Towards an awareness-based semantics for security protocol analysis

Rafael Accorsi, David Basin, Luca Viganò

Institut für Informatik
Albert-Ludwigs-Universität Freiburg
Georges-Köhler-Allee 52, D-79110 Freiburg, Germany
{accorsi,basin,luca}@informatik.uni-freiburg.de
http://www.informatik.uni-freiburg.de/~{accorsi,basin,luca}

Abstract

We report on work-in-progress on a new semantics for analyzing security protocols that combines complementary features of security logics and inductive methods. We use awareness to model the agents' resource-bounded reasoning and, in doing so, capture a more appropriate notion of belief than those usually considered in security logics. We also address the problem of modeling interleaved protocol executions, adapting ideas from inductive methods for protocol verification. The result is an intuitive, but expressive, doxastic logic for formalizing and reasoning about attacks. As a case study, we use awareness to characterize, and demonstrate the existence of, a man-in-the-middle attack upon the Needham-Schroeder Public Key protocol. This is, to our knowledge, not only the first doxastic analysis of this attack but also the first practical application of an awareness logic. Even though defining the awareness sets of the agents, a task that is left unspecified in formal works on awareness logics, turns out to be surprisingly subtle, initial results suggest that our approach is promising for modeling, verifying and reasoning about security protocols and their properties.

1 Introduction

1.1 Context

Security protocols describe how agents should exchange messages to achieve security goals such as confidentiality and integrity of data, or authentication of the identity of agents in the network. Although security protocols often consist of only a few message exchanges, designing correct protocols has been likened to "programming Satan's computer" [2] as the protocols should work in the presence of a hostile, powerful opponent who can read and alter messages at will.

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

A number of formal methods have been proposed for rigorously analyzing security protocols. Some of these methods are based on specialized *security logics*, such as the foundational BAN logic for authentication protocols (see [1,5] as well as [15,17] and the other logics discussed there). The idea behind these logics is to formalize the doxastic or epistemic reasoning of agents executing a protocol. More precisely, the logics provide constructs, axioms and inference rules for expressing the basic notions of security protocols (e.g. secrecy of data and "goodness" of cryptographic keys) and for describing how the beliefs or the knowledge of the agents involved in a protocol execution evolve as messages are exchanged.

Despite their successes in detecting protocol design flaws, security logics leave much to be desired. In particular, they suffer from two major drawbacks:

(i) They are often quite limited in the kinds of analysis they support.

(ii) They often lack an appropriate semantics for the notion of belief.

To illustrate (i), observe that attacks such as the "man-in-the-middle" attack upon the Needham-Schroeder Public Key (NSPK) protocol [11] occur when the attacker participates in two interleaved protocol executions, but BAN (and some of its extensions) is not able to model interleaved executions as it only considers single executions by design.

Abadi and Tuttle [1] give a semantics for an extension of BAN that considers multiple executions; it is however quite unclear how to use their logic to model the beliefs of agents in interleaved executions (especially since they do not give applications of their logic). Most importantly, Abadi and Tuttle "give a possible-worlds definition of belief as a form of resource-bounded, defeasible knowledge" [1, p. 202], meaning that an agent's beliefs at an execution state are expressed in terms of the states that are indistinguishable to him from there, i.e. the states in which he has the same information. But, as stated in (ii) above, this semantics and, similarly, those for other extensions of BAN, e.g. [4,10], are unsatisfactory: an agent may reason about, and believe in, any formula in the language, even a formula containing terms that the agent may ignore such as the name or public key of another agent in the network. In fact, an agent may even reason about protocol executions he is not participating in.

An alternative way of reasoning about security protocols is to consider protocols as sets of all possible communication traces. There are several ways in which this idea can be formalized. For example, Paulson [14] uses inductive definitions in higher-order logic. In his work, a protocol plus an attacker model correspond to a set of rules expressing the observable events (i.e. exchange of messages) that can occur in the network; the closure under such rules yields an inductively defined set of traces modeling all the possible communications between the agents as well as the attacker's interference. Paulson uses these models for verification: he interactively proves, by induction, that violations of security properties (i.e. some bad situation, such as an attacker learning the private key of an agent) cannot occur in any trace. Similar inductive definitions are used by Basin in [3], where they provide the basis for automatic, infinite-state, model-checking of protocols using lazy evaluation in the programming language Haskell. In a nutshell, inductive methods are successful in finding attacks in security protocols, but characterizing attacks as properties of states or traces can be tricky and require ingenuity (depending on the particular model and the kinds of attacks one is trying to formalize).

1.2 Contribution

In this paper we report on work-in-progress on an awareness-based semantics for the analysis of security protocols that combines complementary features of security logics and inductive methods, as well as features of traditional approaches to doxastic reasoning in multi-agent systems. Our goal is to obtain an intuitive, but expressive, logic for modeling, verifying and reasoning about security protocols and their properties.

The central ideal is that of *awareness* [8,9,18]. The notion of awareness has been used to address the *logical omniscience* problem, i.e. the problem of agents having unbounded reasoning power, which is a characteristic of most doxastic/epistemic logics. Technically, awareness is a filter defining, for each of the agents, the set of formulas he is able (and, possibly, entitled) to reason about, thus limiting his reasoning power. We use awareness to model the resource-bounded reasoning of the agents participating in security protocols and, in doing so, capture a more appropriate notion of belief than those usually provided in security logics: while *implicit belief* expresses the standard, logically omniscient notion of belief, *explicit belief* restricts an agent's implicit beliefs to the formulas that he is aware of. In other words, in our approach resource-bounded reasoning is a consequence of the possible incompleteness of the awareness set of the agents, who are aware of a proper subset of the logical language, implying that the formulas they explicitly believe in are a subset of the formulas they implicitly believe in. Combining this with the inductive approach allows us to formalize and reason about attacks in possibly interleaved protocol executions in a simple and intuitive way in terms of the awareness and beliefs of the agents.

As a case study, we formalize the NSPK protocol in our logic and use awareness to characterize, and demonstrate the existence of, the man-in-the-middle attack shown in Fig. 1 (where "$\circ$" and $\{|M|\}_K$ denote concatenation and encryption of M under K respectively; we assume familiarity with the protocol and the attack and refer to, e.g., [11] for details). To our knowledge, this is the first doxastic (or epistemic) analysis of this attack, as well as the first practical application of an awareness logic. Even though defining the awareness sets of the agents, a task that is left unspecified in formal works on awareness logics, turns out to be surprisingly subtle, initial results suggest that our approach is promising for modeling, verifying and reasoning about

$$
\begin{array}{ll}
 & ev_1 = A \to \text{Spy} : \{\!|A \circ N_1|\!\}_{K_{\text{Spy}}} \\
\mathsf{ns}_1 = A \to B : \{\!|A \circ N_1|\!\}_{K_B} & ev_2 = \text{Spy} \to C : \{\!|A \circ N_1|\!\}_{K_C} \\
\mathsf{ns}_2 = B \to A : \{\!|N_1 \circ N_2|\!\}_{K_A} & ev_3 = C \to A : \{\!|N_1 \circ N_2|\!\}_{K_A} \\
\mathsf{ns}_3 = A \to B : \{\!|N_2|\!\}_{K_B} & ev_4 = A \to \text{Spy} : \{\!|N_2|\!\}_{K_{\text{Spy}}} \\
 & ev_5 = \text{Spy} \to C : \{\!|N_2|\!\}_{K_C}
\end{array}
$$

Fig. 1. The NSPK protocol (left) and the man-in-the-middle attack (right)

security protocols and their properties.

1.3 Organization

We proceed as follows. In §2 we present our logic: its syntax, model of computation, and its awareness-based possible-worlds semantics. In §3 we give the rules for generating the set of formulas an agent is aware of and in §4 we show how to use the logic to analyze the NSPK protocol. We discuss related work in §5, and in §6 we draw conclusions and discuss future work.

2 An awareness-based security logic

2.1 Syntax

Let the sets $\mathcal{P}$ of *primitive propositions* and $\mathcal{T}$ of *primitive terms* be given, where $\mathcal{T}$ consists of three disjoint subsets: $\mathcal{T}_I$ of *agent identifiers*, $\mathcal{T}_K$ of *cryptographic keys*, and $\mathcal{T}_N$ of *nonces*.[1] The sets of messages and formulas are built as follows.

Definition 2.1 The set $\mathcal{M}$ of messages is the smallest set closed under the following rules: (i) $M \in \mathcal{M}$ if $M \in \mathcal{T}$; (ii) $M \circ M' \in \mathcal{M}$ if $M, M' \in \mathcal{M}$; and (iii) $\{\!|M|\!\}_K \in \mathcal{M}$ if $M \in \mathcal{M}$ and $K \in \mathcal{T}_K$.

The set $\mathcal{F}$ of formulas is the smallest set closed under the following rules: (i) $\perp \in \mathcal{F}$; (ii) $p \in \mathcal{F}$ if $p \in \mathcal{P}$; (iii) $\varphi \to \psi \in \mathcal{F}$ if $\varphi \in \mathcal{F}$ and $\psi \in \mathcal{F}$; (iv) $\mathsf{says}_A(B, M)$, $\mathsf{sees}_A(M)$, $\mathsf{has}_A(M)$, $\mathsf{sec}_{\mathcal{G}}(M) \in \mathcal{F}$ if $A, B \in \mathcal{T}_I$, $M \in \mathcal{M}$, and $\mathcal{G} \subseteq \mathcal{T}_I$; and (v) $\mathcal{A}_A\varphi$, $\mathcal{B}_A\varphi$, $\mathcal{I}_A\varphi \in \mathcal{F}$ if $A \in \mathcal{T}_I$ and $\varphi \in \mathcal{F}$. ■

The operators represent the logical consequences of the message exchange between the agents. Intuitively, $\mathsf{says}_A(B, M)$ denotes agent A's saying $M \in$

[1] We assume an underlying algebra where $(K^{-1})^{-1} = K$ for all keys $K \in \mathcal{T}_K$, and the function $\cdot^{-1} : \mathcal{T}_K \to \mathcal{T}_K$ maps a key K to its inverse key K^{-1}. We consider here (asymmetric) public key protocols; for protocols employing (symmetric) shared keys we also have $K^{-1} = K$. Note also that we introduce sorts to distinguish different kinds of data and all the terms have a unique sort. Although this simplifies the encoding, it prevents us from modeling type-flaw attacks where, for example, an agent accepts a nonce as a session-key. An extension to analyze this kind of attack should be possible but remains as future work.

$$\frac{}{\langle\rangle \in P}\,\mathsf{empty} \qquad \frac{t \in P \quad N_1 \notin \mathit{used}\ t}{t, A \to B : \{\!|A \circ N_1|\!\}_{K_B} \in P}\,\mathsf{ns}_1$$

$$\frac{t \in P \quad N_2 \notin \mathit{used}\ t \quad A' \to B : \{\!|A \circ N_1|\!\}_{K_B} \in t}{t, B \to A : \{\!|N_1 \circ N_2|\!\}_{K_A} \in P}\,\mathsf{ns}_2$$

$$\frac{t \in P \quad A \to B : \{\!|A \circ N_1|\!\}_{K_B} \in t \quad B' \to A : \{\!|N_1 \circ N_2|\!\}_{K_A} \in t}{t, A \to B : \{\!|N_2|\!\}_{K_B} \in P}\,\mathsf{ns}_3$$

$$\frac{t \in P \quad X \in \mathsf{synth}(\mathsf{analz}(\mathsf{spies}(t)))}{t, \mathrm{Spy} \to B : X \in P}\,\mathsf{attacker}$$

Fig. 2. The NSPK protocol as an inductively defined set

$\mathcal{M}$ to agent B, $\mathsf{sees}_A(M)$ denotes agent A's seeing M, $\mathsf{has}_A(M)$ denotes A's possessing M, and $\mathsf{sec}_{\mathcal{G}}(M)$ denotes that M is a secret known only to the agents in the group $\mathcal{G}$. The operators $\mathcal{A}_A$, $\mathcal{B}_A$ and $\mathcal{I}_A$ denote the *awareness*, *explicit belief* and *implicit belief* of an agent A. Other operators can be defined in the usual manner, e.g. negation $\neg\varphi \equiv \varphi \to \bot$ and conjunction $(\varphi \wedge \psi) \equiv \neg(\neg\varphi \vee \neg\psi)$.

We adopt the following notation: the variables A, B and C range over agent identifiers, the constant $\mathrm{Spy} \in \mathcal{T}_I$ denotes the *attacker* agent, *groups* of agents are denoted by $\mathcal{G}$, and Greek letters denote *formulas*. All variables may be annotated by subscripts or superscripts.

2.2 Model of computation

The model of computation we employ combines ideas from inductive methods for protocol verification [3,14] with ideas from security logics [1,5] and from traditional approaches to doxastic/epistemic reasoning in multi-agent systems [8,9,18]. More specifically, we combine Paulson's model of a protocol as an inductively defined set of traces with an appropriate notion of awareness to formalize protocol attacks in terms of the awareness and the beliefs of the participating agents.

From the inductive method we take the notions of *event* and *trace*: each step of a protocol corresponds to an event, and traces are event sequences resulting from any interleaving of (possibly partial) protocol executions.

Definition 2.2 A *trace* is a sequence of *events*, where each event is a message exchange of the form $A \to B : M$. The set τ of traces is defined inductively as follows: $\langle\rangle \in \tau$, and if $t \in \tau$ and $A \to B : M$ is an event, then $t, (A \to B : M) \in \tau$ where "," denotes concatenation. ■

A protocol can then be modeled as a subset $P \subseteq \tau$ of traces. Paulson inductively defines P in terms of the closure of a set of rules representing the protocol steps and possible actions by the attacker. For example, the rules used in [14] to build the set P corresponding to the NSPK protocol are given in Fig. 2 (we adapted the syntax to be consistent with ours). These rules model how, when the conditions above the line are met, traces are extended with the new events below the line. The rule empty starts off the inductive definition: the empty trace always belongs to P. The rules ns_1–ns_3 model the identically named steps of the protocol. In ns_2, for example, a trace $t \in P$ can be extended with $B \to A : \{\!|N_1 \circ N_2|\!\}_{K_A}$ whenever N_2 has not been used in t (i.e. it is a fresh nonce) and t contains an event $A' \to B : \{\!|A \circ N_1|\!\}_{K_B}$ (possibly resulting from an application of the rule ns_1, or from an application of the $\mathsf{attacker}$ rule). The superscript in A' means that B received the message $\{\!|A \circ N_1|\!\}_{K_B}$ from some agent, but not necessarily from A. It could be any agent masquerading as A. Hence B cannot, from this message alone, determine who sent it.

The $\mathsf{attacker}$ rule formalizes the attacker model of Dolev and Yao, who assume the following about the attacker: "(a) he can obtain any message passing through the network; (b) he is a legitimate user of the network, and thus in particular can initiate a conversation with any other user; and (c) he will have the opportunity to be a receiver to any user A" [7, p. 199]. We follow Paulson and model these assumptions by defining in the $\mathsf{attacker}$ rule that the Spy can say anything that he can *synthesize* from the *analyzable* parts of the messages he *spies* from the network. The rule employs the operators synth and analz, which, like the operator parts defined below, compute particular extensions of a set $\mathbf{M}$ of messages: $\mathsf{parts}(\mathbf{M})$ is the set of all sub-messages of (the messages in) $\mathbf{M}$, $\mathsf{analz}(\mathbf{M})$ consists of all the messages that can be analyzed from $\mathbf{M}$, and $\mathsf{synth}(\mathbf{M})$ contains all the messages that can be synthesized from $\mathbf{M}$. Finally, $\mathsf{spies}(t)$ is the set consisting of all messages that have been sent in a trace t, which formalizes the assumption that the attacker has control over the network.

Since we also use parts, analz and synth in our model of computation, we provide their formal definitions as follows:

Definition 2.3 Let $\mathbf{M}$ be a set of messages. The set $\mathsf{parts}(\mathbf{M})$ is the smallest extension of $\mathbf{M}$ obtained by adding the components of compound messages and the bodies of encrypted messages:

$$\frac{M \in \mathbf{M}}{M \in \mathsf{parts}(\mathbf{M})}\ \mathsf{parts\text{-}inj} \qquad \frac{M_1 \circ M_2 \in \mathsf{parts}(\mathbf{M})}{M_i \in \mathsf{parts}(\mathbf{M})}\ \mathsf{parts}\text{-}i\ (i \in \{1,2\})$$

$$\frac{\{\!|M|\!\}_K \in \mathsf{parts}(\mathbf{M})}{M \in \mathsf{parts}(\mathbf{M})}\ \mathsf{parts}\text{-body}$$

The set $\mathsf{analz}(\mathbf{M})$ is the smallest extension of $\mathbf{M}$ closed under projection and decryption by keys in $\mathsf{analz}(\mathbf{M})$:

$$\frac{M \in \mathbf{M}}{M \in \mathsf{analz}(\mathbf{M})}\ \mathsf{analz}\text{-inj} \qquad \frac{M_1 \circ M_2 \in \mathsf{analz}(\mathbf{M})}{M_i \in \mathsf{analz}(\mathbf{M})}\ \mathsf{analz}\text{-}i\ (i \in \{1,2\})$$

$$\frac{\{\!|M|\!\}_K \in \mathsf{analz}(\mathbf{M}) \quad K^{-1} \in \mathsf{analz}(\mathbf{M})}{M \in \mathsf{analz}(\mathbf{M})}\ \mathsf{analz}\text{-dec}$$

The set $\mathsf{synth}(\mathbf{M})$ is the smallest extension of $\mathbf{M}$ including agent identifiers and closed under pairing and encryption:

$$\frac{M \in \mathbf{M}}{M \in \mathsf{synth}(\mathbf{M})}\ \mathsf{synth}\text{-inj} \qquad \frac{M_1 \in \mathsf{synth}(\mathbf{M}) \quad M_2 \in \mathsf{synth}(\mathbf{M})}{M_1 \circ M_2 \in \mathsf{synth}(\mathbf{M})}\ \mathsf{synth}\text{-pair}$$

$$\frac{M \in \mathsf{synth}(\mathbf{M}) \quad K \in \mathsf{synth}(\mathbf{M})}{\{\!|M|\!\}_K \in \mathsf{synth}(\mathbf{M})}\ \mathsf{synth}\text{-enc}$$

■

From the doxastic/epistemic approach to modeling multi-agent systems and security protocols [1,5,8,9] we take the notions of *local* and *global state*. At any point during a protocol execution, each agent has a local state, which represents the agent's internal, persistent information. A global state is a tuple consisting of the local states of all the agents, including that of the attacker. It provides an abstract snapshot of the whole system at that point (although note that we do not model an explicit notion of time but simply consider the "time points" generated by the sequences of events in the traces). We first define the structure of a global state and afterwards explain how to construct global states from traces.

Definition 2.4 The *local state* of an agent $A \in \mathcal{T}_I$ is a pair consisting of the set of actions that A has performed and the set of messages in A's possession. A *global state* w is an n-tuple of local states, where n is the number of agents in the system including the attacker. ■

The *actions* that an agent A can perform are *sending* a message M to another agent B, in symbols $\mathsf{Send}_A(B, M)$, and *receiving* a message M, in symbols $\mathsf{Rec}_A(M)$, where the identity of the sending agent is not know a priori (it may be derivable in some cases, e.g., after an authentication protocol).

We now combine the notions of trace and state: we define functions that, given a trace, compute the local state of each agent participating in the (possibly partial, interleaved) protocol executions in that trace. The global state is obtained by tupling the resulting local states.

Definition 2.5 Given $t \in P$, the sets of actions and possessions of an agent

A are recursively defined using the functions $Ac_A(t)$ and $Po_A(t)$ as follows:

$$Ac_A(B \to C : M, evs) = \begin{cases} \{\mathsf{Send}_B(C, M)\} \cup Ac_A(evs) & \text{if } A = B \\ \{\mathsf{Rec}_C(M)\} \cup Ac_A(evs) & \text{if } A = C \\ \{\mathsf{Send}_B(C, M), \mathsf{Rec}_C(M)\} \cup Ac_A(evs) & \text{if } A = \text{Spy} \\ Ac_A(evs) & \text{otherwise} \end{cases}$$
$$Ac_A(\langle\rangle) = \emptyset$$

$$Po_A(B \to C : M, evs) = \begin{cases} \{M\} \cup Po_A(evs) & \text{if } A \in \{B, C, \text{Spy}\} \\ Po_A(evs) & \text{otherwise} \end{cases}$$
$$Po_A(\langle\rangle) = \mathit{initState}(A)$$

where *evs* ranges over event sequences and *initState* is a protocol-dependent function that returns the information that an agent possesses in an empty trace (e.g. his private and public keys, and the public keys and identifiers of some other agents).

So, given a trace $t \in P$, the local state $s_A(t)$ of an agent A is simply $\langle Ac_A(t), Po_A(t)\rangle$, and the global state w is the n-tuple of the local states $s_A(t)$ for all n agents. Given a global state w, we will (overloading notation) write $s_A(w)$ to denote the local state of an agent A at w, and $Ac_A(w)$ and $Po_A(w)$ to denote the two components of $s_A(w)$. ■

Hence, the Spy's local state contains the actions performed by all the agents, as well as the messages they exchange, while the local state $s_A(w)$ of an agent A different from Spy is built only from the events that A participated in. Note that since the Spy possesses all the messages sent in the network, $Po_{\text{Spy}}(w)$ captures the same information as the set spies used in [14] to formalize the attacker's control over the network, and thus expresses Dolev and Yao's attacker model in our setting.

In our approach we will also need to relate agents and messages at particular global states. For example, given an agent A and a message M, we will need to compute the sub-messages of M that A can possess, or express that A has used M' to build a message M. We cannot use the operators synth and analz to capture these notions: analz, for instance, would not compute that $M \in \mathsf{analz}(\{\{\!|M|\!\}_K\})$ even if agent A had the appropriate key since $K^{-1} \notin \mathsf{analz}(\{\{\!|M|\!\}_K\})$. We thus introduce the operators submsg and comp: $\mathsf{submsg}_A(w, M)$ is the set consisting of all sub-messages that A can obtain from M given the keys he possesses at a global state w, while the set $\mathsf{comp}_A(w, M)$ contains all the sub-messages that A used to construct the message M at w.

Definition 2.6 The set $\mathsf{submsg}_A(w, M)$ is defined as follows:

$$\frac{M \in Po_A(w)}{M \in \mathsf{submsg}_A(w, M)}\ \text{sub-inj} \qquad \frac{M_1 \circ M_2 \in \mathsf{submsg}_A(w, M)}{M_i \in \mathsf{submsg}_A(w, M)}\ \text{sub-}i\ (i \in \{1, 2\})$$

$$\frac{\{\!|M'|\!\}_K \in \mathsf{submsg}_A(w, M) \quad K^{-1} \in KeysOf_A(w)}{M' \in \mathsf{submsg}_A(w, M)}\ \text{sub-dec}$$

where $KeysOf_A(w) = \{K \mid K \in \mathsf{analz}(Po_A(w)) \text{ and } K \in \mathcal{T}_K\}$ is the set of keys that an agent A possesses at a global state w.

The set $\mathsf{comp}_A(w, M)$ is defined as

$$\{M' \mid M' \in \mathsf{analz}(Po_A(w)), M \in \mathsf{synth}(Po_A(w)) \text{ and } M' \in \mathsf{parts}(\{M\})\} .$$

Observe that both comp and submsg are finite, computable, sets. ■

We will now exploit this model of computation to introduce our awareness-based possible-worlds semantics for reasoning about agents' beliefs during interleaved protocol executions.

2.3 Semantics

We begin by fixing a set $\overline{\mathcal{T}}_I$ of *agent names*, where, for simplicity, we identify its elements with the previously defined set $\mathcal{T}_I$ of agent identifiers; thus, from now on we will simply talk of *agents*.

A subset $P \subseteq \tau$ of traces modeling a protocol gives rise to a model $\mathfrak{M} = (W, \sim, \pi, \alpha)$, where W is a non-empty set of global states (or worlds), $\sim$ is an agent-indexed family of equivalence relations on W, π is a valuation function $\pi : \mathcal{P} \to \wp(W)$, and α is an agent-indexed family of *awareness functions* $\alpha : (\overline{\mathcal{T}_I} \times W) \to \mathcal{F}$.

We define how to compute α_A in §3 below; informally, if $\varphi \in \alpha_A(w)$ then agent A is aware of φ at the global state w. The family of relations $\sim$ captures *indistinguishability*: two global states are indistinguishable to an agent A iff the local state of A is the same at these two global states. More formally, $w \sim_A w'$ iff $s_A(w) = s_A(w')$, i.e. $Ac_A(w) = Ac_A(w')$ and $Po_A(w) = Po_A(w')$.

Definition 2.7 Truth of a formula φ at a global state w in a model $\mathfrak{M} = (W, \sim, \pi, \alpha)$, in symbols $\mathfrak{M}, w \models \varphi$, is the smallest relation satisfying:

$$
\begin{aligned}
\mathfrak{M}, w \models p &\text{ iff } w \in \pi(p), \text{ for a primitive proposition } p \in \mathcal{P} \\
\mathfrak{M}, w \models \varphi \to \psi &\text{ iff } \mathfrak{M}, w \not\models \varphi \text{ or } \mathfrak{M}, w \models \psi \\
\mathfrak{M}, w \models \mathsf{has}_A(M) &\text{ iff } M \in \mathsf{analz}(Po_A(w)) \\
\mathfrak{M}, w \models \mathsf{sees}_A(M) &\text{ iff } \mathsf{Rec}_A(M') \in Ac_A(w) \text{ and} \\
&\phantom{\text{ iff }} M \in \mathsf{submsg}_A(w, M') \text{ for some } M' \\
\mathfrak{M}, w \models \mathsf{says}_A(B, M) &\text{ iff } \mathsf{Send}_A(B, M') \in Ac_A(w) \text{ and} \\
&\phantom{\text{ iff }} M \in \mathsf{comp}_A(w, M') \text{ for some } M' \\
\mathfrak{M}, w \models \mathsf{sec}_{\mathcal{G}}(M) &\text{ iff } \mathfrak{M}, w \models \mathsf{has}_A(M) \text{ for all } A \in \mathcal{G} \text{ and} \\
&\phantom{\text{ iff }} \mathfrak{M}, w \not\models \mathsf{has}_B(M) \text{ for all } B \notin \mathcal{G} \\
\mathfrak{M}, w \models \mathcal{A}_A\varphi &\text{ iff } \varphi \in \alpha_A(w) \\
\mathfrak{M}, w \models \mathcal{I}_A\varphi &\text{ iff } \mathfrak{M}, w' \models \varphi \text{ for all } w' \text{ such that } w \sim_A w' \\
\mathfrak{M}, w \models \mathcal{B}_A\varphi &\text{ iff } \mathfrak{M}, w \models \mathcal{A}_A\varphi \text{ and } \mathfrak{M}, w \models \mathcal{I}_A\varphi
\end{aligned}
$$

We write $\mathfrak{M} \models \phi$ iff $\mathfrak{M}, w \models \phi$ for all $w \in W$. ■

In other words, at a global state w, an agent A *has* M iff M is an analyzable message in his set of possessions, he *sees* M iff he received an M' such that M is a readable sub-message of M', and he *says* M to an agent B iff he sent an M' to B such that M was used to compose M'. A message M is a *secret* shared among the agents in a group $\mathcal{G}$ at w iff at w all the agents in $\mathcal{G}$ possess M and there is no agent not in the group who possesses M. Furthermore, an agent A is *aware* of a formula φ at w iff φ is an element of the set $\alpha_A(w)$, which we define in §3 below, A *implicitly believes* in φ at w iff φ holds in all the worlds indistinguishable to A from w, and A *explicitly believes* in a formula φ at w iff at w he is aware of φ and implicitly believes in it.

3 Awareness

Awareness defines the set of formulas that an agent is able (and, possibly, entitled) to reason about, and thereby limits his reasoning power. Defining an appropriate set of rules to compute $\alpha_A(w)$, i.e. the set formulas that an agent A is aware of at a world w, turned out to be surprisingly subtle. One of the reasons for this is that there are no prior applications of awareness logics or guidelines for generating awareness rules in the literature. More importantly, the problem is that the awareness set will typically be not only incomplete, meaning that it is possible that neither one of φ and $\neg\varphi$ is in the set, but also possibly inconsistent, meaning that both φ and $\neg\varphi$ can be in the set [8,9,18].

Moreover, awareness does not imply truth, i.e. it is not necessarily the case that if $\mathfrak{M}, w \models \mathcal{A}_A\varphi$ then $\mathfrak{M}, w' \models \varphi$ for some w'. That is, an agent might not only be unaware of some true facts, but he might also be aware of some contradictory or false facts. [2] While these are desirable properties, since it is on them that we base our method for defining and finding attacks, they make typical logical criteria (e.g. constructions based on completeness and/or consistency) inappropriate for computing the awareness set.

Our approach to the problem of computing $\alpha_A(w)$ has been to distinguish between the different forms of reasoning that agents perform to become aware of formulas, and identify means to express them. In particular, agents are aware of the formulas representing their local states, from which they can infer other formulas they should also be aware of. These ideas give rise to three kinds of rules: the *basic rules* mirror the semantic definitions of says, sees and has given above and thereby capture the agents' reasoning about their own actions and possessions, while the contents of the messages are used in the rules expressing agents' inferences about *secrecy* and *identification*. Note that these rules generate inductively the awareness set; they are part of the semantic definition, as opposed to being part of a deductive system.

As an example, we give now the awareness rules we used for reasoning about the NSPK protocol.

Definition 3.1 The *basic rules* represent the fact that agents are aware of their local states:

$$\frac{\mathsf{Send}_A(B, M') \in Ac_A(w) \quad M \in \mathsf{comp}_A(w, M')}{\mathsf{says}_A(B, M) \in \alpha_A(w)} \ \mathrm{basic}_1$$

$$\frac{\mathsf{Rec}_A(M') \in Ac_A(w) \quad M \in \mathsf{submsg}_A(w, M')}{\mathsf{sees}_A(M) \in \alpha_A(w)} \ \mathrm{basic}_2$$

$$\frac{M \in \mathsf{analz}(Po_A(w))}{\mathsf{has}_A(M) \in \alpha_A(w)} \ \mathrm{basic}_3$$

The *secrecy rules* define how an agent becomes aware that he is sharing a secret with another agent and how he explicitly believes in that:

$$\frac{\mathsf{says}_A(B, M) \in \alpha_A(w) \quad M \in \mathcal{T}_N}{\mathsf{sec}_{\{A,B\}}(M) \in \alpha_A(w)} \ \mathrm{sec}_1$$

$$\frac{\mathsf{says}_B(A, M) \in \alpha_A(w) \quad \mathsf{sec}_{\{A,B\}}(M) \in \alpha_A(w)}{\mathcal{B}_A\mathsf{sec}_{\{A,B\}}(M) \in \alpha_A(w)} \ \mathrm{sec}_2$$

[2] There are additional theoretical issues concerning awareness that we will consider in more detail in future work. For instance, closure rules such as $\varphi \in \alpha_A(w) \Rightarrow \neg\varphi \in \alpha_A(w)$ are proposed in [8]. While such closure conditions may be suitable in other applications of awareness logics, they seem to be undesirable for security protocols analysis, as, for example, awareness of a secret should not imply awareness of its negation.

The *identification rules* model how an agent A may use secrets to become aware of who uttered a message (and the sub-messages of it that A can obtain):

$$\frac{\mathsf{sees}_A(M') \in \alpha_A(w) \quad \mathsf{sec}_{\{A,B\}}(M) \in \alpha_A(w) \quad M \in \mathsf{submsg}_A(w, M')}{\mathsf{says}_B(A, M') \in \alpha_A(w)}\ \mathrm{ident}_1$$

$$\frac{\mathsf{says}_B(A, M') \in \alpha_A(w) \quad M \in \mathsf{submsg}_A(w, M')}{\mathsf{says}_B(A, M) \in \alpha_A(w)}\ \mathrm{ident}_2$$

■

Some remarks on the intuitions behind these rules. In the case of the NSPK protocol we only need to consider groups with two members and reason about secrecy of nonces (since the nonces used to authenticate the agents should be secret). In general, we could consider groups of arbitrary size and intended secrets other than nonces. To this end, we could replace in the rule sec_1 the premise $M \in \mathcal{T}_N$, stating that M is a nonce, with a premise $M \in \mathfrak{S}$, stating that M belongs to some "secrecy type" consisting of the message items that should be secret for a particular protocol, such as private or shared keys. As it is, the rule sec_1 states that if agent A is aware that he sent a nonce to agent B, then he is aware that it is a secret between B and himself. But he is only aware that he explicitly believes in the nonce's secrecy when he receives it back from B (as stated by the rule sec_2). In contrast to the rule basic_1, which states that each agent is aware of what he said, the identification rules model how nonces are used to perform authentication (via challenge and response): in the rule ident_1, if an agent receives a message M' including as a readable sub-message a secret M shared between B and himself, then A is aware that the message came from B. Note that, in case of an attack, A might conclude the wrong identity of the sender. (Indeed, recall that awareness of a formula does not imply its truth.)

In the next section we show that these rules suffice for reasoning about the man-in-the-middle attack upon the NSPK protocol. We expect that they will also apply to other public key protocols and that only slight changes and additions to the secrecy and identification rules will be needed for shared key protocols; the basic rules given above will apply to all protocols as they mirror semantic definitions. We have already begun experimenting in this direction. Modeling the Otway-Rees protocol [6], for example, requires groups of agents with more than two members and different strategies for reasoning about secrecy and identification: nonces and shared keys should be secrets that are shared between more than two agents to ensure authentication and (correct) key-distribution.

$$w_0 \xrightarrow{ev_1} w_1 \xrightarrow{ev_2} w_2 \xrightarrow{ev_3} w_3 \xrightarrow{ev_4} w_4 \xrightarrow{ev_5} w_5$$

Fig. 3. Sequence of events in the man-in-the-middle attack upon the NSPK protocol (ev_1–ev_5 as in Fig. 1)

4 Case study: the man-in-the-middle attack upon the NSPK protocol

We now give an application of our approach by showing how to use awareness and the agents' beliefs to characterize and reason about the man-in-the-middle attack upon the NSPK protocol. As shown in Fig. 1, the attack consists of a sequence of five events, and Fig. 3 names the global states generated by these events. To express the attack in terms of awareness, observe that in the single execution of the NSPK protocol in Fig. 1 the agents A and B mutually authenticate each other with the help of the secret nonces N_1 and N_2. According to Lowe [11], the attack is not upon A, as he voluntarily starts the protocol with Spy, but it is upon C, who is tricked into concluding that he is communicating with A instead of Spy. [3] We can represent this in terms of the secrecy of N_2 as follows: after the second event Spy $\rightarrow C : \{\!|A \circ N_1|\!\}_{K_C}$, C generates N_2 as a challenge-response secret nonce between himself and the agent whose identity is bound to the nonce N_1, i.e. A, and not between himself and the agent who actually initiated the protocol with him, i.e. Spy. As a result of his own action, after the third event $C \rightarrow A : \{\!|N_1 \circ N_2|\!\}_{K_A}$, C is aware of N_2 being a secret between A and himself, which amounts to $\mathcal{A}_C \mathsf{sec}_{\{A,C\}}(N_2)$ holding at w_3, w_4 and w_5. If C were however able to consider the attack sequence, at the end of it he would not implicitly believe in the secrecy of N_2, since his challenge has been responded to but the nonce is now possessed also by Spy.

Let now $\mathfrak{M}$ be a model generated by a subset $P \subseteq \tau$ of traces modeling the NSPK protocol. $\mathfrak{M}$ then contains the worlds w_0–w_5 of Fig. 3, and we can express the man-in-the-middle attack by proving that

$$\mathfrak{M} \not\models \mathcal{A}_C \mathsf{sec}_{\{A,C\}}(N_2) \rightarrow \mathcal{I}_C \mathsf{sec}_{\{A,C\}}(N_2)\,. \tag{1}$$

That is, there is at least one world in $\mathfrak{M}$ at which C is aware that N_2 is a secret he shares with A but he does not implicitly believe in this, which implies that $\mathfrak{M} \not\models \mathcal{B}_C \mathsf{sec}_{\{A,C\}}(N_2)$, i.e. that C does not explicitly believe in the secrecy of his own secret nonce.

Proof of (1)**:** We begin by computing the global state w_5, where, for sim-

[3] Note that A does not know that he is executing the protocol with an attacker; indeed, Spy could just be a previously honest agent now turned bad or could represent an honest agent whose private key has been compromised. Moreover, note that one could argue that also A is attacked, as the Spy uses his nonce N_1 to masquerade as him to C. That A wrongfully assumes N_1 to be a secret between himself and the Spy can also be shown using our approach.

plicity, we only show the primitive terms that the agents possess:

$$s_A(w_5) = \begin{cases} Po_A(w_5) = \{A, K_A, K_A^{-1}, \mathrm{Spy}, K_{\mathrm{Spy}}, N_1, N_2\} \\ Ac_A(w_5) = \{\mathsf{Send}_A(\mathrm{Spy}, \{\!|A \circ N_1|\!\}_{K_{\mathrm{Spy}}}), \mathsf{Rec}_A(\{\!|N_1 \circ N_2|\!\}_{K_A}), \\ \qquad \mathsf{Send}_A(\mathrm{Spy}, \{\!|N_2|\!\}_{K_{\mathrm{Spy}}})\} \end{cases}$$

$$s_C(w_5) = \begin{cases} Po_C(w_5) = \{C, K_C, K_C^{-1}, A, K_A, N_1, N_2\} \\ Ac_C(w_5) = \{\mathsf{Rec}_C(\{\!|A \circ N_1|\!\}_{K_C}), \mathsf{Send}_C(A, \{\!|N_1 \circ N_2|\!\}_{K_A}), \\ \qquad \mathsf{Rec}_C(\{\!|N_2|\!\}_{K_C})\} \end{cases}$$

$$s_{\mathrm{Spy}}(w_5) = \begin{cases} Po_{\mathrm{Spy}}(w_5) = \{\mathrm{Spy}, K_{\mathrm{Spy}}, K_{\mathrm{Spy}}^{-1}, A, K_A, C, K_C, N_1, N_2\} \\ Ac_{\mathrm{Spy}}(w_5) = \{\mathsf{Rec}_{\mathrm{Spy}}(\{\!|A \circ N_1|\!\}_{K_{\mathrm{Spy}}}), \mathsf{Send}_{\mathrm{Spy}}(C, \{\!|A \circ N_1|\!\}_{K_C}), \\ \qquad \mathsf{Rec}_{\mathrm{Spy}}(\{\!|N_2|\!\}_{K_{\mathrm{Spy}}}), \mathsf{Send}_{\mathrm{Spy}}(C, \{\!|N_2|\!\}_{K_C})\} \\ \qquad \cup\ Ac_A(w_5) \cup\ Ac_C(w_5) \end{cases}$$

where $initState(A) = \{A, K_A, K_A^{-1}, \mathrm{Spy}, K_{\mathrm{Spy}}\}$, $initState(C) = \{C, K_C, K_C^{-1}\}$ and $initState(\mathrm{Spy}) = \{\mathrm{Spy}, K_{\mathrm{Spy}}, K_{\mathrm{Spy}}^{-1}, A, K_A, C, K_C\}$.

We now show that awareness and belief allow us to characterize the attack at the end of the event sequence. That is, we show that $\mathfrak{M}, w_5 \models \mathcal{A}_C \mathsf{sec}_{\{A,C\}}(N_2)$ and $\mathfrak{M}, w_5 \not\models \mathcal{I}_C \mathsf{sec}_{\{A,C\}}(N_2)$, which implies (1). By definition, $\mathfrak{M}, w_5 \models \mathcal{A}_C \mathsf{sec}_{\{A,C\}}(N_2)$ iff $\mathsf{sec}_{\{A,C\}}(N_2) \in \alpha_C(w_5)$, which holds since

$$\frac{\mathsf{Send}_C(A, \{\!|N_1 \circ N_2|\!\}_{K_A}) \in Ac_C(w_5) \quad N_2 \in \mathsf{comp}_C(w_5, \{\!|N_1 \circ N_2|\!\}_{K_A})}{\mathsf{says}_C(A, N_2) \in \alpha_C(w_5)}\ \mathrm{basic}_1$$

and

$$\frac{\mathsf{says}_C(A, N_2) \in \alpha_C(w_5) \quad N_2 \in \mathcal{T}_N}{\mathsf{sec}_{\{A,C\}}(N_2) \in \alpha_C(w_5)}\ \mathrm{sec}_1$$

where the only sub-goal that does not follow directly by assumption or from the local state of the agent C is $N_2 \in \mathsf{comp}_C(w_5, \{\!|N_1 \circ N_2|\!\}_{K_A})$, which by definition of comp holds iff (i) $N_2 \in \mathsf{analz}(Po_C(w_5))$, (ii) $\{\!|N_1 \circ N_2|\!\}_{K_A} \in \mathsf{synth}(Po_C(w_5))$, and (iii) $N_2 \in \mathsf{parts}(\{\{\!|N_1 \circ N_2|\!\}_{K_A}\})$. Sub-goals (i) and (iii) follow from

$$\frac{N_2 \in Po_C(w_5)}{N_2 \in \mathsf{analz}(Po_C(w_5))}\ \mathsf{analz\text{-}inj}$$

and

$$\dfrac{\dfrac{\dfrac{\{\!|N_1 \circ N_2|\!\}_{K_A} \in \{\{\!|N_1 \circ N_2|\!\}_{K_A}\}}{\{\!|N_1 \circ N_2|\!\}_{K_A} \in \mathsf{parts}(\{\{\!|N_1 \circ N_2|\!\}_{K_A}\})}\,\mathsf{parts\text{-}inj}}{N_1 \circ N_2 \in \mathsf{parts}(\{\{\!|N_1 \circ N_2|\!\}_{K_A}\})}\,\mathsf{parts\text{-}body}}{N_2 \in \mathsf{parts}(\{\{\!|N_1 \circ N_2|\!\}_{K_A}\})}\,\mathsf{parts\text{-}2}$$

and (ii) follows from

$$\dfrac{\dfrac{N_1 \in Po_C(w_5)}{N_1 \in \mathsf{synth}(Po_C(w_5))}\,\mathsf{synth\text{-}inj} \quad \dfrac{N_2 \in Po_C(w_5)}{N_2 \in \mathsf{synth}(Po_C(w_5))}\,\mathsf{synth\text{-}inj}}{N_1 \circ N_2 \in \mathsf{synth}(Po_C(w_5))}\,\mathsf{synth\text{-}pair}$$

and

$$\dfrac{N_1 \circ N_2 \in \mathsf{synth}(Po_C(w_5)) \quad \dfrac{K_A \in Po_C(w_5)}{K_A \in \mathsf{synth}(Po_C(w_5))}\,\mathsf{synth\text{-}inj}}{\{\!|N_1 \circ N_2|\!\}_{K_A} \in \mathsf{synth}(Po_C(w_5))}\,\mathsf{synth\text{-}enc}$$

Hence, $\mathfrak{M}, w_5 \models \mathcal{A}_C\mathsf{sec}_{\{A,C\}}(N_2)$, i.e. at w_5 agent C is aware that N_2 is a secret between A and himself. To show that he does not implicitly believe in it, i.e. that $\mathfrak{M}, w_5 \not\models \mathcal{I}_C\mathsf{sec}_{\{A,C\}}(N_2)$, observe that by definition $\mathfrak{M}, w_5 \models \mathcal{I}_C\mathsf{sec}_{\{A,C\}}(N_2)$ iff $\mathfrak{M}, w' \models \mathsf{sec}_{\{A,C\}}(N_2)$ for all w' such that $w_5 \sim_C w'$. Since w' can only be w_5 by the definition of $\sim_C$, we check whether $\mathfrak{M}, w_5 \models \mathsf{sec}_{\{A,C\}}(N_2)$, which holds iff $\mathfrak{M}, w_5 \models \mathsf{has}_i(N_2)$ for all agents $i \in \{A, C\}$, and $\mathfrak{M}, w_5 \not\models \mathsf{has}_j(N_2)$ for all agents $j \notin \{A, C\}$. Since j can only be Spy and $\mathfrak{M}, w_5 \models \mathsf{has}_{\mathsf{Spy}}(N_2)$, it follows that $\mathfrak{M}, w_5 \not\models \mathsf{sec}_{\{A,C\}}(N_2)$, and we conclude the proof. ■

In [11], Lowe also shows how to fix the NSPK protocol to avoid the man-in-the-middle attack. The solution is simply to include the identity of the responder B in the second step of the protocol, i.e. to replace ns_2 by $\mathsf{ns}_2' = B \to A : \{\!|B \circ N_1 \circ N_2|\!\}_{K_A}$ so that an attempted attack stops after ev_3 since A will not reply to B's message as B is not the agent with whom A has started the protocol with the nonce N_1. It is straightforward to show that in the corrected version of the protocol we have $\mathfrak{M} \models \mathcal{B}_B\mathsf{sec}_{\{A,B\}}(N_2)$, i.e. the protocol is correct with respect to the explicit beliefs of a responder B, which rules out this attack.

We can straightforwardly generalize the insights obtained from this case study as follows. Let $\mathfrak{M}$ be a model generated by a subset $P \subseteq \tau$ of traces modeling an authentication protocol, where authentication is realized in terms of secret nonces. The protocol can be attacked, meaning that the secrecy of some nonce N can be violated whenever for some agent A there is an agent

B such that

$$\mathfrak{M} \not\models \mathcal{A}_A \mathsf{sec}_{\{A,B\}}(N) \to \mathcal{I}_A \mathsf{sec}_{\{A,B\}}(N)\,.$$

This formula represents (an agent's view of) an attack upon the secrecy of the nonces used for authentication, as in the NSPK and in a number of other public and shared key protocols. As we mentioned in the previous section, we can generalize our awareness rules by introducing a secrecy type to capture secrets other than nonces, but this is but a first step towards generality. Authentication can be defined (and realized) in different ways [12] and there are several other security properties that are implemented by security protocols, such as confidentiality and integrity of data. We are currently working on extending our approach to analyze these properties and, more generally, on characterizing the class of attacks upon these properties that can be expressed in terms of awareness.

5 Related work

In the introduction we surveyed some of the related work to place our work in context. We now expand on this and compare our work in more detail with other approaches to giving semantics for authentication logics. To this end, let us first briefly review these works.

The GNY logic of Gong, Needham and Yahalom [10] was the first extension of the BAN logic. Its semantics is close to that of BAN, i.e. an operational semantics where the local state of an agent consists of a set possessions and a set of formulas he believes in.

Abadi and Tuttle [1] define a possible-worlds semantics for the logic AT, which is an extension of BAN that considers multiple executions of a protocol. Although the logic lacks an explicit notion of awareness, the *hide* operator provides a basis for a weak form of "belief as a form of resource-bounded, defeasible knowledge" [1, p. 202], and thereby captures some of the notions that our use of awareness makes explicit. Technically, the *hide* operator is introduced to model the assumption of perfect cryptography; the idea is to conceal the contents of unreadable encrypted messages that an agent possesses, thus preventing him from believing that a message contains information he does not (or is not yet ready to) understand.

In [17], Syverson and van Oorschot give the logic SVO, which combines features of the GNY and AT logics with additional notions such as key agreement. The SVO logic has a possible-worlds semantics that is close to the semantics given in [1], but *hiding* is replaced by a more refined notion of *comprehension*, which is closely related to our awareness: each agent is equipped with a set of *comprehended messages*, which can be intuitively seen as the set of messages that the agent "can ultimately tie back to cleartext he has seen" [17, p. 20].

Kessler and Wedel [19] extend SVO to the logic AUTLOG in which resource-bounded reasoning is captured by means of the *sight* operator, which performs a task similar to Syverson and van Oorschot's comprehension.

To summarize, the above illustrates how notions close to resource-bounding and awareness have been implicitly modeled in a number of previous approaches to giving a semantics for BAN or its descendants. The major novelty of our approach is that awareness is the heart of the matter, i.e. we take awareness as the explicit, central concept, and use it to model and reason about the agents' resource-bounded reasoning and attacks in possibly interleaved protocol executions. As a further comparison, note that the logics BAN, GNY, AT, SVO and AUTLOG all share the view that a system's engineer would have of the system, meaning that belief is ascribed to the agents by an external observer, instead of being developed internally by the agents themselves, which is what we capture in our approach.

6 Conclusions and further work

We have reported on work-in-progress on an awareness-based semantics for the analysis of security protocols that combines complementary features of security logics and inductive methods, and as an example we have showed how to model the man-in-the-middle attack upon the NSPK protocol. Aside from the characterization questions mentioned at the end of §4, current work includes: (i) the formalization of a deduction system for our approach, possibly a Gentzen-style or tableaux-style one to ease its usability and mechanization, (ii) the investigation of the metatheoretic properties of the deduction system, such as its soundness and completeness with respect to the semantics we have given here, and its computational complexity, and (iii) the mechanization of our logic using the generic theorem prover Isabelle [13], which will allow us not only to machine-check our proofs but also to investigate possible combinations of our approach with Paulson's inductive method, which employs the Isabelle encoding of higher-order logic.

As we remarked above, we are also currently working on applying our method to analyzing other security protocols, and thereby investigate the expressiveness of our semantics and lay the foundations for a more detailed comparison with other related approaches. The analysis of other protocols, such as the Otway-Rees protocol, will require us to consider groups of agents with more than two members, and here we also intend to explore the connections with standard notions of groups of agents [9,16]. It will also require us to devise additional awareness rules and possibly modify the secrecy and identification rules we have given here (e.g. to consider the agents' nested and mutual, or second-level, beliefs). Our research will in particular be aimed at developing more principled ways of devising these rules.

References

[1] Abadi, M. and M. R. Tuttle, *A semantics for a logic of authentication*, in: *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing*, ACM Press, 1991 pp. 201–216.

[2] Anderson, R. J. and R. Needham, *Programming Satan's computer*, in: J. van Leeuwen, editor, *Computer Science Today: Recent Trends and Developments*, LNCS 1000, Springer-Verlag, 1995 pp. 426–440.

[3] Basin, D., *Lazy infinite-state analysis of security protocols*, in: R. Baumgart, editor, *Proceedings of CQRE'99*, LNCS 1740, Springer-Verlag, Berlin, 1999 pp. 30–42.

[4] Bleeker, A. and L. Meertens, *A semantics for BAN logic*, in: *Proceeding of DIMACS Workshop on Design and Formal Verification of Crypto Protocols*, 1997 .

[5] Burrows, M., M. Abadi and R. Needham, *A logic of authentication*, ACM Transactions on Computer Systems **8** (1990), pp. 18–36.

[6] Clark, J. and J. Jacob, *A survey of authentication protocol literature: Version 1.0* (1997), manuscript available at the URL `http://www.cs.york.ac.uk/~jac/papers/drareview.ps.gz`.

[7] Dolev, D. and A. Yao, *On the security of public key protocols*, IEEE Transactions on Information Theory **2** (1983), pp. 198–208.

[8] Fagin, R. and J. Y. Halpern, *Belief, awareness and limited reasoning*, Artificial Intelligence **34** (1987), pp. 39–76.

[9] Fagin, R., J. Y. Halpern, Y. Moses and M. Y. Vardi, "Reasoning about knowledge," MIT Press, 1995.

[10] Gong, L., R. Needham and R. Yahalom, *Reasoning About Belief in Cryptographic Protocols*, in: D. Cooper and T. Lunt, editors, *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, IEEE Computer Society, 1990 pp. 234–248.

[11] Lowe, G., *Breaking and fixing the Needham-Schroeder public-key protocol using FDR*, in: T. Margaria and B. Steffen, editors, *Proceedings of TACAS'96*, LNCS 1055, Springer-Verlag, Berlin, 1996 pp. 147–166.

[12] Lowe, G., *A hierarchy of authentication specifications*, in: *Proceedings of The 10th Computer Security Foundations Workshop*, IEEE Computer Society Press, 1997 .

[13] Paulson, L. C., "Isabelle: a generic theorem prover," LNCS 828, Springer-Verlag, Berlin, 1994.

[14] Paulson, L. C., *The inductive approach to verifying cryptographic protocols*, Journal of Computer Security **6** (1998), pp. 85–128.

[15] Syverson, P. F., *Knowledge, belief, and semantics in the analysis of cryptographic protocols*, Journal of Computer Security **1** (1992), pp. 317–334.

[16] Syverson, P. F. and S. G. Stubblebine, *Group principals and the formalization of anonymity*, in: J. Wing, J. Woodcock and J. Davies, editors, *Proceedings of FM'99*, LNCS 1708, Springer-Verlag, Berlin, 1999 pp. 814–833.

[17] Syverson, P. F. and P. C. van Oorschot, *On unifying some cryptographic protocol logics*, in: *Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, IEEE Computer Society Press, 1994 pp. 14–28.

[18] Thijsse, E., *On total awareness logics*, in: M. de Rijke, editor, *Defaults and Diamonds*, Kluwer Academic Publishers, Dordrecht, 1993 pp. 309–347.

[19] Wedel, G. and V. Kessler, *Formal semantics for authentication logics*, in: E. Bertino, H. Kurth, G. Martella and E. Montolivo, editors, *Proceedings of ESORICS'96*, LNCS 1146, Springer-Verlag, Berlin, 1996 pp. 219–241.