary basic actions as in PDL, then $(?\psi; (!\psi \cup a))^*$ can express a protocol which makes choices repeatedly between an announcement and a basic action while ψ holds.

To understand the expressivity better, we identify a fragment of L_{EP} which can be translated into PDL. Call a formula *echo-free* if it has no public announcements in the scope of a star. Any echo-free L_{EP} formula can be translated into a formula without announcements, by proceeding in two steps. The first translation t is as follows:

$$
\begin{array}{rcl}
t(\top) & = & \top \\
t(p) & = & p \\
t(\neg\phi) & = & \neg t(\phi) \\
t(\phi_1 \wedge \phi_2) & = & t(\phi_1) \wedge t(\phi_2) \\
t([a]\phi) & = & [a]t(\phi) \\
t([?\psi]\phi) & = & t(\psi) \to t(\phi) \\
t([!\psi]\phi) & = & [!t(\psi)]t(\phi) \\
t([\pi_1 \cup \pi_2]\phi) & = & t([\pi_1]\phi) \wedge t([\pi_2]\phi) \\
t([\pi_1; \pi_2]\phi) & = & t([\pi_1][\pi_2]\phi) \\
t([\pi^*]\phi) & = & [\pi^*]t(\phi)
\end{array}
$$

This yields an equivalent formula where each program π either has the form $!\phi$ or is announcement-free. Thus the transformed formulas of t can be regarded as LCC formulas in [13], if we consider the public announcements as the corresponding single-pointed action models. Next, apply the translation procedure T in [13] to transform the LCC formulas into PDL.

The translation $T \circ t$ thus yields an equivalent PDL formula, for all echo-free L_{EP} formulas. Note that this translation cannot be extended to the full L_{EP} language, due to the result of [11] which states that the satisfiability problem of a language containing at least iterated relativization $((!\phi)^*)$ and common knowledge operators is undecidable, even on finite models. However, for model checking problem we have:

Proposition 1 *Model checking L_{EP} on finite Kripke models is decidable.*

Proof: The idea of the proof is based on the observation that the epistemic programs are eliminative in nature. We want to turn L_{EP} model checking $\mathcal{M}, s \vDash \phi$ into PDL-style model checking on a larger finite model $\mathcal{N}$ such that instead of interpreting π in ϕ as model changers on $\mathcal{M}$, we can see π as a label for a relation in $\mathcal{N}$. Intuitively, we build $\mathcal{N}$ by making all the possible pointed sub-models $(\mathcal{M}', s')$ of $\mathcal{M}$ as the states in $\mathcal{N}$. In $\mathcal{N}$, the a-relations $\sim_a^{\mathcal{N}}$ are defined as follows: $(\mathcal{M}', s') \sim_a^{\mathcal{N}} (\mathcal{M}'', s'') \iff \mathcal{M}' = \mathcal{M}''$ and $s' \sim_a s''$ in $\mathcal{M}'$. Valuations $V^{\mathcal{N}}((\mathcal{M}', s')) = V^{\mathcal{M}'}(s')$. Now we are ready to compute all the corresponding relations of π in $\mathcal{N}$ by usual treatments in PDL model checking algorithms for PDL-operators $;, \cup$ and * and the following operation to deal with $!\phi'$: $\mathcal{M}', s' \to_{!\phi'} \mathcal{M}'', s''$ iff $\mathcal{M}'' = \mathcal{M}'|_{\phi'}$ and $s' = s''$. To compute $\mathcal{M}'|_{\phi'}$ we need to call the model checking algorithm again but since ϕ' is strictly simpler than ϕ, we will finally arrive at a situation that can be handled by the PDL model checking algorithm. $\square$

3 Epistemic protocols

In this section, we address the epistemic protocols and their verification problem formally. We first restrict ourselves to a sub-language L_{SP} of L_{EP} with proposition set P and agent set Ag, in order to specify the epistemic protocols in a simpler but adequate form:

$$
\begin{array}{rcl}
\phi & ::= & \top \mid p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid [\pi]\phi \\
\pi & ::= & \pi_{act} \mid \pi; \pi' \mid \pi^* \\
\pi_{act} & ::= & ?\phi; !\psi \mid \pi_{act} \cup \pi'_{act}
\end{array}
$$

Intuitively we want to specify a sequence of conditional announcements which may involve non-deterministic choices. Actions $?\phi; !\psi$ are guarded announcements, with ϕ as the precondition and ψ as the message.

Note that a protocol often comes with assumptions about the initial situation where the protocol can be applied and implicitly a set of *"axioms"* about the facts which should remain true while the protocol is running. These define the *"physical setting"* of the protocol. For example, in RCP we may assume that at the beginning each player has a certain number of different cards and for each card c it holds that $has_i c \to \bigwedge_{j \neq i} \neg has_j c$, where $has_j c$ is a basic proposition with the obvious meaning: j has card c. In this paper we will not focus on the specification of the initial states, but assume that given a protocol $\texttt{Prot}$, there is an initial model $\mathcal{M}_{\texttt{Prot}}$, with a set of *axioms* $\mathcal{T}_{\texttt{Prot}}$ which are valid on $\mathcal{M}_{\texttt{Prot}}$ and remain valid while executing the protocol[2]. $\mathcal{T}_{\texttt{Prot}}$ is used in the following to define deterministic protocols which intuitively can execute only one action at any time starting from the initial situation.

Definition 1 (Epistemic Protocol) *An epistemic protocol* $\texttt{Prot}$ *is a pair* $\langle \pi_{\texttt{Prot}}, \Phi_{\texttt{Prot}} \rangle$ *where* $\pi_{\texttt{Prot}}$ *is an epistemic program in the language* L_{SP} *and* $\Phi_{\texttt{Prot}}$ *is a set of (announcement-free) formulas serving as the goal of the protocol. A step of* $\texttt{Prot}$ *is a guarded action or a choice among guarded actions. A step* $?\phi_1; !\psi_1 \cup \cdots \cup ?\phi_n; !\psi_n$ *is called deterministic if it holds that* $\forall i \neq j < n$:

$$\text{If } \{\phi_i \wedge \phi_j\} \cup \mathcal{T}_{\texttt{Prot}} \text{ is satisfiable then } \psi_i = \psi_j.$$

A protocol is called deterministic if each of its steps is, and non-deterministic if it is not deterministic. A protocol $\texttt{Prot}$ *is of definite length if* $\pi_{\texttt{Prot}}$ *is echo-free, otherwise it is of indefinite length. The length of a protocol of definite length is the number of its steps.*

Note that the goals in the above definition are intended to be met at the end of the protocol, but our formalism can also express checks of protocol steps, by sequentially composing the step π with a guarded command $?\phi; !\top$ (effectively a check of ϕ).

An epistemic protocol is expected to be implemented in an environment where every announcement is publicly broadcasted in the possible presence of some passive eavesdropper. A *run* of the protocol is a sequence of guarded announcements $?\phi; !\psi$, which is executable on the initial model according to the protocol specification. We will assume that no different instantiations of a protocol run in parallel.

Definition 2 (Verification of Epistemic Protocol) *Verification of an epistemic protocol* $\texttt{Prot}$ *is checking whether all the goals hold after any run or some run of the protocol against the initial model* $\mathcal{M}_{\texttt{Prot}}$ *under any initial state. The usual distinction between safety and liveness properties applies. Checks of safety properties have the form* $\mathcal{M}_{\texttt{Prot}} \vDash [\pi]\phi$*, those of liveness properties are of the form* $\mathcal{M}_{\texttt{Prot}} \vDash \langle \pi \rangle \phi$*, where* $\mathcal{M}_{\texttt{Prot}} \vDash \varphi$ *iff for all* $s \in \mathcal{M}_{\texttt{Prot}}, s \vDash \varphi$.

Note that if a deterministic protocol is of definite length then checking $[\pi]\phi$ coincides with $\langle \pi \rangle \phi$, if the protocol is always executable on $\mathcal{M}_{\texttt{Prot}}$. For protocols of indefinite length, we may want to check $\langle \pi \rangle \phi$ to make sure the protocol achieves its goal ϕ at some finite run.

We call a model *connected* if every state is connected to all other states by a path of epistemic relations. Note that in most applications, the initial models are connected, assuming that the agents are perfect reasoners who can imagine the possibilities others may think given the facts they can observe. Now we can use common knowledge to reformulate the verification problem:

Proposition 2 *Suppose* $\mathcal{M}$ *is a connected model then for any* $s \in \mathcal{M}$,

$$\mathcal{M} \vDash \phi \iff \mathcal{M}, s \vDash C\phi$$

This means to verify that a protocol is correct under any possible initial information distribution is to check the common knowledge of the correctness of the protocol at *some arbitrary* initial situation.

As we motivated in the introduction, we assume the protocol is commonly known in the following sense:

1. The guards of the actions are commonly known;
2. The truthfulness of the announcements is commonly known;
3. The goals of the protocol are commonly known.

Such requirements call for a bit of further streamlining on the form of protocols in our framework. To motivate this, consider the following choice:

$$(?q; !\neg p) \cup (?\neg q; !p).$$

According to requirement (1) above, the agent should be able to learn q from the announcement of $\neg p$. In general, for each guarded action $?\phi; !\psi$, we can make the precondition commonly known by announcing it too. Thus we may assume: $\psi \to \phi$ is valid. According to the above requirement (2), for each $?\phi; !\psi$ that occurs in a protocol we may assume that $\phi \to \psi$ is valid. Some reflection shows that the above requirements together boil down to the requirement that in

[2]Formally $\mathcal{T}_{\texttt{Prot}}$ can be considered as a subset of the formulas that are not only valid at $\mathcal{M}_{\texttt{Prot}}$ but also preserved under any π operations.

each guarded action the guard and the announcement are the same. Thus, we can restrict our attention to choices of the following *normal form*:

$\bigcup_i ?!\phi_i$, where $?!\phi_i$ is an abbreviation for $?\phi_i; !\phi_i$ [3].

Given an epistemic protocol `Prot`, we can transform every step $?\phi_1; !\psi_1 \cup \cdots \cup ?\phi_n; !\psi_n$ of it into the right shape under the assumptions of (1) and (2) by the following procedure:

- Lump the same actions in each step of the protocol together: if $\psi_1 = \cdots = \psi_m$ then we can replace $?\phi_1; !\psi_1 \cup \cdots \cup ?\phi_m; !\psi_m$ by an equivalent single guarded action $?(\phi_1 \vee \cdots \vee \phi_m); !\psi_1$.
- Transform every appearance of $?\phi_1; !\psi_1$ into $?(\phi_1 \wedge \psi_1); !(\phi_1 \wedge \psi_1)$.

To implement requirement (3), the straightforward idea would be simply checking the common knowledge of the correctness of the protocol ($C[\pi]\phi$ or $C\langle\pi\rangle\phi$). We will show this is indeed enough to guarantee the protocol is correct under the assumption that the agents know the goals that the protocol should fulfil.

Let us start from the observation made in [15] that just checking $\mathcal{M}, w \vDash [!\psi]\phi_{goal}$ is sometimes not sufficient, for a one step protocol $!\psi$ aiming at establishing ϕ_{goal}. Indeed, if the agents know the intended goal of the protocol then they will assume that others do not perform actions which do not lead to the goal. Such an assumption gives agents the power to reason more, as we also saw in Section 1, which sometimes destroys the correctness of the protocol. Now we can try to *make agents know* the goal of protocol by announcing it.

In [15] the author proposed that the verification should be undertaken while an announcement $!\psi$ is interpreted more than just announcing ψ[4]:

$$\mathcal{M}, w \vDash [!(\psi \wedge [!\psi]\phi_{goal})]\phi_{goal}$$

The idea behind this is that we announce the goal of the announcement to make sure that under the assumption that agents know the goal, the protocol is still correct. However, if $[!(\psi \wedge [!\psi]\phi_{goal})]\phi_{goal}$ is now assumed and known by agents, we still need to make sure that knowing *this* again does not affect the correctness of the protocol. We can iterate such reasoning *ad libitum*. Formally we define:

- $\eta_0 = [!\psi]\phi$
- $\eta_{i+1} = [!(\psi \wedge \eta_0 \wedge \cdots \wedge \eta_i)]\phi$

We can simplify η_{i+1}, by making use of the valid formula $[!(\psi \wedge [!\psi]\phi)]\chi \leftrightarrow [!\psi][!\phi]\chi$:

Proposition 3 $\eta_{i+1} = [!\psi; \underbrace{!\phi; \ldots; !\phi}_{i}]\phi$

We actually need to check all η_i, since there are cases where all the η_i are logically different[5].

Notice that if ϕ_{goal} is in the shape of $C\phi$ then

$$\eta_{i+1} = [!\psi]\underbrace{[!C\phi]\ldots[!C\phi]}_{i}C\phi \Leftrightarrow [!\psi]C\phi = \eta_0.$$

due to the fact that $[!C\phi]C\phi$ is a valid formula [18], thus making the infinite process of checking η_i manageable. In [15], the author suggests that instead of checking property ϕ, we should check property $C\phi$. Note that the simplification in Proposition 3 works on the one-announcement-protocols, but it is not very clear how to deal with the more complicated forms of the epistemic protocols as we defined in this paper. Thus checking common knowledge *after* the run of the protocol may not be grounded.

Now we take another perspective, instead of reinterpreting each announcements, we address the formula to be checked as a whole. Intuitively, we strengthen $[\pi]\phi_{goal}$ by some ψ such that:

- ψ should imply $[\pi]\phi_{goal}$.
- if $[\pi]\psi_{goal}$ is true then truthfully announcing ϕ in advance should not change the truth value of $[\pi]\phi_{goal}$.

Thus formally we require:

$$\psi \leftrightarrow [\pi]\phi_{goal} \wedge \langle!\psi\rangle[\pi]\phi_{goal}$$

Unfortunately, $f(X) = [\pi]\phi_{goal} \wedge \langle!X\rangle[\pi]\phi_{goal}$ is not a monotonic function, given no restriction on $[\pi]\phi_{goal}$. However, it is not hard to see that the common knowledge of the *correctness of the protocol itself* is indeed a fixed point for $f(X)$:

Proposition 4 *$C[\pi]\phi_{goal}$ is a fixed point of $f(X)$, but $[\pi]C\phi_{goal}$ is not always a fixed point of $f(X)$.*

[3] Although equivalent to $!\phi$, $?!\phi$ is still used in the following for its clearer reading in protocol specification according to L_{SP}.

[4] In the original setting of [15], it is suggested that the announcement of $!\psi$ by agent a aiming at establishing ψ is actually $!(\psi \wedge [!\psi]K_a\phi)$, we omit the details in [15] that are relevant to the context of Russian cards problem.

[5] Consider the dynamic epistemic analysis of the traditional Muddy Children puzzle [18]. There is always a model $\mathcal{M}, s$ such that $\mathcal{M}, s \vDash [!\psi; \underbrace{!\phi; \ldots; !\phi}_{i}]\phi$, but $\mathcal{M}, s \nvDash [!\psi; \underbrace{!\phi; \ldots; !\phi}_{i+1}]\phi$ where ϕ is the formula that expresses "We do not know whether we are dirty or not".

To see that $[\pi]C\phi_{goal}$ is not always a fixed point of $f(X)$, let $\pi = !(q \vee K_a p)$ and $\phi_{goal} = r$. $[\pi]C\phi_{goal} \not\rightarrow \langle![\pi]C\phi_{goal}\rangle[\pi]C\phi_{goal}$ can be witnessed on the following model[6] $M, \blacktriangle$:

$$p, \neg q, \neg r : \blacktriangle \xleftrightarrow{a} \bullet : \neg p, q, \neg r$$

Moreover, $C[\pi]\phi$ is indeed a stronger requirement than $[\pi]C\phi$ as the following shows:

Proposition 5 *For any epistemic program π of L_{SP} that is in the normal form, $C[\pi]\phi \rightarrow [\pi]C\phi$ is valid. However, the converse does not hold in general.*

The above fixed point analysis also apply to $C\langle\pi\rangle\phi_{goal}$ in case we check liveness properties. Thus we can now define verification epistemic protocols under common knowledge:

Definition 3 ***(Verification under Common Knowledge)*** *Verification of an epistemic protocol* $\mathtt{Prot}$ *under common knowledge is checking $\mathcal{M}_{\mathtt{Prot}} \vDash C[\pi_{\mathtt{Prot}}]\phi_{goal}$ for safety properties or checking $\mathcal{M}_{\mathtt{Prot}} \vDash C\langle\pi_{\mathtt{Prot}}\rangle\phi_{goal}$ for liveness properties.*

Now let us look at a variation of classic Muddy Children to demonstrate how we specify and verify an epistemic protocol.

Example 2 (Sequential n-Muddy children [19]) *The setting is as follows: some of n children $(1, 2, \ldots, n)$ got mud on their foreheads while playing. The children can see whether other kids are dirty, but there is no mirror for them to discover themselves whether they are dirty or not. Now the father walks in and states: "At least one of you is dirty!" Then he asks the children $1, 2, \ldots, n$ one by one (i.e., sequentially), "Do you know whether you are dirty?" until he has asked everyone. The children have to answers "Yes" or "No" truthfully. Suppose child j is the last dirty child in the sequence. Then j will know that he is dirty when it is his time to answer. And all the children after him will know that they are clean. But the $j-1$ children before j will remain ignorant all the time about whether they are dirty or clean (under the usual assumption that the children are honest and perfect reasoners). Here are the formal details of the protocol:*

- *Let d_i be the basic proposition expressing "child i is dirty" and c_i be $\neg d_i$.*
- *Let $\mathtt{ToKorNot}_i = (?!\mathtt{Know}_i) \cup (?!\neg\mathtt{Know}_i)$ express that i truthfully announces whether he knows he is dirty or not, where $\mathtt{Know}_i = K_i d_i \vee K_i c_i$.*
- *Let $\mathtt{LastDirty}_i$ be the formula $d_i \wedge \bigwedge_{j>i} c_j$ expressing that d_i is the last dirty child according to the ordering $>$.*

Then $\mathtt{Prot}_{SMD} = \langle \pi_{SMD_n}, \{\phi_{SMD_n}\}\rangle$ where

$$\pi_{SMD_n} = (!\bigvee_i d_i); \mathtt{ToKorNot}_1; \mathtt{ToKorNot}_2; \ldots; \mathtt{ToKorNot}_n$$

$$\phi_{SMD_n} = \bigwedge_i (\mathtt{LastDirty}_i \rightarrow (\bigwedge_{j<i} \neg\mathtt{Know}_j \wedge \bigwedge_{j\geq i} \mathtt{Know}_j)$$

A straightforward initial model $\mathcal{M}_{\mathtt{Prot}} = \{W, \{\sim_i | \ i \in I\}, V\}$ is a connected model where:

- $W = \{\langle cd_1, cd_2, \cdots, cd_n\rangle \mid cd_i \in \{c_i, d_i\}\}$
- $w \sim_i v \iff \langle cd_1, cd_2, \cdots, cd_n\rangle = w$, $\langle cd'_1, cd'_2, \cdots, cd'_n\rangle = v$ and $cd_j = cd'_j$ for all $j \neq i$.
- $V(d_i)(w) = 1 \iff w = \langle cd_1, \cdots, cd_n\rangle$ and $cd_i = d_i$.

Clearly $\mathtt{Prot}_{SMD}$ is deterministic since the child can only know or not know whether himslef is dirty, no matter what $\mathcal{T}_{\mathtt{Prot}}$ is. Nevertheless, the following intuitive axiom says all the children can see whether others are dirty or not: $\mathcal{T}_{\mathtt{Prot}} = \{C \bigwedge_{j\neq i}(K_i d_j \vee K_i c_j)\}$. We can then verify the π_{SMD_n}:

Proposition 6 $\mathcal{M}_{\mathtt{Prot}} \vDash C[\pi_{SMD_n}]\Phi_{SMD_n}$.

4 Deterministic Protocols for Russian Cards Problem

4.1 Formalizing Russian Cards Problem

In this section, we study deterministic 2-step protocols for Russian cards problem that can be executed under *arbitrary* initial distribution of cards, not for a particular distribution as in many previous discussions[15, 18]. We show that there can only be deterministic protocols for $RCP_{3.3.1}$ with uneven appearances of cards in the announcements.

We first model the general case of $RCP_{n.n.k}$: Let $I = \{A, B, E\}$ be the set of players, $Dk = \{0, 1, ..., 2n + k - 1\}$ be the set of $2n + k$ cards, Hs^h be the set of h-hands (e.g. $Hs^3 = \{\{x, y, z\} \mid x, y, z \in Dk$ and x, y, z are different$\}$). Let $has_i x$ be the basic proposition meaning that player i has card x; $has_i X$ be the shorthand of $\bigwedge_{x\in X} has_i x$. $\mathcal{T}_{\mathtt{Prot}} = \{\mathtt{OneCardInOneP}, \mathtt{EkCards}, \mathtt{ABnCards}, \mathtt{KnowThyself}\}$ where:

- $\mathtt{EkCards}$: $\bigvee_{X\in Hs^k} has_E X$;
  $\mathtt{ABnCards}$: $\bigwedge_{i\in\{A,B\}} \bigvee_{X\in Hs^n} has_i X$

[6] Reflexive arrows are omitted.

- `OneCardInOneP:` $\bigwedge_{i\neq j}(\bigwedge_{x\in Dk}(has_i x \to \neg has_j x)$
- `KnowThyself:` $\bigwedge_{i\in I}\bigwedge_{X\in Hs^k}(has_i X \to K_i has_i X)$

$\mathcal{M}_{\texttt{Prot}} = \{W, \{\sim_i |\ i \in I\}, V\}$ is a connected model[7] where:

- $W = \{\langle X, Y, Z\rangle \mid X, Y \in Hs^n, Z \in Hs^k, X \cup Y \cup Z = Dk\}$
- $w \sim_i v \iff w_i = v_i$ where $\langle X, Y, Z\rangle_A = X$; $\langle X, Y, Z\rangle_B = Y$; $\langle X, Y, Z\rangle_E = Z$.
- $V(has_i x)(w) = 1 \iff x \in w_i$.

The goals of the protocol are:

$$\phi_1 = \bigwedge_{x\in Dk}(has_A x \to K_B has_A x)$$

$$\phi_2 = \bigwedge_{x\in Dk}(has_B x \to K_A has_B x)$$

$$\phi_3 = \bigwedge_{x\in Dk}((has_A x \to \neg K_E has_A x)\wedge(has_B x \to \neg K_E has_B x))$$

If the protocol `Prot` is deterministic, and executable on arbitrary initial distribution then according to the previous section, we check $\mathcal{M}_{\texttt{Prot}} \vDash C\langle\pi_{\texttt{Prot}}\rangle(\phi_1 \wedge \phi_2 \wedge \phi_3)$. The following proposition shows that we can then safely focus on the first step of the protocol which should satisfy ϕ_1 and ϕ_3.

Proposition 7 *If there is an one-step protocol π such that $\mathcal{M}_{Prot} \vDash C\langle\pi\rangle(\phi_1 \wedge \phi_3)$, then there is a π' such that $\mathcal{M}_{Prot} \vDash C\langle\pi;\pi'\rangle(\phi_1 \wedge \phi_2 \wedge \phi_3)$.*

Proof:Let $\pi' = \bigcup_{X\in Hs^k} ?!K_B has_E X$ we show that if $\mathcal{M}_{\texttt{Prot}} \vDash C\langle\pi\rangle(\phi_1 \wedge \phi_3)$ then $\mathcal{M}_{\texttt{Prot}} \vDash C\langle\pi;\pi'\rangle(\phi_1 \wedge \phi_2 \wedge \phi_3)$. Let $K_i Card_j$ be the abbreviation for $\bigwedge_{x\in Dk}(has_j x \to K_i has_j x)$. It is obvious that $\mathcal{M}_{\texttt{Prot}} \vDash C\langle\pi;\pi'\rangle K_A Card_E$. From $\mathcal{T}_{\texttt{Prot}} \vDash K_A Card_E \to K_A Card_B$, we have $\mathcal{M}_{\texttt{Prot}} \vDash C\langle\pi;\pi'\rangle\phi_2$. It is not hard to see that ϕ_1 is monotonic to model relativizations, namely if it is true at $\mathcal{M}, s$ then it is true in any possible restrictions of $\mathcal{M}, s$. Thus $\mathcal{M}_{\texttt{Prot}} \vDash C\langle\pi;\pi'\rangle\phi_1$. For ϕ_3, first we know from Proposition 5, $\mathcal{M}_{\texttt{Prot}} \vDash \langle\pi\rangle C\phi_1$. Thus from $\mathcal{T}_{\texttt{Prot}} \vDash K_B Card_A \to K_B Card_E$, we have $\mathcal{M}_{\texttt{Prot}} \vDash \langle\pi\rangle C K_B Card_E$. Therefore for each world s where E's actual hand is X, $\mathcal{M}_{\texttt{Prot}}, s \vDash \langle\pi\rangle K_E K_B has_E X$. Thus truthfully announcing $K_B has_E X$ will not change the worlds that E considers possible. Thus for factual formula ψ (without knowledge operators), $\mathcal{M}_{\texttt{Prot}} \vDash \langle\pi\rangle\neg K_E\psi \leftrightarrow \langle\pi;\pi'\rangle\neg K_E\psi$. Then it is not hard to see that $\mathcal{M}_{\texttt{Prot}} \vDash C\langle\pi;\pi'\rangle(\phi_1 \wedge \phi_2 \wedge \phi_3)$. □

[7]It is easy to see that $\mathcal{M}_{\texttt{Prot}} \vDash \mathcal{T}_{\texttt{Prot}}$, but not everything valid in $\mathcal{M}_{\texttt{Prot}}$ are specified in $\mathcal{T}_{\texttt{Prot}}$, e.g. let $\phi = \bigwedge_{i\neq j\in I}\bigwedge_{x\in Dk}(has_j x \to \neg K_i has_j x)$, then $\mathcal{M}_{\texttt{Prot}} \vDash \phi$ but $\mathcal{T}_{\texttt{Prot}} \nvDash \phi$. $\mathcal{T}_{\texttt{Prot}}$ includes the hard facts which remain unchanged while the protocol is applied. However, some agents may know something they did not know before.

Now we restrict the form of our protocol further by the adaption of a result from [15], which states that to announce only A's alternative hands is enough.

Proposition 8 *If a correct 2-step protocol of the Russian cards problem $RCP_{n.n.k}$ exists, there is another correct protocol with the first step in the form of:*

$$\pi ::= ?!Pa_0 \cup ?!Pa_1 \cup \cdots \cup ?!Pa_m$$

where Pa_i is in the form of $\bigvee_{j\leq m} has_A X_j$ (i.e. A's alternative hands).

We now prove a lemma for our negative result later in this section.

Lemma 1 *The first step of a correct 2-step deterministic protocol of the Russian cards problem $RCP_{n.n.k}$ should at least satisfy:*

1. *each possible hand appears once and only once in $?!Pa_0 \cup ?!Pa_1 \cup \cdots \cup ?!Pa_m$.*
2. *any two hands in one announcement Pa_j can only share at most $n-k-1$ common cards.*

Proof: For (1): From Proposition 8 and the requirement that $\mathcal{M}_{\texttt{Prot}} \vDash C\langle\pi_{\texttt{Prot}}\rangle\top$ we know every hand should appear at least once. From the fact that protocol should be deterministic, every hand can only appear once. In the following, given a hand X of A, let $Pa(X)$ be the announcement Pa_j in the protocol such that $has_A X$ is a disjunct of Pa_j.
For (2): To let B know A's cards after A's announcement, we should make sure that given A's hand X, for any B's hand Z, the alternatives in $Pa(X)$ will be ruled out. Namely, for any different hands $X, Y \in Pa(X)$, any hand $Z \subseteq Dk\backslash X$ that B may have: $Z \cap Y \neq \emptyset$. This means that for every two hands X, Y in Pa_i, the number of cards different from the cards in $X \cup Y$ must be less than n. Otherwise there is a possible hand Z which does not intersect with both X and Y. Thus, we have $|Dk| - |Y \cup X| < n$. Since $|Dk| = 2n + k$, $|Y \cup X| > n + k$. Therefore it is not hard to see that $|X \cap Y| < n - k$. □

In the following we will concentrate on the original Russian cards problem $RCP_{3.3.1}$ as coined in [15]. We first show that there is a deterministic protocol:

Theorem 1 *There is a correct, 2-step deterministic protocol for $RCP_{3.3.1}$*[8].

[8]The solution was found with the help of the Alloy Analyzer cf.[10]

Proof: Let $has_A Pa_i$ be the abbreviation for $\bigvee_{X \in Pa_i} has_A X$ where

Pa_0 :	012	036	045	134	156	235	246
Pa_1 :	013	025	046	126	145	234	356
Pa_2 :	014	026	035	136	245		
Pa_3 :	015	024	123	256	346		
Pa_4 :	016	034	124	135	236	456	
Pa_5 :	023	056	125	146	345		

Let $\pi = \bigcup_{0 \le i \le 5}(?!has_A Pa_i)$. Note that π satisfies the conditions in Lemma 1. Moreover we can verify that $M \vDash C\langle\pi\rangle(\phi_1 \wedge \phi_3)$. Thus from Proposition 7, there is a deterministic protocol for $RCP_{3.3.1}$. □

However, the above protocol is *biased* in the sense that not all the cards appear evenly in all the possible announcement(e.g. in $Pa_2, 0$ appears 3 times but others only appear twice). Thus an eavesdropper may learn that some card is more likely to be held by A. Thus it is preferable to have an unbiased protocol with evenly appearances of cards. Here are some properties of the unbiased protocol for $RCP_{3.3.1}$, if exists:

Lemma 2 *The first step of an unbiased deterministic protocol for $RCP_{3.3.1}$ must satisfy the following:*

1. *each announcement Pa_j contains, and only contains, 7 alternative hands.*
2. *there are in total 5 alternative announcements in the protocol.*
3. *every two hands in the same announcement have exactly one card in common.*

Proof: For (1): If all the cards appear evenly (suppose g times) in any announcement with k hands, then $3k = 7g$. So the minimal k is 7, and each card appears 3 times. We claim that if k is greater than 7 then there must be two hands which share more than 1 cards. Note that there are only $C_7^2 = 21$ different pairs of cards and each hand contains 3 different pairs. From the second statement in Lemma 1 any two hands should not have a pair of cards in common, 7 hands then covers all the possible different pairs. Thus adding one more hand must result in two hands share two cards in common.
For (2): From the first statement in Lemma 1, we know the $C_7^3 = 35$ hands should all appear in the protocol once. Thus from (1) the protocol should have 5 alternative 7-hand announcements.
For (3): Suppose there are two hands X, Y in an announcement such that $X \cap Y = \emptyset$. Without loss of generality let $X = 123, y = 456$. Since each of the possible 21 pairs should appear in some hand the announcement as argued in (1), then the hands $14c$ and $24c'$ must also appear in the same announcement for some cards c, c'. Since every two hands should not have two cards in common then $c, c' \notin X \cup Y$ thus $c, c' \in Dk \backslash X \cup Y$, namely $c = c' = 0$. However now $14c$ and $24c'$ have two cards in common, contradiction. □

Moreover, we need to require that E cannot infer some card is more likely than others to be held by A. To formally specify this requirement we need some form of graded modality as in [6]. Since here we only need to express whether $has_A x$ and $has_A y$ are equally possible to E, we introduce a 2-ary modalities B_a into the language L_{SP} with the following semantics:

$$\mathcal{M}, s \vDash B_a(\psi, \phi) \iff \sharp_a(s, \phi) = \sharp_a(s, \psi)$$

where $\sharp_a(s, \chi)|\{t \mid s \sim_a t \text{ and } \mathcal{M}, t \vDash \chi\}|$. Clearly, adding this modality does not destroy the decidability of the model checking problem on finite models.

Now we can show that an unbiased protocol, if exists, also guarantees that player E does not have a lucky guess:

Proposition 9 *If there exists an unbiased protocol π_{Prot} for $RCP_{3,3,1}$ then $\mathcal{M}_{Prot} \vDash C\langle\pi_{Prot}\rangle \bigwedge_{x,y \in Dk}((\neg K_E \neg has_A x \wedge \neg K_E \neg has_A y) \to B_E(has_A x, has_A y))$.*

Proof:Given an announcement Pa(as a set of alternative hands) in an unbiased protocol for $RCP_{3.3.1}$, for any card $c \in Dk$, let $S_E = \{X | c \in X \text{ and } X \in Pa\}$. From Lemma 1, we know that the alternative hands in S_E^c should not have 2 cards in common. So any card that appears in S_E^c only appears once. From the proof of statement 1 in Lemma 2, we know every card in $Dk \backslash \{c\}$ must appear in S_E^c. Thus every card in $Dk \backslash \{c\}$ only appears once in the hands in S_E^c. Since Pa is unbiased then $Pa \backslash S_E^c$ is still unbiased. Thus it is not hard to see that $\mathcal{M}_{\mathtt{Prot}} \vDash C\langle\pi_{\mathtt{Prot}}\rangle \bigwedge_{x,y \in Dk}((\neg K_E \neg has_A x \wedge \neg K_E \neg has_A y) \to B_E(has_A x, has_A y))$. □

The authors of [2] showed that *unbiased* non-deterministic protocols exist, by making use of probabilistic selections[9]. However, in the following, we show that there is no deterministic protocol which is unbiased.

Theorem 2 *There is no correct deterministic 2-step protocol which is unbiased for $RCP_{3.3.1}$.*

Proof:We prove the theorem by proving the following stronger claim first:

[9]The verification there was not purely formal, due to the lack of specification languages.

There are no 3 sets of 7 hands each, such that: (1) all the 21 hands that appear in these sets are different; (2) every two hands in the same set have one and only one common card; (3) all the cards appear evenly in every set.

Suppose towards contradiction that there exist 3 sets Pa_2, Pa_3, Pa_4 satisfying (1), (2) and (3). Assume without loss of generality that $012 \in Pa_2$, $013 \in Pa_3$ and $014 \in Pa_4$. Since $01x \in Pa_x$ then from (2) and (3) we know that $xab, xcd \in Pa_x$, $0ac, 0bd \in Pa_x$ and $1ad, 1bc \in Pa_x$ such that $ab|cd$ $ac|bd$ and $ad|bc$ are three different partitions of $Dk\backslash\{0,1,x\}$. Since for 4 cards there are only 3 different partitions, we can list all the remaining hands in Pa_2, Pa_3, Pa_4:

$$\begin{array}{lllll} \text{for } Pa_2: & p_1^2 34, p_1^2 56 & p_2^2 35, p_2^2 46 & p_3^2 36, p_3^2 45 \\ \text{for } Pa_3: & p_1^3 24, p_1^3 56 & p_2^3 25, p_2^3 46 & p_3^3 26, p_3^3 45 \\ \text{for } Pa_4: & p_1^4 23, p_1^4 56 & p_2^4 25, p_2^4 36 & p_3^4 26, p_3^4 35 \end{array}$$

where $p_i^x \in \{0,1,x\}$. First, there exists an $x \in \{2,3,4\}$ such that $p_1^x = x$, otherwise there must be either two 056 or two 156 in Pa_2, Pa_3, Pa_4, contradictory to (1). Suppose w.l.g. that $p_1^2 = 2$. It is easy to see that $p_1^3 \neq 3$ and $p_1^4 \neq 4$, otherwise 234 appears twice in Pa_2, Pa_3, Pa_4. Moreover, obviously $p_1^3 \neq p_1^4$. Suppose w.l.g. that $p_2^3 = 3$. Then $p_2^4 \neq 4$ since $346 \in Pa_3$, therefore $p_3^4 = 4$. Now let us fill in the known p_i^x as following:

$$\begin{array}{lllll} \text{for } Pa_2: & 234, 256 & p_2^2 35, p_2^2 46 & p_3^2 36, p_3^2 45 \\ \text{for } Pa_3: & p_1^3 24, p_1^3 56 & 325, 346 & p_3^3 26, p_3^3 45 \\ \text{for } Pa_4: & p_1^4 23, p_1^4 56 & p_2^4 25, p_2^4 36 & 426, 435 \end{array}$$

Now we know that $p_3^2, p_3^3, p_2^4 \in \{0,1\}$ and $p_3^3 \neq p_2^4$ since $p_1^3 \neq p_1^4$. Therefore $p_3^2 = p_3^3$ or $p_3^2 = p_2^4$, but in any case, there will be one hand appear in two announcements, contradiction.
The Theorem follows from above claim and Lemma 2. □

5 Non-deterministic Protocols for Secret-key Generation

In this section, we demonstrate the use of our framework in specifying the non-deterministic protocols, by considering a simplified version of the One-Bit Secret Key exchange protocol in [7].

Example 3 *A deck of cards is distributed randomly to players A, B, E such that A, B, E hold $k+1, 1, k$ cards respectively. Players A and B want to generate a 1-bit secret key by public announcements in the presence of the eavesdropper E according to the following protocol:*
1. Player A announces that "I have one card in $\{x,y\}$" where one card in $\{x,y\}$ is in his hand and the other not.
2. Player B announces that "I have also one card in $\{x,y\}$" if either x or y is his card. Otherwise he announces that " I do not have any card in $\{x,y\}$".
3. If B's announcement is negative then the players proceed by going to step 1 again as if the deck shrinks to its subset without x, y. If B's announcement is positive then A, B should know that A has one card and B has the other in $\{x,y\}$, while E still does not know which belongs to whom. In the end A, B generate one bit secret key by the agreement that bit $= 1$ if A has $max(x,y)$, bit $= 0$ otherwise.

We can give the initial model $\mathcal{M}_{\mathtt{Prot}'}$ and axioms $\mathcal{T}_{\mathtt{Prot}'}$ for the above scenario, similar to $\mathcal{M}_{\mathtt{Prot}}$ and $\mathcal{T}_{\mathtt{Prot}}$in the previous section. Note that the step 3 in the above informal description requires that agents continue "*as if*" the deck of cards (Dk) changes to $Dk\backslash\{x,y\}$, namely agent A will not announce anything she mentioned before. Since after B's negative response, agent A knows the card y is at E's hand and E knows that x is at A's, we can then equally specify the step by adding an epistemic pre-condition for agent A's announcement: she only mentions $\{x,y\}$ if (1) she is sure that E does not know which card in $\{x,y\}$ belongs to her; (2) she does not know that E has a card in $\{x,y\}$ already and (3) she does not know B's card (otherwise the execution of the protocol should be terminated).

Now let $has_A(x \boxplus y)$ be the abbreviation of $((has_A x \wedge \neg has_A y \wedge K_A \neg K_E has_A x \wedge \neg K_A has_E y) \vee (has_A y \wedge \neg has_A x \wedge K_A \neg K_E has_A y \wedge \neg K_A has_E x)) \wedge \bigwedge_{z \in Dk} \neg K_A has_B z$. Let $has_i(x \oplus y)$ be the abbreviation of $has_i x \vee has_i y$. Then the above protocol can be formalized as $\mathtt{Prot}' = \langle \pi_{\mathtt{Prot}'}, \Phi_{\mathtt{Prot}'} \rangle$ where $\pi_{\mathtt{Prot}'}$ is as follows:

$$\Big(\bigcup_{x,y \in Dk} (?!(has_A(x \boxplus y)); (?!has_B(x \oplus y) \cup ?!\neg has_B(x \oplus y)))\Big)^*$$

and $\Phi_{\mathtt{Prot}'} = \{\bigwedge_{x \in Dk}(K_A has_B x \rightarrow \bigvee_{y \in Dk}(K_B has_A y \wedge \neg K_E has_A y \wedge \neg K_E has_B x))\}$.

Note that the protocol $\pi_{\mathtt{Prot}'}$ is non-deterministic and indefinite in length, since for any $x, z \neq y \in Dk$, $has_i(x \boxplus y)$ and $has_i(x \boxplus z)$ may both hold at some state in $\mathcal{M}_{\mathtt{Prot}'}$ where A has x but not y or z. Just like the informal description of the protocol, for any x that A has, she randomly chooses the announcement she wants to make. Also note that in the specification we use * operator to avoid giving a bound of steps of the protocol which depends on the specific k. The verification problem is as follows:

Proposition 10 $\mathcal{M}_{Prot'} \models C[\pi'_{Prot}]\Phi_{Prot'}$.

With some efforts, we may specify the n-bit secret exchange multi-party protocols in [7, 8], based on the above example.

6 Conclusion and Future work

The logical framework for formally specifying and verifying epistemic protocols of this paper made it possible to formally state and verify the crucial requirement that the protocols remain correct even if the protocol and its goals are commonly known, not only to the participants in the protocol but also to eavesdroppers. This fleshes out remarks to that effect in [15, 18, 2].

Future work We have restricted ourselves to protocols involving public announcement only. This guarantees decidability of the model checking for protocols, and decidability of satisfiability for the echo-free part of the language (where no public announcements occur in the scope of an iteration operator). The exact complexity of the model checking problem of this logic is left for further investigation. The restriction also gives some hope for the synthesis problem for restricted forms of epistemic protocols (cf. the ideas mentioned [1]). Some obvious extensions of the language are subgroup announcements and actions for factual change (cf. [17, 13]). Another interesting extension is concurrent action, for modelling simultaneous announcements. In order to verify some non-deterministic protocols where probabilistic choices of announcements play an important role, e.g. the unbiased protocols mentioned in [2], we may need to extend our language with probabilistic programs and modalities about conditional probability.

In any case, the long term goal of our project is to design and analyse an epistemic protocol language that can be used to specify and automatically verify real life protocols for secure communication or social mechanism design, thus exploring a new field of *"Epistemic Engineering"*.

Acknowledgements The authors thank the three anonymous referees for their insightful remarks and comments. The first author is supported by the project *Verification and Epistemics of Multi-Party Protocol Security* funded by the Netherlands Organization for Scientific Research (NWO, project number 612.000.528). The second author's work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. Special thanks to Hans van Ditmarsch for his help with some historic references.

References

[1] T. Agotnes, P. Balbiani, H. van Ditmarsch, and P. Seban. Group announcement logic. 2008.

[2] M. D. Atkinson, H. van Ditmarsch, and S. Roehling. Avoiding bias in cards cryptography. *Australasian Journal of Combinatorics*, 44:3–17, February 2009.

[3] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *Proceedings of the Royal Society of London, Series A, Mathematical and Physical Sciences*, 426(1871):233–271, December 1989.

[4] D. Chaum, C. Crépeau, and I. Damgard. Multiparty unconditionally secure protocols. In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 11–19, New York, NY, USA, 1988. ACM.

[5] R. Fagin, J. Y. Halpern, M. Y. Vardi, and Y. Moses. *Reasoning about knowledge*. MIT Press, Cambridge, MA, USA, 1995.

[6] K. Fine. In so many possible worlds. *Notre Dame J. Formal Logic*, 13(4):516–520, 1972.

[7] M. J. Fischer and R. N. Wright. Multiparty secret key exchange using a random deal of cards. In *In Proc. CRYPTO*, pages 141–155, 1991.

[8] M. J. Fischer and R. N. Wright. Bounds on secret key exchange using a random deal of cards. *Journal of Cryptology , Springer Verlag*, 9:71–99, 1996.

[9] G. Gamow and M. Stern. *Puzzle-math*. Viking Adult, February 1958.

[10] D. Jackson. Alloy: a lightweight object modelling notation. *ACM Trans. Softw. Eng. Methodol.*, 11(2):256–290, April 2002.

[11] J. Miller and L. Moss. The undecidability of iterated modal relativization. *Studia Logica*, 79, April 2005.

[12] S. Petride and R. Pucella. Perfect cryptography, s5 knowledge, and algorithmic knowledge. In *TARK '07: Proceedings of the 11th conference on Theoretical aspects of rationality and knowledge*, pages 239–247, New York, NY, USA, 2007. ACM.

[13] J. van Benthem, J. van Eijck, and B. Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, November 2006.

[14] H. van Ditmarsch. Descriptions of game actions. *J. of Logic, Lang. and Inf.*, 11(3):349–365, 2002.

[15] H. van Ditmarsch. The russian cards problem. *Studia Logica*, pages 31–62, October 2003.

[16] H. van Ditmarsch. Unconditionally secure protocols with card deals, September 2008.

[17] H. van Ditmarsch and B. Kooi. Semantic results for ontic and epistemic change. In G. Bonanno, W. van der Hoek, and M. Wooldridge, editors, *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, pages 87–117, October 2008.

[18] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. (Synthese Library). Springer, 1st edition, November 2007.

[19] G. van Tilburg. Doe wel en zie niet om (do well and don't look back). *Katholieke Illustratie (Catholic Illustrated Journal)*, 90(32):47, 1956.