Automatic Verification of Epistemic Specifications under Convergent Equational Theories

Ioana Boureanu
Ecole Polytechnique Federal de Lausanne
ioana.boureanu@epfl.ch

Andrew V. Jones
Department of Computing
Imperial College London
andrew.jones@ic.ac.uk

Alessio Lomuscio
Department of Computing
Imperial College London
a.lomuscio@ic.ac.uk

ABSTRACT

We present a methodology for the automatic verification of multi-agent systems against temporal-epistemic specifications derived from higher-level languages defined over convergent equational theories. We introduce a modality called rewriting knowledge that operates on local equalities. We discuss the conditions under which its interpretation can be approximated by a second modality that we introduce called empirical knowledge. Empirical knowledge is computationally attractive from a verification perspective. We report on an implementation of a technique to verify this modality with the open source model checker mcmas. We evaluate the approach by verifying multi-agent models of electronic voting protocols automatically extracted from high-level descriptions.

Categories and Subject Descriptors
D.4.6 [Security and Protection]: Verification

General Terms
Security, Verification

Keywords
Epistemic Logic, Equational Rewriting, Model Checking

1. INTRODUCTION

Over the past decade there has been increased interest in developing methodologies for the verification of multi-agent systems (MAS). An approach that has been shown effective is that of symbolic model checking [15, 18] for MAS specified in semantics for temporal-epistemic logic [12]. This has been effectively used in a number of practical applications, including autonomous underwater vehicles [11] and cryptographic protocols [23].

A clear advantage of MAS-based approaches using temporal-epistemic logic is the intuitiveness of the resulting specifications to be checked. Concepts emerging from the MAS community are now being exported to other close disciplines that increasingly see the benefit of using powerful, expressive languages.

One of these areas is security. It has long been recognised [5] that cryptographic protocols can benefit from specifications in which knowledge-based concepts feature prominently. Concepts such as anonymity, privacy and non-repudiation can be both naturally and powerfully expressed in epistemic languages. Influential works in the area include the formulation of secrecy by Halpern et al. [14], advances on algorithmic knowledge [20] and the epistemic modelling of unlinkability [23]. These have found applications in MAS in a variety of ways, including attack detectability [4].

Still, fundamental problems remain. Firstly, the indistinguishability relations to be used when interpreting the knowledge modalities need to account for the cryptographic primitives used in the messages exchanged. For instance, the set of indistinguishable states should be computed by taking into account the agent’s ability to decipher a given message. While some approaches (e.g., [7]) support cryptographic primitives such as encryption and decryption, existing approaches fall short of addressing the more general classes including digital signatures and bit-commitments. Yet, these primitives are prominent in several classes of protocols, e.g., e-voting or zero-knowledge.

Secondly, little or no support for cryptography-inspired modalities is currently provided in existing tools. An extension to mcmas [15] that caters for explicit knowledge exists [16], but we are unaware of any model checker supporting epistemic modalities for cryptographic concepts, or, indeed, other application-driven epistemic modality of use in many MAS settings. In fact, recent approaches [4] have been restricted to protocols in which receivers can decode all messages down to their atomic constituents immediately upon their receipt. This assumption is not natural in many settings including e-voting, where principals are often only able to decipher messages only at the end of a run.

In this paper we develop an approach aimed at overcoming these limitations. Specifically, in Section 2 we define a novel epistemic modality that is interpreted with respect to a general equational theory defining the system. This differs from the standard approach in which the agents’ knowledge is interpreted on the equality of the local states. The high computational cost of deducing equivalence under equational theories has been previously discussed [8]. Thus, in Section 3, we put forward a computationally efficient approximation. Section 4 discusses the implementation of this revised modality on top of the model checker mcmas. In Section 5 we evaluate the techniques presented by verifying e-voting protocols modelled as MAS as per the formalism developed in Section 2. We discuss the results and conclude in Section 6.
2. A TEMPORAL-EPISTEMIC LOGIC FOR SECURITY PROTOCOLS

In this section we introduce a MAS-based semantics for an epistemic logic under convergent equational theories.

2.1 Preliminaries

We assume familiarity with the concepts presented in this subsection; the following is intended to fix the notation only.

Interpreted Systems. The interpreted systems (IS) formalism [10] is a MAS-based semantics for temporal-epistemic logic (CTLK) [12]. We assume a signature \( \Sigma \), the set of \( \Sigma \)-equations. The notation \( \text{ground}\) denotes the sorted set of variables (e.g., symbol \( \text{nat}\)).

\( \text{Env}\) and \( \text{Act}\). An agent \( i\) is described by a set \( L_i\) of possible local states, a set \( Act_i\) of local actions, a local protocol function \( P_i : L_i \rightarrow 2^{Act_i}\), and a local evolution function \( t_i : L_i \times Act_i \times \ldots \times Act_n \times Act_{Env} \rightarrow L_i\). An IS is defined by the set \( G = \prod_{i \leq n} L_i \times L_E\) of global states, the set \( Act = Act_1 \times \ldots \times Act_n \times Act_{Env}\) of joint actions, the joint protocol \( P = (P_1, \ldots, P_n, P_{Env})\) and the global evolution function \( \tau (t_i) = (t_1, \ldots, t_n, t_{Env})\).

2.2 Protocol Model

In this subsection we put forward a technique for producing a fully instantiated interpreted system that models a finite number of protocol sessions running concurrently. The aim of this construction is to obtain interpreted systems in which the epistemic relation for each agent is an equivalence relation under the underlying equational theory formalising the effects of the protocol-primitives executed by each role (i.e., in our presentation, by some agent assuming that role). We assume that all protocols referred to henceforth are specified in CAPSL.

A Communication Protocol \( Pr_1\).

1. \( A \rightarrow B : n\)
2. \( B \rightarrow A : m\)
3. \( A \rightarrow B : \leq(n, m)\)

The protocol \( Pr_1\) describes a set of send-receive rules of the two roles: the A-role and the B-role. An agent assuming the A-role initiates the protocol by sending the term \( n\) to its B-role partner agent. This receiver replies with the term \( m\).

High-level security languages such as Common Authentication Protocol Specification Language (CAPSL) [9] provide precise descriptions of security protocols, including the underlying equational theory formalising the effects of the protocol-primitives executed by each role (i.e., in our presentation, by some agent assuming that role). We assume that all protocols referred to henceforth are specified in CAPSL.
On the algebra A, let the set \( T_{Σ,X} | _{Pr} \) denote the terms used only in the actual description of the protocol Pr. When Pr is implicit, we simply write simply \( T_{Σ,X} \) to mean \( T_{Σ,X} | _{Pr} \). In doing so, we underline the protocol-terms (i.e., messages and their subparts) and, later, their values. Thus, for the protocol Pr1, the set \( T_{Σ,X} | _{Pr1} \) of terms is \( \{ A, B, n, m, ≤ (n, m) \} \). To highlight variables describing a role of the protocol (i.e., variables A and B in Pr1), we introduce an additional sort called role. Variables of sort role (i.e., \( X_{role} \)) range over support set \( A_{role} \). This is taken to be a set of strings, for example \{alice, bob, greg, \ldots \}.

### Protocol Roles to Agents.

An entire protocol-role (e.g., a sender) is described by a null values within a role, we will use designated values, called while the variable written variable role A variables instantiations are as follows: an assignment. Then, by the definition above, the initial role assignment of all terms in the assignment \( t \) on a role \( A \) is implicit, we simply write an \( A \) -sorted set of constant function symbols. When the sort \( T \) nat,nat, the set \( T \) nat,nat of terms is \( \{ n / 3, m / 2, A / alice, B / bob \} \) of the protocol Pr1. We now present the formal description of the agent \( ag_R^δ \).

We consider several concurrent instantiations of each role by different agents. So, a free term (\( \perp \)) representing the role of the sender can later be instantiated to potentially different values, depending on the value received from other agents. A receipt may trigger the instantiations of other local terms as prescribed by the equational theory of the protocol. For instance, in Pr1 with \( δ = [n / 3, m / 2, A / alice, B / bob] \) an A-role participant may receive the value 2 for \( m \) from a B-role agent. Following this, the A-role agent will “apply” the equational theory \( E_1 \) to rewrite the term \( ≤ (m, n) \) to \( ≤ (3, 2), ≤ (2, 1), ≤ (1, 0) \) and, finally, to \( F^1 \). To permit this, the term \( ≤ (m, n) \) of sort Bool, which is free in the role of A, should range over \( A_{nat} × A_{nat} \) and \( A_{bool} = (N × N) \cup \{ T, F \} \). However, the term \( n \) should efficiently range only over \( A_{nat} = N \) for this agent, since \( n \) is bound to the A-role and its initial value cannot be changed. These value-range restrictions optimise the size of the fully instantiated model. The following definition formalises this by giving the possible values of a portion of a message held by an agent during the run.

### Range of a Term for an Agent.

The range \( Range_R(t) \) of a term \( t \in T_{Σ,X} | _{Pr} \) for an \( ag_R^δ \) agent is as follows:

\[
\text{Range}_R(t) = \begin{cases} 
A_s & \text{if } t \in (B_R) \cap X_s \quad (1) \\
A_s \cup A_{\bot} & \text{if } t \in (F_R) \cap X_s \\
A_s \times \ldots \times A_{\bot} \cup A_s & \text{if } t \in (B_R) \cap X_{\bot} \\
A_s \times \ldots \times A_{\bot} \cup A_{\bot} & \text{if } t \in (F_R) \cap X_{\bot} \end{cases}
\]

### Stores and Views for an Agent.

A store for an agent \( ag_R^δ \) is a relation \( t :: Range_R(t) \) between terms and their respective ranges for the agent \( ag_R^δ \).

An initial view for an agent \( ag_R^δ \) encodes an initial \( R \)-role instantiation \( δ \). An non-initial view for \( ag_R^δ \) encodes an actual assignment \( [y/0] \), for some \( y \in (F_R) \), \( v \in A_s \) (i.e., \( v \neq \perp \)).

The store of \( ag_R^δ \) in a model for Pr1 is as follows: \( \text{store}_{ag_R^δ} = (A :: String, B :: String, n :: N, m :: N) \).

\( \leq (n, m) :: (N × N) \cup \{ T, F \} \).

A possible non-initial view for \( ag_R^δ \) in a model for Pr1 is:

\( \text{view}_{ag_R^δ} = (A :: alice, B :: bob, n :: 3, m :: 2, \leq (n, m) :: \perp) \).

The non-initial view \( \text{view}_{ag_R^δ} \) shows that \( ag_R^δ \) has updated the value for \( m \) in its view from the initially held \( \perp \) to the received value 2. In the above, \( ag_R^δ \) has not yet “calculated” the value of \( ≤ (n, m) \), i.e., \( ≤ (n, m) \) is still \( \perp \) in \( \text{view}_{ag_R^δ} \).

To complete the description of instantiated protocol roles, we introduce the set of adjacent terms Adj. These terms are

\( T \) and \( F \) are the concrete values for \( true \) and \( false \).
unrelated to the equational theory $E$, but are induced by the (CAPSL) protocol description (e.g., flag variables to denote protocol steps, stages of rewriting, etc.).

**Local States of Agents.** An (initial) local state is an (initial) view together with certain protocol-driven adjacent terms and their assigned values. The set $L_{ag_R}$ is the set of all possible local states of $ag^δ_R$.

Let step $\in Adj$ be an adjacent atomic term of sort $nat$ (i.e., denoting protocol steps). Then, let $Range_R(step) = \mathbb{N}$ and step $\mapsto 1$ in an initial setup, for any role $R$. An initial state $i_{L}$ and local state $l$ of agent $ag^δ_R$ in a model for $Pr_R$ are as follows:

- $i_{L} = (step \mapsto 1, A \mapsto alice, B \mapsto bob, n \mapsto 3, m \mapsto \perp_{n_i})$
- $l = (step \mapsto 5, A \mapsto alice, B \mapsto bob, n \mapsto 3, m \mapsto 2, \perp_{n_i})$

Let $l \in L_{ag^δ_R}, t \in T_{S_X}$ and $x \in Range_R(t)$. In the following, we use the following notations:

- $l.x$x denotes the view encoded inside the local state $l$
- $l|_{t=x}$ denotes the fact that $l = x$
- $l|_{(x,x)}$ denotes that $l|_{(x,x)}$ and $t = |_{(x,x)}$
- $l|_{step} = \perp_{(T.,F.)}$

In the following, let $i$ denote the map $ag^δ_R$ of an initiator role $R$ and $i'$ denote the map $ag^δ_R$ of a receiver role $R'$.

**Local Actions and Protocol of Agents.** Let step $\in Adj, j \in \{1,2,3\}, nr_j \in Range_R(step)$ and $t, x, t'$ be as above. The set $LAct = \{send(t, x, t'), receive(t, x), \text{rewrite}, \text{empty}\}$ is the set of possible local actions of agent $i$. The local protocol $P_i$ of agent $i$ is as follows: $P_i(l|_{step=nr_j}, t=x,l.R = R'.R') = \{send(t, x, t'), receive(t, x), \text{rewrite}\}$. This gives the set of all possible local states of the Environment agent.

**The Environment Agent.** We assume that the environment agent $Env$ records all communication. Therefore, the local states of the $Env$ agent are given by maps of the form $(t: \cup Range_R(t) :: Ag :: Ag(t) | t \in T_{S_X})$. This gives the set $R_{E}^{\text{role}}$ of possible local states of the Environment agent. The environment only has one possible action denoted by $listen$, which is enabled by its protocol at every local state.

**Global States and Joint Actions.** Let $i \in Ag = \{1, \ldots, n\}, l_i \in LAct_i, l_{Env} \in L_{Env}, a_i \in LAct_i$ and $a_{Env} \in LAct_{Env}$. A global state $g$ is a tuple $(l_1, \ldots, l_n, l_{Env})$. The set $G$ of global states is the set of all possible states $g$ as above. A joint action $a$ is a tuple $(a_1, \ldots, a_n, a_{Env})$. The set $Act$ of joint actions is the set of all possible joint actions $a$ as above.

**Agents’ Local Evolution Function.** Let $i$ denote the $ag^δ_R$ agent, $i'$ denote the $ag^δ_R$ agent as above, let $l \in L_i$ be a local state of agent $i$ and $a \in Act$ be a joint action. The local evolution function $E_i$ of agent $i$ is defined below. In this definition, the preconditions for enabling a state-update upon receipt express the following: 1) the action receive of agent $i$ is synchronised with the action send of agent $i'$ and with the action $listen$ of the $Env$ agent; 2) agent $i$ is in the step $\text{nr}$ where it awaits message $t$; 3) the purported sender is the agent of the $R'$-role$^2$ (i.e., $i.R' = i'.R'$); 4) the values $x_i$

$^2$If protocols use anonymous channels, then this condition is dropped.

of certain subterms $t_j$ in the received term $t$ are consistent with agent $i$’s view, i.e., $l_{step=nr}$. These conditions are inspired by the matching-receive semantics [3, 21].


\[
\begin{align*}
\text{if } l|_{step=nr+1} & \text{ then } \begin{cases}
\text{if } l|_{step=x, t'=i'.R'} & \text{ for } a_i = \text{send}(t, x, i'), a_{Env} = \text{listen}, \\
\text{a}_i' = \text{receive}(t, x) \end{cases} \\
\text{if } l|_{step=x, t'/x} & \text{ for } a_i = \text{receive}(t, x), a_{Env} = \text{listen}, \\
\text{a}_i' = \text{send}(t, x, i'), t_j \in \text{Sub}(t) \\
\text{if } l|_{step=nr+1} & \text{ for } a_i = \text{rewrite}, a_{Env} = \text{listen}, \\
\text{a}_i' = \text{empty}, t \in T_{S_X}, t' = t_E \\
\end{align*}
\]

To illustrate further, let $i = ag^δ_A$ in the $Pr_R$ protocol and, by $E_i$, let the action receive($m, 2$) be performed at the local state $l|_{step=1}$ of agent $i$. The implicit rewriting-driven state-update is: $\leq (3, 2) \rightarrow \leq (\text{succ}(2), \text{succ}(1)) \rightarrow \leq (2, 1) \rightarrow \leq (\text{succ}(1), \text{succ}(0)) \rightarrow \leq (1, 0) \rightarrow F$. For protocols where the intermediate rewriting is not of interest, we collapse such a state-update sequence in one update, i.e., the sequence $l|_{step/3, \leq (n, m)/x, \text{succ}(0)} \rightarrow l|_{step/4, \leq (n, m)/(1, 0)}$ and $l|_{step/5, \leq (n, m)/F}$ is reduced to $l|_{step/3, \leq (n, m)/F}$. The above presentation of the local evolution function $E_i$ formalises such optimisations.

**The Global Evolution Function.** The global evolution function $f : G \times Act \rightarrow G$ is such that $f(g, a) = g'$ if $act \in P_i(g, a), E_i(g, a) = g_i$, for all $i \in Ag \cup \{Env\}$, for $g, g' \in G$ and $a \in Act$.

A path is a sequence of global states described by the global evolution function. Paths naturally define the set of reachable states. Henceforth, $G$ refers to the set of reachable states.

**Equational Interpretated System for $Pr_R$.** An equational interpreted system for $Pr_R$, denoted by $T_{IS}^{Pr_R}$, is a tuple $(G, Act, P, l, I_0, V)$, where the components $Act$ and $t$ are as previously defined, $I_0 \subset G$ is a set of initial global states, $P = \{P_i | i \in Ag \cup \{Env\}\}$, and $V : G \times PV \rightarrow \{\text{true}, \text{false}\}$ is a valuation function for the propositions $PV$ of a logic language.

**Local Satisfaction of Equational Equalities of Terms.** The local state $l \in L_i$ satisfies $t \equiv_{E} t'$, written $l|_{t=x} \equiv_{E} l|_{t'=x'}$ for $t \equiv_{E} t'$, with $t = t_E, t \equiv_{E} l|_{t=x} \equiv_{E} t_E, x'$, and $t \equiv_{E} t'$, if it is the case that $l|_{t=x} \equiv_{E} t'$ if and only if $l'|_{t=x} \equiv_{E} t'$, for all $l \in T_{S_X} |_P, i \notin Adj$.

By the definition above, a local state $l$ satisfies the equality $t \equiv_{E} t'$ of terms modulo $E$ if the term $l$ has been rewritten to the normal term $t'$ in local state $l$ of $T_{IS}^{Pr_R}$.

**Equational Indistinguishability.** Two local states $l_1$ and $l_2$ in $L_i$ are $i$-indistinguishable modulo $E$, written $l_1 \equiv_{E} l_2$, if it is the case that $l_1 \equiv_{E} l_2$ and only if $l_2 \equiv_{E} l_2$, for all $l \in T_{S_X} |_P, i \notin Adj$.

Two reachable global states $g, g' \in G$ are $i$-indistinguishable modulo $E$, written $g \equiv_{E} g'$, if $g, g' \equiv_{E} g_i$. The relation $\equiv_{E} \subseteq G \times G$ is the quotient-indistinguishability relation.

By the definition above, two local states are indistinguishable modulo $E$ if they satisfy the same equalities of terms modulo $E$.

As an example, let $i = ag^δ_A$ in the $T_{IS}^{Pr_R}$ for the protocol $Pr_R$. As none of the following states of $ag^δ_A$ satisfy $\leq (m, n) = E_i, F$, it holds that $l|_{step=2, \leq (m,n)=(3,2)} \equiv_{E_i} l|_{step=2, \leq (m,n)=(2,1)} \equiv_{E_i} l|_{step=2, \leq (m,n)=(1,0)}$. However, the state $l|_{step=5, \leq (m,n)/F}$ does satisfy $\leq (m, n) = E_i, F$. 

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$step=nr+1$</td>
<td>Send</td>
<td>$l</td>
</tr>
<tr>
<td>$step=x, t'/x$</td>
<td>Receive</td>
<td>$a_i = \text{receive}(t, x), a_{Env} = \text{listen}$, $a_i' = \text{send}(t, x, i'), t_j \in \text{Sub}(t)$</td>
</tr>
<tr>
<td>$step=nr+1$</td>
<td>Rewrite</td>
<td>$a_i = \text{rewrite}, a_{Env} = \text{listen}$, $a_i' = \text{empty}, t \in T_{S_X}, t' = t_E$</td>
</tr>
</tbody>
</table>
The interpretation of the knowledge modalities is as follows:

- **Logically Extended Signatures.**
  - Equationally Extended Signatures.
  - The extended equational theory can describe more properties of a logical signature and $\Sigma$.
  - A logically extended signature specifies symbols related to facts, e.g., symbols related to algebraic operators, e.g., $\Sigma_L$.

We use the notation $\mathcal{I}$ both for the equational interpreted system for $Pr$ and the equational multi-agent system model for $Pr$; the context will disambiguate. In our implementation, we optimise the formalism above when generating the $\mathcal{T}_E^{\mathcal{I}}$; this is not discussed here.

### 2.3 The Epistemic Logic CTLK

Let $\mathcal{I}$ be the equational multi-agent system model $M^E_\Delta$ of $Pr$, $p \in PV$ and $i \in A_g \cup \{Env\}$. The specification language CTLKR, used to express the system requirements is defined by the following BNF:

- $\varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid K_i \varphi \mid R_i \varphi \mid AX \varphi \mid AG \varphi \mid A(\varphi U \varphi)$.

The operator $K_i$ denotes the knowledge modality ($K_i \varphi$ reads “agent $i$ knows the fact $\varphi$”) while the operator $R_i$ is the rewriting-knowledge modality ($R_i \varphi$ reads “agent $i$ knows the fact $\varphi$ modulo the equational theory $\Sigma$”). The semantics for CTL on $M^E_\Delta$ is as on standard interpreted systems [12].

The interpretation of the knowledge modalities is as follows:

$$(\mathcal{I}, g) \models K_i \varphi \iff (\text{for all } g' \in G)(g_i = g' \text{ implies } (\mathcal{I}, g') \models \varphi)$$

$$(\mathcal{I}, g) \models R_i \varphi \iff (\text{for all } g' \in G)(g \approx_\Delta^E g' \text{ implies } (\mathcal{I}, g') \models \varphi).$$

The logic CTLK extends the commonly used logic CTLK by means of the rewriting epistemic modality $R$.

### 3. AN APPROACH FOR AUTOMATIC VERIFICATION

We wish to use the logic CTLK as a specification language for model checking security protocols encoded in the MAS-based formalism presented in the previous section. This would enable us to surpass the significant limitations of the state-of-the-art as discussed in the introduction. However, locally parametrised properties of type $t =_E t'$ make the computation of the indistinguishability relation particularly costly, thereby increasing the verification time. To circumvent this, we approximate the $R$ modality and interpret it over an abstraction of $\mathcal{T}_E^\mathcal{I}$ through the use of local predicates. In the following, we will show that important classes of protocols are amenable to analysis through this approximation.

### 3.1 Empirical Interpreted Systems

An $S$-sorted logical signature contains logic symbols of type $[\omega]$ for $\omega \in S^*$. Informally, a (standard) signature specifies symbols related to algebraic operators, e.g., $\text{decrypt}$, whereas a logical signature specifies symbols related to facts, e.g., $\text{isDecrypted}$.

**Logically Extended Signatures.** A logically extended signature is given by a tuple $(\Sigma, \Sigma_L)$, where $\Sigma$ is an $S$-sorted signature and $\Sigma_L$ is an $S$-sorted logical signature.

The tuple $(\Sigma, \Sigma_L, E)$ is the logically extended equational theory corresponding to the equational theory $(\Sigma, E)$. A logically extended equational theory can describe more properties of a protocol than the underlying equational theory alone.

The set $T_{E, \Sigma_L, X}$ of logical terms is defined on logically extended signatures $(\Sigma, \Sigma_L)$ in the same way the set $T_{E, X}$ of terms is defined on the signature $\Sigma$.

The denotation of logically extended signatures is given through a logical extension of the algebra $\mathcal{A}_\perp$. In this extension the interpretation $i.p^{\mathcal{A}_\perp}(\delta)$ of a logical term $p \in T_{E, \Sigma_L, X}$ under assignments $\delta \in \Delta$ is a predicate $p^{\mathcal{A}_\perp}$ evaluated over $\{\text{true}, \text{false}\}$. When $\mathcal{A}_\perp$ is implicit, we simply write $i.p(\delta)$ instead of $i.p^{\mathcal{A}_\perp}(\delta)$.

Let $j$ be an arbitrary agent in an IS formalisation.

**Logical Terms and Experiments of Agents.** A fixed set $In_j \subseteq T_{E, \Sigma_L, X}$ denotes the set of logical terms of agent $j$. The set $\text{InEx}_j = \{i.p(\delta) \mid p \in In_j\}$ of predicates contains the local experiments for agent $j$. An $\text{InEx}$ set is denoted as a local experiment-set. The $Ag$-indexed set $InEx = (InEx_j \mid j \in Ag)$ is the experiment-set.

**Possible Experiments For Agent** $ag_A$ **in** $Pr_1$. The sets $In$ of logical terms and their respective experiment-sets $InEx$ are as follows:

- $In_{1,1} = \{\text{smaller}(n, m)\}$; $InEx_{1,1} = i.\text{smaller}(n, m)(\delta)$;
- $In_{2,1} = \{\text{diffOne}(n, m)\}$; $InEx_{2,1} = i.\text{diffOne}(n, m)(\delta)$.

Similarly to [17], we introduce an indistinguishability relation defined over local predicates, i.e., here on local experiments of agents.

**Local Empirical Indistinguishability.** Two local states $l, l' \in L_j$ are indistinguishable modulo $InEx_j$, or $l \approx^{InEx_j} l'$, if $i.p(\delta) = i.p(\delta')$ for all $p \in In_j$, where $\delta, \delta' \in \Delta$ respectively describe $l$ and $l'$, i.e., $l|_\Delta$ and $l'|_\Delta$. Two global states $g, g' \in G$ are indistinguishable modulo $InEx_j$, written $g \approx^{InEx_j} g'$, if $g_j \approx^{InEx_j} g'_j$. The relation $\approx^{InEx_j} \subseteq G \times G$ is the empirical indistinguishability relation.

Then, two local states $l$ and $l'$ are indistinguishable through experiments if these are evaluated identically at $l$ and at $l'$, as exemplified below.

**Empirical Indistinguishability in a Model for** $Pr_1$. Let $i$ be the agent $ag_A$ in a model for $Pr_1$ and two local states of $ag_A$ respectively described by $\delta[n/g, m/s]$ and $\delta[n/g, m/s]$, i.e., $l_{ag_A}[\delta[n/g, m/s]]$ and $l'_{ag_A}[\delta[n/g, m/s]]$. Let $InEx_{1,1}$ and $InEx_{2,1}$ be the experiment-sets above. Then,

- $i.\text{smaller}(n, m)(\delta[n/g, m/s]) = \text{false}$ and $i.\text{smaller}(n, m)(\delta[n/g, m/s]) = \text{false}$;
- $i.\text{diffOne}(n, m)(\delta[n/g, m/s]) = \text{true}$ and $i.\text{diffOne}(n, m)(\delta[n/g, m/s]) = \text{false}$.

Therefore, $l \approx^{InEx_{1,1}} l'$ holds, but $l \approx^{InEx_{2,1}} l'$ does not hold, i.e., $l \not\approx^{InEx_{2,1}} l'$.

Therefrom, we have just augmented the $\mathcal{T}_{E}^{\mathcal{I}}$ formalisation of protocol executions with local experiments. The resulting models are formally defined below.

**Empirical Equational Interpreted System.** Let $Pr$ be a protocol specified by $(\Sigma, \Sigma_L, E)$ and $\mathcal{I}$ be the equational interpreted system $T_E^{\mathcal{I}}$ for $Pr$. An empirical equational interpreted system is the tuple $\mathcal{T}' = (\mathcal{I}, InEx)$, where $InEx = (InEx_j \mid j \in Ag \cup \{\text{Env}\})$. 

3.2 Extended Protocol Logic

We use $P_i$ as the empirical knowledge modality ($P_i \varphi$ reads “agent $i$ empirically knows the fact $\varphi$”). On an empirical equational multi-agent system model $T' = (I, InEx)$, consider the language $L' = CTLKR \cup P_i$, for $i \in Ag \cup Env$.

The language CTLKR is interpreted on $T'$ as on $I$. Let $g \in G$ be an arbitrary reachable state of $T'$. The interpretation of the empirical knowledge modality is as follows:

$$(T', g) \models P_i \varphi \text{ if and only if } (\text{for all } g' \in G)(g \sim_{InEx}^{\varphi} \text{ implies } (T', g') \models \varphi).$$

The empirical knowledge of an agent refers to the information obtained only by theoretically enquiring the agent’s local predicates, i.e., its experiments.

3.3 Experiments Sets of Convergent Equational Theories

In this section we explicitly link the convergence of the underlying equational theory to the experiments of the agents. Once again, we assume that the normal terms of the theory are encoded in the model, i.e., by using a rewriting system, and that the number of protocol instantiations considered is bounded.

Let $Pr$ be a protocol specified by a convergent equational theory $(\Sigma, E)$, $I$ be the equational interpreted system for $Pr$ and $j$ denote the $ag_{jk}$ agent as before. Let $\Sigma_L$ be a logical signature containing the (special) logical symbols $pred \in \Sigma_L$ of type $\omega$, for all $\omega \in S^*$. Let $t$ be an arbitrary term of type $\omega$, i.e., $t \in T_{\Sigma, X}$.

Predicates for Terms. A logical term $pred(t) \in T_{\Sigma, L, X}$ is a logical term for $t \in T_{\Sigma, X}$. The interpretation $i \cdot pred^E(t)(\delta)$ of a predicate for $t$ in $E$ is always true, i.e., $i \cdot pred^E(t)(\delta) = true$, for all $\delta \in \Delta$.

By the definition above, a predicate $i \cdot pred^E(t)$ for a term $t \in T_{\Sigma, X}$ is true under all assignments $\delta \in \Delta$ for $t$. Since $\delta$ is in $\Delta$, i.e., $\delta$ is not an identity instantiation, it means that $\delta(t) \neq \bot$. In the next definition we use predicates for terms to express special experiments, which simulate the recording of the normal terms.

Local Experiments of Convergent Theories. $InEx_j^E = \bigcup_{t \in T_{\Sigma, X}} \{pred(t') | t' = t \downarrow \} \subseteq T$ is the set of local experiments for the convergent theory $E$ of agent $j$. $\text{InEx}_j^E = \{i \cdot pred^E(t)(\delta) | pr(t) \in InEx_j^E\}$ is set of local experiments of the convergent theory $E$ for agent $j$.

Importantly, one can automatically produce the exact set of experiments of a convergent theory $E$ for a protocol $Pr$ by using the CAPSL description of the protocol, the finite set of instantiations given and the normal terms implied by $E$.

Empirical Equational IS for Convergent Theories. Let $\text{InEx}_j^E$ be the local experiments for the convergent theory $E$ of agent $j$ and $\text{InEx}_j^E = (InEx_j^E_j | j \in Ag)$. An empirical equational interpreted system $T_{IS}^E$ for the convergent theory $E$ is given by the tuple $T' = (I, \text{InEx}_j^E)$.

The unwinding of $T_{IS}^E$ follows as in previous definitions. By the above, the system $T_{IS}^E$ is a special empirical system, i.e., agents “track” normal terms under $E$. We now prove that in these systems the $P_i$ modality coincides with $R_i$.

**Theorem 3.1.** Let $Pr$ be a protocol specified by a convergent theory $(\Sigma, E)$ and $I$ be an $MI^{IS}_E$ model for $E$. Then, $T = \varphi$ if and only if $T = \varphi$, for any $\varphi \in L$, where $\varphi \in L'$ is obtained from $\varphi$ by uniformly substituting $R_i$ for $P_i$, for any $j \in Ag$.

**Proof (sketch).** We only need to prove that $T = R_i \psi$ if $T = P_i \psi$, for some arbitrary $\psi \in L$ and an agent $j = ag_{jk}$ under an initial $R$-role instantiation $\alpha$.

Thus, $T = R_i \psi$ def. of $\models \varphi$ (for all $g' \in G$) implies $(I, g') = \psi$ def. of $\models \varphi$ (for all $g' \in G$) (for all $t \in T_{\Sigma, X}, t' = t \downarrow \psi$) $g' = \psi$ implies $(I, g') = \psi$ (1). Let $\delta, \delta'$ assignments extending the initial $R$-role instantiation $\alpha$ and w.l.o.g. denote the local states in (1) as $g'_{\delta}, \ g'_{\delta'}$. Then, $InEx_j^E = \bigcup_{t \in T_{\Sigma, X}} \{pred(t_{\psi})\}$. $InEx_j^E = \cup_{t \in T_{\Sigma, X}} \{i \cdot pred(t_{\psi})(\delta) = \text{true}\}$. Since the definition def. of $\models \varphi$ (1) from (1) with the above and (4), it follows that: def. of $\models \varphi$ $\varphi \models \psi$.

4. Model Checking Knowledge of Protocol Participants

In this section we present a procedure for model checking empirical knowledge that allows for the specification and verification of standard interpreted systems equipped with local experiment-sets, i.e., not only for the equationally-driven $T_{IS}^E$.

**Algorithm 1 SAT$_\text{E}$(\varphi : FORMULA, j : AGENT) : SET OF STATES**

1. $X \leftarrow \{\neg \varphi\}$
2. $Y \leftarrow X$
3. while $X \neq \emptyset$ do
4. $g \leftarrow X.pop()$
5. $\phi_g \leftarrow \text{true}$
6. for $exp \in InEx_j$ do
7. if $g \in exp$ then
8. $\phi_g \leftarrow \phi_g \land \text{exp}$
9. else
10. $\phi_g \leftarrow \phi_g \land \neg \text{exp}$
11. end if
12. end for
13. $Y \leftarrow Y \cup \{\phi_g\}$
14. $X.remove(\{\phi_g\})$
15. end while
16. return $\neg Y$

The approach for calculating the set $[P_i \varphi]$, i.e., the set of states that satisfy the formula $P_i \varphi$, is shown in Algorithm 1. Lines 8 and 10 construct the formula $\phi_g$ representing the conjunction of the evaluation of experiments for the agent $j$ at the current state $g$. The set $Y$ is constructed iteratively from each $g \in \{\neg \varphi\}$ (the set $X$). At Line 13, $[\phi_g]$ contains the set of states that are empirically indistinguishable from the state $g$ (i.e., $[\phi_g] = \{g' \in G | g' \sim_{InEx_j} g\}$). To calculate $Y = \{g \in G | (\exists g' \in G) (g \sim_{InEx_j} g) \land (g = \neg \varphi)\}$ efficiently, we remove $[\phi_g]$ from $X$ (Line 14) as these states have an identical experiment-set evaluation. At Line 15, $Y$ contains the reachable states that either directly refute $\varphi$, or are empirically indistinguishable from a state that does. Therefore we obtain that $Y = [P_i(\neg \varphi)]$, where $T_{IS}^E(\neg \varphi) = \neg P_i(\neg \varphi)$ is the dual of $P_i(\varphi)$. Finally, at Line 16, the algorithm calculates
Table 1: E-Voting Specifications in CTLKR

<table>
<thead>
<tr>
<th>Specification</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP</td>
<td>$AG(votes(i, v) \rightarrow AG \bigwedge_{v' \neq v} Q_{at}(votes(i, v'))) \bigwedge_{v'}$</td>
</tr>
<tr>
<td>VVU</td>
<td>$AG(votes(i, v) \rightarrow \bigwedge_{v' \neq v} \bigwedge_{i' \neq i} [votes(i', v') \rightarrow AGQ_{at}(votes(i, v') \land votes(i', v))])$</td>
</tr>
<tr>
<td>RF</td>
<td>$AG(votes(i, v) \rightarrow \bigwedge_{v' \neq v} \bigwedge_{i' \neq i} [votes(i', v') \rightarrow AGQ_{at}(votes(i, v') \land votes(i', v) \land votes(i, v'))])$</td>
</tr>
<tr>
<td>CR</td>
<td>$AG(votes(i, v) \rightarrow \bigwedge_{v' \neq v} \bigwedge_{i' \neq i} \bigwedge_{i' \neq i} [votes(i', v') \rightarrow AGQ_{at}(votes(i, v') \land votes(i', v) \land votes(i, v'))])$</td>
</tr>
</tbody>
</table>

$\neg Y$, i.e., the set difference between the set of global states $G$ and $Y$. So, it returns $[P_i(\phi)]$.

**Proposition 4.1.** Algorithm 1 calculates the set of states $[P_i(\phi)]$.

**Implementation.** We have implemented Algorithm 1 as an experimental extension of the model checker mcmas [15]. This extension, titled mcmas-e, is available from [1]. ISPL, the input-language of mcmas, was extended to allow for the definition of experiments at the agent level, as well as to support the specification of empirical knowledge formulae.

5. VERIFYING E-VOTING PROTOCOLS

The applicability of previous research [3] in this line has been limited to protocols for which the specifications can be expressed by using standard notions of knowledge; this included authentication and key-establishment. We herein analyse e-voting protocols, which were out of the scope of [3].

To illustrate that the models and knowledge modalities introduced so far surpass this limit, we analyse more sophisticated e-voting protocols than previously possible with our extension of mcmas.

**E-Voting in the $\mathcal{T}_{IS}^E$ Formalism.** Assume a $\mathcal{T}_{IS}^E$ model and the propositions $votes(j)$ and $votes(j, x)$, representing that an honest agent $j$ has voted and that agent $j$ has voted $x$, respectively. Let $i, i'$ be two different agents. We consider only fair paths representing voting sessions in which eventually both agent $i$ and agent $i'$ vote and that the voting is not unanimous.

The specifications for e-voting requirements we consider, i.e., vote privacy (VP), voter-vote unlinkability (VVU), receipt-freeness (RF) and coercion-resistance (CR), are formalised in Table 1. We use the notation $Q_{j}\phi$ to represent $\neg R_j \neg \phi$, for any agent $j$.

VP stipulates that whenever agent $i$ has voted $v$, there does not exist a point where the attacker $at$ can be sure that it was $i$ who voted $v$. Similarly, VVU expresses that the attacker at will always consider it possible that agents $i$ and $i'$ have swapped votes. RF states that, whenever agent $i$ counterbalances the vote of the receipt-providing agent $i$, the attacker $at$ is not at any point able to link any of the votes to their respective agents. CR is similar to RF, but is analysed on a stronger threat-model. The formulae VVU, RF and CR are inspired by the specifications of total role-interchangeability [23], whereas VP is inspired by the specifications of anonymity in [14] and their extensions to privacy in [23].

We verify these specifications against the FOO’92 e-voting protocol [13]. We formalise the execution of a finite number of concurrent sessions as three, specialised $\mathcal{T}_{IS}^E$ systems. The first model, $M_1$, is a $\mathcal{T}_{IS}^E$ model with an added Attacker agent $(at)$ representing a passive intruder. This model satisfies the vote-privacy property. A receipt-providing agent $i$, and a stronger Attacker are modelled in $M_2$, which specialises $M_1$ and supports receipt-freeness. To model coercion, the formalisation $M_3$ extends $M_2$, with a further enhanced Attacker and a coercible agent $i_c$.

**Experiments.** The high-level description of the FOO’92 protocol was initially provided in CAPSL [9]. This encoding was then passed to an ISPL translator [1]. The translator is an extension of the PD2IS toolkit [3] where the instantiated, 1-enhanced normal terms are inserted into the ISPL models. In this way we can automatically generate the $M_1$, $M_2$ and $M_3$ formalisations of FOO’92, as well as the e-voting requirements in ISPL. The generated ISPL files are in the region of 8000 lines and take approximately 15 seconds to build. mcmas-e was then used to verify these models. The machine employed was an Intel Core 2 Duo processor 3.00 GHz with a 6144 KiB cache running the 32-bit Linux kernel 2.6.32.10. The averaged results obtained across two runs of mcmas-e are summarised in Table 2.

Table 2: Averaged Experiments on FOO’92.

<table>
<thead>
<tr>
<th>Model</th>
<th>Form.</th>
<th>Mem. (KiB)</th>
<th>Time (s)</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>VP/VVU</td>
<td>176832</td>
<td>66441</td>
<td>6.69 · 10^11</td>
</tr>
<tr>
<td>$M_2$ (weakened RF)</td>
<td>175496</td>
<td>66168</td>
<td>6.69 · 10^11</td>
<td></td>
</tr>
<tr>
<td>$M_3$</td>
<td>VP/VVU</td>
<td>181926</td>
<td>70401</td>
<td>6.69 · 10^11</td>
</tr>
</tbody>
</table>

The leftmost column shows the class of model considered. The Memory and Time columns respectively show the average memory usage and the average CPU time in each run. States reports the number of reachable states in each model.

**Discussion of Results.** The Formulae column reports the strongest e-voting specification that was found to hold on the model (strength grows from VP, VVU, RF to CR); vote privacy (VP and VVU) were found to hold on all three classes of models considered. On models $M_2$ and $M_3$ a path was found where eventually the intruder is able to link the receipt-providing agent and its vote, i.e., receipt-freeness (RF) was refuted. Our findings are in-line with known results (i.e., vote-privacy holding for FOO’92). An alternative approach based on applied-pi [8] exhibits similar results. The models verified are of large sizes and are not optimised for e-voting, consequently our verification times are seen as favourable. The complete set of ISPL models and specifications verified are available from [1].

6. CONCLUSIONS AND RELATED WORK

In this paper we have introduced an approach to model checking MAS-based models of security protocols, using specifications expressed in a specialised temporal-epistemic logic. This work surpasses the current state-of-the-art of temporal-epistemic verification of protocols specified as MAS in two
ways. Firstly, it advances a formalism integrating equational theories with epistemic logic. This allows for the modelling of several cryptographic primitives of interest. Secondly, we present an automatic methodology for the verification multi-agent systems-based models against relevant specifications through an open-source dedicated model checker. We emphasise nonetheless that the methodology presented is not directly optimised for e-voting primitives; in fact, it aims at a generic MAS-based verification method.

The empirical indistinguishability relation introduced is related to that of explicit knowledge [17], although in that line no support for cryptographic primitives was available. In [20] agents are empowered with deduction algorithms for generating new local knowledge, but the technical details are different and no automatic technique is discussed. In a theoretical setting of cryptographic modelling, [22] studied the decidability of model checking with respect to an epistemic extension of ATL∗; given the specification language, it is clear that the protocol model, the operators and the semantics in [22] differ from those we present.

Semi-decidable tools have been used to show static equivalence of applied-pi frames modulo certain convergent equational theories [6, 8]. Such approaches could be applied to verify symbolically an infinite number of e-voting sessions. However, they focus mainly on the problem of deciding static equivalence in process calculi (thus a comparison on protocol verification cannot be drawn). Comparatively, we assume a bounded number of fully instantiated protocol sessions where the normal terms of the theory are encoded in the model. Thus, we attain a decidable and fully automatic method of MAS-based protocol verification. In the context of bounded size modelling, the epistemic modalities and indistinguishability relations we have introduced can be correlated to process equivalence [6, 8] and, respectively, static frame-indistinguishability in applied-pi calculus.

Our specifications of e-voting requirements follow the formulations of anonymity in [14, 23] and are model-independent (unlike those in [8], where e-voting specifications are expressed as reachability or process equivalence properties in a model-dependent manner).

Acknowledgments.

The authors acknowledge the support of EPSRC grants EP/E035655/1 and EP/I00520X/1, and the Swiss-funded MICS project.

7. REFERENCES