Revisiting the Foundations of Authentication Logics

Joseph Y. Halpern
Department of Computer Science
Cornell University

Riccardo Pucella
Department of Computer Science
Cornell University

Ron van der Meyden
School of Computer Science and Engineering
University of New South Wales

Abstract

In this paper, we make the point that the problems with logics in the BAN tradition are not with the idea of basing reasoning about security protocols using epistemic notions, but with some of the specific decisions taken in the formulation of these logics. To illustrate this statement, we describe a formal logic for security protocol analysis based on well-understood modal operators, knowledge, time, and probability. We show how the logic can capture the intuitive high-level concepts of BAN and later logics. In particular, we formalize a translation of the BAN operators into our logic to model reasoning about security protocols in the presence of a Dolev-Yao adversary. We validate our translation by analyzing the Needham-Schroeder authentication protocol using our formalism. This analysis highlights some strong assumptions on nonces made by the Dolev-Yao model of the adversary. We address these concerns by formulating a different translation of the BAN operators using probability, and show how to analyze protocols in the presence of Dolev-Yao adversaries that are allowed guesses.

1 Introduction

For the past fifteen years, there has been an intuition in the world of security that formal theories of knowledge and belief should have something interesting to say about security protocols, and cryptographic protocols in particular. The idea is simply to define a logic to put protocol analysis on a firm formal basis, giving protocol designers a way to validate their designs. Unfortunately, this intuition has not come to fruition, despite many attempts.

Setting the tone for this line of research, the BAN logic (Burrows, Abadi, and Needham 1990a) was developed in the late 1980s to address the problem of analyzing security protocols. The BAN logic focused on the notion of *trust*, which is useful for analyzing authentication protocols. BAN has been the subject of many criticisms (Nessett 1990; Boyd and Mao 1993), some more fair than others, and defenses (Burrows, Abadi, and Needham 1990b; van Oorschot 1993a). In no small part, the crux of the criticism rested on the fact that BAN has a multitude of operators, but no semantics to speak of.[1] The main problem with not having an independently motivated semantics is that it is not clear exactly what one is proving when a "proof" of security is exhibited for a protocol. In general, when a BAN style analysis manages to exhibit a bug in a protocol, chances are good there is indeed a bug, but a proof of security does not guarantee much. Compounding the problem are the various generalizations of BAN, further complicating the logic (Gong, Needham, and Yahalom 1990; Abadi and Tuttle 1991; van Oorschot 1993b; Stubblebine and Wright 1996; Syverson and van Oorschot 1994). Starting with the work of Abadi and Tuttle (1991) (AT from now on)

[1] Burrows, Abadi, and Needham (1990a) did sketch a semantics, but it merely encodes the rules of the logic.

attempts have been made to supply a more adequate semantics for BAN style logics (Syverson and van Oorschot 1994; Stubblebine and Wright 1996; Wedel and Kessler 1996), but, as we argue below, they do not seem to completely capture the intuitions behind all the BAN constructs.

Other approaches to the formal analysis of security protocols have emerged in recent years that seem to provide stronger correctness guarantees than BAN logic. These approaches work by reasoning directly about the behavior of adversaries, something that is not captured by the BAN primitives (although the semantic models for BAN since AT take account of it). Examples include model checking (Lowe 1998; Mitchell, Mitchell, and Stern 1997), inductive-assertions methods (Paulson 1998) and *strand spaces* (Thayer, Herzog, and Guttman 1999). These more recent approaches do not make explicit use of logics for knowledge and belief, and yet have been quite successful. One might therefore ask whether the intuitions about the utility of using knowledge and belief to reason about security protocols were misplaced.

We argue in this paper that the answer to this question is "No". The problems with logics in the BAN tradition are not with the idea of basing reasoning about security protocols using epistemic notions, but with some of the specific decisions taken in the formulation of these logics. BAN was originally formulated for dealing with authentication protocols, and contains some primitives specific to that domain. There has been a debate in the literature about the appropriate set of primitives such a logic should be. Our attitude to this is that, where possible, one should minimize the number of primitives, and use constructs that are well understood and have utility in a broader domain. Taking the original BAN logic as a case study, we argue that the intuitive meaning of its primitives can be expressed using notions from the logic of knowledge and time.[2] These notions are well understood, both semantically and axiomatically, and are relevant not just to authentication protocols but to distributed protocols (Dwork and Moses 1990; Moses and Tuttle 1988), reactive systems (Pnueli 1977) and game theory (Aumann 1999; Brandenburger 1989). Our representation has the form of a translation from BAN logic to the logic of knowledge and time, extended by appropriately chosen primitive propositions.

We validate our claim to have captured the intention of BAN logic in two ways. First, we show that our translation validates the BAN axioms with respect to a semantics that models the Dolev-Yao adversary. In order to achieve this result, we need to make some strong assumptions about properties of the models we consider. These assumptions are not artefacts of our translation, but assumptions built into BAN. Second, we consider a simple and well-known example that has been a popular case study for formal methods for security: the Needham-Schroeder public key authentication protocol. We show that this protocol generates a model in our semantics that satisfies the specifications for the protocol given in (Burrows, Abadi, and Needham 1990a). The model is very similar to that used in more recent work that is not in the BAN tradition. In fact, our proofs are very similar to those developed for this protocol by Paulson (1998) using an inductive-assertions method. Thus, although the more recent approaches have eschewed epistemic operators, they can be applied to derive epistemic goals of the type sought in BAN analyses. Significantly, our reasoning makes no use of the idealization step that has frequently been argued to be one of the weakest points of BAN. This suggests that the problem with the BAN approach is not with the types of conclusions that (Burrows, Abadi, and Needham 1990a) set out to prove.

We next take a closer look at what has been achieved in our analysis of the Needham-Schroeder protocol with respect to the Dolev-Yao model. We argue that although we have an apparent justification of the protocol, the semantic model misses some of the key intuitions underlying the protocol—in particular, the role of nonces. The difficulty applies to both our BAN logic style conclusions and to the conclusions established for this protocol by Paulson, and raises concerns about the meaning of what has been proved. (In the case of Paulson's results, our observations reinforce our claim that epistemic notions are central to the analysis of cryptographic protocols.)

In order to address these concerns, we formulate a second semantic model, in which we enable the

[2] We believe that we could obtain similar results for other logics in the family of BAN logics.

adversary to guess nonces. We again set out to show that the intentions of BAN logic can be justified with respect to this model. Doing so requires a different translation of the logic: we now need to introduce probability as a basic construct. (Like knowledge and time, this is a well-understood notion (Halpern and Tuttle 1993).) We again justify this translation in two ways: by showing that the axioms are satisfied, and that the conclusions for the Needham-Schroeder protocol claimed by (Burrows, Abadi, and Needham 1990a) in fact hold in the model of the protocol generated with respect to this adversary. This second set of results provides what we believe to be a much stronger justification for the use of epistemic notions in security protocol analysis.

One of the outcomes of our analysis is to identify a subtlety in the interpretation of belief in cryptographic protocols: although BAN is generally conceived as reasoning about the final beliefs of the principals, we show that in order to obtain the desired conclusions, we need to interpret the beliefs with respect to the initial, not final distributions. In other words, the beliefs are beliefs that principals hold before the protocol executes about the state of the world at the end of a protocol execution, rather than beliefs that agents hold at the end of a protocol execution. This is an extremely important distinction for the purposes of security, since the latter kind of belief offers much stronger guarantees than the former. Indeed, it is easy to come up with situation where a principal has strong initial beliefs about the outcome of a protocol execution (for instance, that some negative event will not happen) only to find himself knowing without any uncertainty after the protocol executes that the bad event must have happened. In contrast, a strong final belief would indicate that after the protocol executes, the principals finds it unlikely that the bad event happened. It is interesting that BAN style analyses lead to the weaker form of guarantee.

The structure of the paper reflects the above discussion. In Section 2, we review the multiagent systems approach of (Fagin, Halpern, Moses, and Vardi 1995), on which our semantics is based, and introduce our logic. In Section 3, we review BAN and present a translation that interprets BAN belief as knowledge in our logic. In Section 4, we validate this translation by analyzing the Needham-Schroeder protocol using BAN formulas translated into our logic. We prove formally that the NS protocol is correct with respect to Dolev-Yao style adversaries that cannot make guesses, using our knowledge-based translation. In Section 5, we discuss the role of nonces, in both our analysis and Paulson's inductive-assertions method. In Section 6, we extend our logic with probabilities, and give a second translation that interprets BAN belief as knowing with high probability. We then prove that the NS protocol is correct (under the probability-based interpretation) in the presence of an adversary that can make guesses.

2 A logic for reasoning about security protocols

Our approach consists of defining a logic in which we can specify properties of systems of interacting agents. We can formally define what it means for such a property to be true in a particular system. By choosing systems that correspond to the possible executions of a protocol (run by many agents), we can define what it means for a system based on the protocol to have that property.

2.1 Multiagent systems

We briefly review the multiagent system framework from (Fagin, Halpern, Moses, and Vardi 1995), since this is the framework we use to represent systems and give semantics to the logic.

A multiagent system consists of n agents and an environment, each of which is in some local state at a given point in time. We assume that an agent's local state encapsulates all the information to which the agent has access. In the security setting, the local state of an agent might include some initial information regarding keys, the messages she has sent and received, and perhaps the reading of a clock. The environment state describes information relevant to the analysis which may not be in any agent's state. We can then view the whole system as being in some global state, which is a tuple consisting of each process' local state. A global state has the form $(s_e, s_1, \ldots, s_n)$, where s_i is agent i's state, for $i = 1, \ldots, n$, and s_e is the environment state. In general, the actual form of these local states depends on the application.

A system is not a static entity. To capture its dynamic aspects, we define a run to be a function from time to global states. Intuitively, a run is a complete description of what happens over time in one possible execution of the system. A *point* is a pair (r, m) consisting of a run r and a time m. For simplicity, we take time to range over the natural numbers in the remainder of this discussion. At a point (r, m), the system is in some global state $r(m)$. If $r(m) = (s_e, s_1, \ldots, s_n)$, then we take $r_i(m)$ to be s_i, agent i's local state at the point (r, m). We formally define a *system* to consist of a set $\mathcal{R}$ of runs (or executions).

For simplicity, we restrict attention to a specific class of systems, suited to modeling security protocols. These are message-passing systems in which one (or more) of the agents is an adversary with the capacity to monitor and control message transmission. We assume that the local state of an agent is a sequence of events of the form $\langle e_0, e_1, \ldots, e_m \rangle$, where e_0 is the initial state, and e_i for $i \geq 1$ is a term of the form $\mathsf{send}(i, \mathsf{m})$ or $\mathsf{recv}(\mathsf{m})$, where m is a message and i is an agent. (We discuss the structure of messages below.) The initial state represents information such as the public and private keys to be used during the run, and other values such as nonces to be used by the agent during the run. To make this precise, we assume that the initial state is a set of *bindings* of identifiers to key or nonce values. We will use $\mathtt{id} \mapsto v$ to represent the binding of value v to the identifier $\mathtt{id}$. Identifiers are useful to describe protocols, as we will see in Section 4 when we study the Needham-Schroeder protocol in this framework. The term $\mathsf{send}(i, \mathsf{m})$ represent that the agent sent message m, intending that it be delivered to agent i. The term $\mathsf{recv}(\mathsf{m})$ represents the reception of message m. Note that the sender is not included in $\mathsf{recv}(\mathsf{m})$. Intuitively, this is because the receiver may not be able to determine the sender of a message it receives.[3]

Ultimately, we want to analyze security protocols, not multiagent systems. Intuitively, a protocol for agent i is a description of what actions i may take as a function of her local state. Actions, for our purposes, will consists of sending messages (although we could incorporate other actions, such as choosing keys, or tossing coins to randomize protocols). Typically, each honest principal runs its prescribed protocols, while adversaries run one of a set of possible adversary protocols. The adversary protocols may involve guessing keys and nonces, or attempts to launch attacks by forwarding messages. A *joint protocol* $P = (P_e, P_1, \ldots, P_n)$ consists of a protocol for each agent, and essentially describes how global states evolve. We say a run r is *consistent with joint protocol* P if it could have been generated by P. Given a set I of global states, and a set $\mathcal{P}$ of joint protocols, the system $\mathcal{R}(\mathcal{P}, I)$ consists of all the runs consistent with some joint protocol in $\mathcal{P}$ that start at some initial global state in I. We give a precise definition of $\mathcal{R}(\mathcal{P}, I)$ in Appendix B.

2.2 Syntax of the logic

We introduce a simple multimodal propositional logic for reasoning about security protocols. We assume a finite set of principals that for simplicity we will represent by integers, $1, \ldots, N$, a set $\mathcal{K}$ of keys, a set $\mathcal{N}$ of nonces, a set $\mathcal{T}$ of plaintexts, and a set $\Phi_0 = \{p_1, p_2, \ldots\}$ of (application-specific) primitive propositions. We assume $\mathcal{K}$ contains both symmetric keys (used in shared key cryptography) and asymmetric keys (used in public-key cryptography), and that they can be distinguished. We also assume that keys, nonces, and plaintexts can be distinguished, so that $\mathcal{K}$, $\mathcal{N}$, and $\mathcal{T}$ are disjoint sets, and that encrypted messages can be distinguished from unencrypted messages. Like AT and SW, we assume that formulas are actually a subset of messages, and can include messages in them. Thus, we define formulas and message simultaneously as follows, where we use p for a generic element of Φ_0, i for a generic principal, m for a generic message, t

[3]To reason about protocols in richer settings, such as ones allowing adversaries to derive information from traffic analysis, we can add more information to the local states, such as sender and intended destination of messages, time stamps, etc.

for a generic plaintext, k for a generic key, n for a generic nonce, and φ for a generic formula:

$$\mathsf{m} \quad ::= \quad t \mid k \mid n \mid \mathsf{m}\cdot\mathsf{m} \mid \{\mathsf{m}\}_k \mid \varphi \mid i$$

$$\varphi \quad ::= \quad p \mid \mathsf{sent}_i(\mathsf{m}) \mid \mathsf{recv}_i(\mathsf{m}) \mid \mathsf{extract}_i(\mathsf{m}) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid K_i\varphi \mid \bigcirc\varphi \mid \ominus\varphi \mid \Box\varphi \mid \boxminus\varphi.$$

We write $\mathsf{m}\cdot\mathsf{m}'$ for pairing, and $\{\mathsf{m}\}_k$ for the result of encrypting message m with key k. Besides the application-specific primitive propositions, we also have primitive propositions of the form $\mathsf{sent}_i(\mathsf{m})$, $\mathsf{recv}_i(\mathsf{m})$, and $\mathsf{extract}_i(\mathsf{m})$. As we might expect, $\mathsf{sent}_i(\mathsf{m})$ if i sent m, and similarly for $\mathsf{recv}_i(\mathsf{m})$. The formula $\mathsf{extract}_i(\mathsf{m})$ if i can "extract" the message m from messages it has received (and other information at its disposal). What extract means depends on the application, the capabilities of principals, and the protocol they are running. We view extraction as a black box. In Section 2.4, we will see a concrete implementation of extraction capturing the Dolev-Yao capabilities. We emphasize that our primitive propositions $\mathsf{sent}_i(\mathsf{m})$, $\mathsf{recv}_i(\mathsf{m})$ and $\mathsf{extract}_i(\mathsf{m})$ are *propositions*, not predicates. In other words, the message m that they carry is not a value, but rather an intrinsic part of the proposition.

The knowledge operator $K_i\varphi$ states that agent i knows the fact φ. The temporal operator $\bigcirc\varphi$ states that φ is true at the next time step, while $\ominus\varphi$ states that φ was true at the previous time step. We will use the abbreviations $\bigcirc^t\varphi$ and $\ominus^t\varphi$ (for $t \in \mathbb{N}$) for the t-fold application of $\bigcirc$ and $\ominus$, respectively, to φ. The temporal operator $\Box\varphi$ states that φ is true at the current time, and all subsequent times. Similarly, $\boxminus\varphi$ states that φ is true at the current time, and all previous times. We define the usual abbreviations, namely $\varphi_1 \vee \varphi_2$ for $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \Rightarrow \varphi_2$ for $\neg\varphi_1 \vee \varphi_2$, and so on.

We define a *submessage* relation $\sqsubseteq$ on messages, as the smallest reflexive and transitive relation satisfying $\mathsf{m} \sqsubseteq \mathsf{m}\cdot\mathsf{m}'$, $\mathsf{m} \sqsubseteq \mathsf{m}'\cdot\mathsf{m}$, and $\mathsf{m} \sqsubseteq \{\mathsf{m}\}_k$.

2.3 Semantics of the logic

Semantics for modal logics are traditionally given in terms of Kripke structures (Kripke 1963). A system can be viewed as a Kripke structure if we specify a function π telling us how to assign truth values to the primitive propositions. An interpreted system $\mathcal{I}$ consists of a pair $(\mathcal{R}, \pi)$, where $\mathcal{R}$ is a system and π is an interpretation for the propositions in Φ_0 that assigns truth values to the primitive propositions at the global states. Thus, for every $p \in \Phi_0$ and global state s that arises in $\mathcal{R}$, we have $(\pi(s))(p) \in \{\mathbf{true}, \mathbf{false}\}$. Of course, π also induces an interpretation over the points of $\mathcal{R}$; simply take $\pi(r,m)$ to be $\pi(r(m))$. We refer to the points of the system $\mathcal{R}$ as points of the interpreted system $\mathcal{I}$. That is, we say that the point (r,m) is in the interpreted system $\mathcal{I} = (\mathcal{R}, \pi)$ if $r \in \mathcal{R}$. We assume a fixed interpretation for the primitive propositions $\mathsf{recv}_i(\mathsf{m})$ and $\mathsf{sent}_i(\mathsf{m})$, meaning that π must give the following meaning to those primitive propositions:

$$\pi(r,m)(\mathsf{recv}_i(\mathsf{m})) = \mathbf{true} \text{ iff } \mathsf{recv}(\mathsf{m}) \in r_i(m),$$

$$\pi(r,m)(\mathsf{sent}_i(\mathsf{m})) = \mathbf{true} \text{ iff } \exists \mathsf{m}' \text{ such that } \mathsf{m} \sqsubseteq \mathsf{m}' \text{ and } \mathsf{send}(j, \mathsf{m}') \in r_i(m) \text{ for some } j.$$

Given our interpretation of extraction as a black box, we put no constraints here on how π interprets $\mathsf{extract}_i(\mathsf{m})$ other than the fact that the interpretation depends only on i's local state. That is, if $r_i(m) = r'_i(m')$, we require that $\pi(r,m)(\mathsf{extract}_i(\mathsf{m})) = \pi(r',m')(\mathsf{extract}_i(\mathsf{m}))$.

For an interpreted system $\mathcal{I}$, we associate to every agent i an equivalence relation $\sim_i$ on points, such that $(r,m) \sim_i (r',m')$ if $r_i(m) = r'_i(m')$. In other words, $(r,m) \sim_i (r',m')$ if agent i has the same local state at both points, and are thus indistinguishable from her point of view.

The relation of satisfaction of a formula φ in an interpreted system $\mathcal{I} = (\mathcal{R}, \pi)$ at point (r,m), written $(\mathcal{I}, r, m) \models \varphi$, is defined inductively. The semantics of conjunction and negation is standard; the remaining clauses are give below:

$(\mathcal{I}, r, m) \models p$ iff $\pi(r, m)(p) =$ **true**

$(\mathcal{I}, r, m) \models K_i\varphi$ iff for all $(r', m') \sim_i (r, m)$, $(\mathcal{I}, r', m') \models \varphi$

$(\mathcal{I}, r, m) \models \bigcirc\varphi$ iff $(\mathcal{I}, r, m + 1) \models \varphi$

$(\mathcal{I}, r, m) \models \ominus\varphi$ iff $m \geq 1$ and $(\mathcal{I}, r, m - 1) \models \varphi$

$(\mathcal{I}, r, m) \models \Box\varphi$ iff for all $m' \geq m$, $(\mathcal{I}, r, m') \models \varphi$

$(\mathcal{I}, r, m) \models \boxminus\varphi$ iff for all $m' \leq m$, $(\mathcal{I}, r, m') \models \varphi$.

As usual, we write $\mathcal{I} \models \varphi$ if φ is *valid* in $\mathcal{I}$, that is, if $(\mathcal{I}, r, m) \models \varphi$ for all points (r, m) in $\mathcal{I}$.

These definitions of satisfaction of a formula are quite standard: in particular, they contain nothing that is specific to cryptographic protocols. Exactly the same definitions are used in the literature on reasoning about knowledge in non-cryptographic settings. The literature on semantics of BAN logic has also followed these definitions, but has used some specific assumptions about the nature of local states that are tailored to the cryptographic setting. We consider this issue in the following section.

2.4 Extraction

As we stated above, we do not impose any *a priori* restrictions on the interpretation of $\mathsf{extract}_i(\mathsf{m})$, except for it to depend only on the local state of agent i. Intuitively, the interpretation of $\mathsf{extract}$ is meant to capture the capabilities of the agents to take apart messages and construct new messages. While in principle a principal may be able to extract m_1 from $\mathsf{m}_1 \cdot \mathsf{m}_2$, it may not do so at a particular point in a system because the protocol that it is using does not break up $\mathsf{m}_1 \cdot \mathsf{m}_2$. Similarly, whether i can extract m from $\{\mathsf{m}\}_k$ depends in part on whether i "has" key k in some sense, i's protocol, and i's computational ability.

A typical set of capabilities found in the literature is due to Dolev and Yao (1983), and we now provide an interpretation of $\mathsf{extract}$ that captures them. However, we stress that this interpretation is not required. For instance, we can use an interpretation that captures the guess-and-validate adversary of Lowe (2002).

To make this precise, assume we have a function $\mathit{init}(s)$ that returns for a local state s the set of messages contained in its initial state (i.e., those keys and nonces bound to identifier in the initial state of s). If s is a local state for i, define $\mathit{can\_compute}(s)$ to be the smallest set of messages such that $\mathsf{m} \in \mathit{can\_compute}(s)$ if one of the following conditions hold:

- $\mathsf{recv}(\mathsf{m}) \in s$;
- $\mathsf{m} \in \mathit{init}(s)$;
- there exists m' with $\mathsf{m} \cdot \mathsf{m}' \in \mathit{can\_compute}(s)$ or $\mathsf{m}' \cdot \mathsf{m} \in \mathit{can\_compute}(s)$;
- there exists an asymmetric key k with $\{\mathsf{m}\}_k \in \mathit{can\_compute}(s)$ and $k^{-1} \in \mathit{can\_compute}(s)$;
- there exists a symmetric key k such that $\{\mathsf{m}\}_k \in \mathit{can\_compute}(s)$ and $k \in \mathit{can\_compute}(s)$;
- $\mathsf{m} = \mathsf{m}_1 \cdot \mathsf{m}_2$, with $\mathsf{m}_1 \in \mathit{can\_compute}(s)$ and $\mathsf{m}_2 \in \mathit{can\_compute}(s)$;
- $\mathsf{m} = \{\mathsf{m}_1\}_k$, with $\mathsf{m}_1 \in \mathit{can\_compute}(s)$ and $k \in \mathit{can\_compute}(s)$.

We say that the interpreted system $\mathcal{I} = (\mathcal{R}, \pi)$ *models agent i as a Dolev-Yao agent* if for all $m \geq 0$, and for all messages m,

- if $\pi(r, m)(\mathsf{extract}_i(\mathsf{m})) = \mathbf{true}$, then $\mathsf{m} \in \mathit{can\_compute}(r_i(m))$, and

- if $r_i(m+1) = r_i(m) \cdot \mathsf{sent}_i(\mathsf{m})$, then $\mathsf{m} \in \mathit{can\_compute}(r_i(m))$.

We say that $\mathcal{I}$ *models Dolev-Yao agents* if for every agent i it models agent i as a Dolev-Yao agent. Note that this definition does not specify how extract handles formulas (recall that messages include formulas).

The extract operator is similar in spirit to the notion of *algorithmic knowledge* (Fagin, Halpern, Moses, and Vardi 1995) used in (Halpern and Pucella 2002) to model adversaries. It deals with adversaries' resource limitations by explicitly describing what they can compute. There are other ways of dealing with the fact that agents are not logically omniscient, and cannot compute all the logical consequences of their observations. For example, AT assume that agents do not observe the full details of messages they receive, so that, for example, if they receive $\{\mathsf{m}\}_k$ for some key k whose inverse they have not extracted, what appears in their local state is $\mathsf{recv}_i(\perp)$. This approach is meant to capture the reasonable intuition that all long strings of bits look the same. Syverson and van Oorschot (1994) use a similar approach. While we believe that this approach may have its uses, it does not directly capture *computational* limitations; it seems better for capturing perceptual limitations. We suspect that for security-related applications, operators like extract will prove more useful. We discuss this issue at greater length in the full paper.

3 Revisiting BAN

The BAN logic defines formulas for reasoning about trust in security protocols, and inference rules to derive properties of these protocols. In this section, we show how we can interpret the constructs of BAN logic by essentially rewriting them into the simpler primitives of our logic. Doing so exposes most assumptions underlying these constructs. We then show that our interpretation of BAN is faithful to the original by showing that all the BAN inference rules remain sound under our interpretation. In Section 4, we show that we can use our semantics to establish the same kinds of properties that BAN can establish. However, we stress that the interpretation given in this section is not the only possible interpretation of BAN; indeed, we give another one in Section 6. Our main point in these sections is to illustrate that the intuitions underlying authentication logic can be given a rigorous semantic foundation using the well understood notions of our logic. We deal with BAN logic here primarily because it, amongst the class of authentication logics, has the simplest set of primitives. We believe we could similarly reconstruct the richer logics in this class.

3.1 Definition of BAN

We reformulate the syntax of BAN, along the lines of AT. The set of formulas and set of messages are defined by mutual induction. Messages are defined just as in Section 2.2.[4] Formulas are defined by the following grammar:

$$\begin{array}{lll} \mathsf{m} & ::= & t \mid k \mid n \mid \mathsf{m} \cdot \mathsf{m} \mid \{\mathsf{m}^i\}_k \mid F \mid i \\ F & ::= & i \text{ } \mathbf{believes} \text{ } F \mid i \text{ } \mathbf{controls} \text{ } F \mid i \text{ } \mathbf{sees} \text{ } \mathsf{m} \mid i \text{ } \mathbf{said} \text{ } F \mid i \overset{k}{\leftrightarrow} j \mid \overset{k}{\mapsto} j \mid i \overset{\mathsf{m}}{\rightleftharpoons} j \mid \mathbf{fresh}(\mathsf{m}). \end{array}$$

The superscript i in m^i represents a "from"-field, intended to indicate the original sender of the message. The intuitive reading of the formulas is as follows:

- i **believes** F holds if principal i believes formula F;

[4] The BAN logic has an extra message construct, $\langle \mathsf{m}^i \rangle_{\mathsf{m}'}$, representing a message m combined with a *secret* m'. This is essentially a concatenation $\mathsf{m} \cdot \mathsf{m}'$, except that we need to account for the "from"-field i, intended to indicate the original sender of the message. Since we are not using this construct in the remainder of this paper, and since accounting for it complicates the presentation without adding insight, we defer its treatment to the full paper. Intuitively, we can consider the formula $\langle \mathsf{m}^i \rangle_{\mathsf{m}'}$ as equivalent to $\{\mathsf{m} \cdot \mathsf{m}'^i\}_{k_s}$, where k_s is a key known to all principals and adversaries.

- i **controls** F means that principal i is an authority on or has authority or jurisdiction over F;
- i **sees** m indicates that i has received a message containing m;
- i **said** F indicates that principal i at some time sent a message containing F and (if it was sent recently) that i believes F;
- $\mathbf{fresh}(\mathsf{m})$ indicates that the message m is fresh, that is, it was sent recently.
- $i \overset{k}{\leftrightarrow} j$ means that principals i and j can use key k to communicate (and that the key is a good key);
- $\overset{k}{\mapsto} j$ means that the key k is j's public key (and that the key is a good key);
- $i \overset{\mathsf{m}}{\rightleftharpoons} j$ means that message m is a secret between principals i and j.

(In their revision of BAN logic, AT slightly extend BAN, adding formulas such as i **says** F (i recently said F), and i **has** k (i is in possession of key k). SW also modify the original BAN logic somewhat. For ease of exposition, we do not consider these modifications here.)

Inference rules are used in BAN to derive new formulas from others. These inference rules are given in Appendix A. To illustrate these rules, consider the following rule R3, to which we will return in the next sections.

$$\frac{i \textbf{ believes } \mathbf{fresh}(F) \quad i \textbf{ believes } (j \textbf{ said } F)}{i \textbf{ believes } j \textbf{ believes } F}$$

It says that if i believes both that F is fresh and that j said it, then i also believes that j believes F. This inference rule encapsulates the assumptions that statements do not change their truth value in short intervals of time and that agents say only things that they believe to be true.

The BAN operators seem to capture the intuitive notions we want to take into account when reasoning about security protocols: belief, trust, freshness, jurisdiction. The criticisms aimed at BAN do not argue this point, but rather focus on the question of foundation (that the semantics was imprecise, making it hard to justify the results of analyses), and of pragmatics, especially with respect to the how the logic was used in protocol analysis. As we said in the introduction, we have one more criticism, namely, the choice of primitive notions. We believe that there should be relatively few primitive notions, with all the rest defined in terms of them. We believe that our logic overcomes these deficiencies, while still being able to express the key features of the BAN operators.

3.2 A knowledge interpretation of BAN

We now define a translation from BAN formulas to formulas in our logic. In the translation, let F^K be the result of translating the formula F. Since formulas include messages and are messages, we also need to translate messages; let m^K be the result of translating m to our logic.

- The translation of messages that are not formulas is the obvious one: $\mathsf{t}^K = \mathsf{t}$, and $(\mathsf{m}_1 \cdot \mathsf{m}_2)^K = {\mathsf{m}_1}^K \cdot {\mathsf{m}_2}^K$. We translate encryptions $\{\mathsf{m}^i\}_k$ by treating the "from"-field as concatenated to the end of the encrypted message: $\{\mathsf{m}^K \cdot i\}_k$.
- We translate $(i \textbf{ believes } F)^K$ as $K_i F^K$. This way of capturing belief can be viewed as being in the spirit of the definitions of AT. AT essentially defined belief as "truth at all 'good' points". For us, the good points are all the points that the agent considers possible. (In Section 6, when we deal with guessing adversaries, we weaken this assumption.) AT also had an added complication to deal with resource-boundedness of agents (so that they cannot break cryptographic keys, for example). For us, this problem is handled by $\mathsf{extract}$. We discuss this in more detail below.

- $(i \textbf{ sees } \mathsf{m})^K$ is $\mathsf{extract}_i(\mathsf{m}^K)$. As we said, $\mathsf{extract}$ is doing all the work of deciding what information can be extracted from the set of messages received.

- The translation of i **said** F is somewhat complicated, since the **said** operator is conflating a number of distinct notions that we must untangle to express it in our logic. For one thing, it means that a message is sent. However, we do not wish to interpret an agent as saying F when it sends a message containing F encrypted under a key that the agent does not possess (this situation arises in some protocols that rely on agents to forward server generated tickets.) To capture this intuition, our translation of **said** also incorporates the notion of extraction; nothing is said that was not extracted at the time when the message was sent. This can be captured by the translation $(i \textbf{ said } F)^K$ is $\mathsf{sent}_i(F^K) \wedge \boxminus(\neg\mathsf{sent}_i(F^K) \wedge \bigcirc\mathsf{sent}_i(F^K) \Rightarrow \mathsf{extract}_i(F^K))$. Furthermore, **said** incorporates the notion of belief; BAN assumes that all formulas said recently by i are believed by i. We capture this by imposing an "honesty" restriction on the systems for which our translation of the BAN rules are sound. While it serves our purposes in this paper, the above translation does not quite capture all the subtleties in an intuitive reading of the term **said**. For instance, an agent should not be considered to have said m if it sent $\{\mathsf{m}\}_k$ but did not possess k at the time. But note that it may nevertheless have been able to extract m from some other message. (One way to address such subtleties would be to add a notion of quotation, as is done in AT.)

- Capturing that k is a good key between i and j depends on what we mean by "good key". There are at least two interpretations. According to AT, it should be interpreted as "no one but i and j sends messages encrypted with k" for the length of the protocol interaction. Another possible interpretation is that no one other than possibly i and j has extracted the key. We can encode both interpretations, but we choose the second as it seems more in keeping with the intuitive reading of k being a good key. Hence, we take $(i \stackrel{k}{\leftrightarrow} j)^K$ to be $\mathsf{extract}_i(k) \wedge \mathsf{extract}_j(k) \wedge \left(\bigwedge_{i' \neq i,j} \neg\mathsf{extract}_{i'}(k)\right)$. Note that both BAN and AT interpret k being a good key as a statement that also talk about the future; in essence, assuming k is a good key remains true throughout a protocol interaction. We could capture such an interpretation by prefixing the translated formula by a $\Box$ operator. However, this interpretation precludes the analysis of protocols that leak the key value. (See Nessett (1990) for a discussion of such protocols.) Intuitively, we would like the analysis to reveal such leaks, and not presuppose that such leaks do not happen. There are other subtleties in the interpretation of "good key". Many protocols studied in the literature assume the existence of a key server in charge of distributing session keys to principals. In that setting, a good key is not only known only to the principals exchanging messages, but also of course to the server that initially distributed the key. In other words, there is a sense in which the interpretation of "good key" depends also on the kind of protocol being executed (i.e., whether it requires a key server or not). One advantage of expressing the concept of "good key" using simpler operators is that we can change the interpretation when needed.

- $\stackrel{k}{\mapsto} j$ says that k is j's public key, and that the key is a good key. The formula is intended to mean that only j knows the key k^{-1}. Thus, its translation is similar in spirit to that of the formula for shared keys; we take $(\stackrel{k}{\mapsto} j)^K$ to be $\mathsf{extract}_j(k^{-1}) \wedge \left(\bigwedge_{j \neq i} \neg\mathsf{extract}_j(k^{-1})\right)$.

- $i \stackrel{\mathsf{m}}{\rightleftharpoons} j$ says that m is a secret shared by i and j. This is intended to mean that only i and j know m, that is, that only they can extract it. Hence, we take $(i \stackrel{\mathsf{m}}{\rightleftharpoons} j)^K$ to be $\mathsf{extract}_i(\mathsf{m}^K) \wedge \mathsf{extract}_j(\mathsf{m}^K) \wedge \left(\bigwedge_{i' \neq i,j} \neg\mathsf{extract}_{i'}(\mathsf{m}^K)\right)$. Note that this assumes that both i and j can extract m.

- A message is fresh if it was not extracted, except possibly recently. As we did for the translation of **said**, we leave it to the user to decide what counts as "recently", by choosing a suitable t. Thus,

$\mathbf{fresh}(\mathsf{m})^K$ is just $\bigwedge_i \neg\ominus^t \mathsf{extract}_i(\mathsf{m}^K)$. This translation suggests that it may be useful to be able to talk about a message being fresh with respect to j, which would simply be $\neg\ominus^t \mathsf{extract}_j(\mathsf{m}^K)$. For example, note that for the rule R3 (the only rule that uses freshness in a serious way) it suffices that i believe that m is fresh for j for the conclusion to hold. Note that some formal approaches to reasoning about security advocate the interpretation of a message being fresh as meaning that it never existed before. This seems needlessly strong, and removed from reality. For freshness, it is sufficient for a value to simply be unknown.

- We interpret i **controls** F by the statement "if i believes F, then F must be true". This captures, to some extent, the intuition that i is an authority on φ. Although this seems to be a reasonable consequence of **controls**, and will suffice for our results, it seems somewhat unreasonable to make it the *definition* of **controls**. The problem is that this makes i **controls** F true if i does not believe F, whether or not i is an authority on F. The intuition used by AT and followed by SW is that, roughly speaking, if i says F, then F is true. (AT further say that this is true at all times.) This interpretation suffers from the same problems. It make i **controls** F true if i does not say F or if F is true (or if i never says F or if F is always true, in the case of AT's interpretation). Again, this does not seem to correspond to our intuition about **controls**. The real problem here is that "if ... then" is interpreted as material implication, so that a formula such as $\varphi \Rightarrow \psi$ is automatically true if φ is false or ψ is true. What would really be more appropriate here is interpreting "if ... then" causally, so that "if φ then ψ" essentially means "φ causes ψ to hold". Such an interpretation can be captured using what philosophers call *conditional implication*. While capturing this is beyond the scope of this paper, we remark that an approach to doing so in the context of the multiagent systems framework can be found in (Halpern and Moses 1998). Following this discussion, since in the current translation we interpret BAN belief as knowledge, we take $(i\ \mathbf{controls}\ F)^K$ to be $K_i(F^K) \Rightarrow F^K$. Notice that this last formula is a tautology, since one property of knowledge is that knowledge entails truth. Hence, in the current translation, $(i\ \mathbf{controls}\ F)^K$ is true.

To what extent does the translation above capture BAN? As the following theorem shows, it does provided that we use the appropriate extract function, and make appropriate assumptions about agents.

We say that agents *have no additional prior information* in an interpreted system $\mathcal{I}$ if the initial states of all agents includes only public keys and their own private key, and in the case of nonadversary agents, the nonces needed by the protocol being analyzed. Finally, we need to make precise the intuition that agents tell the truth, since BAN assumes that when a (nonadversary) agent sends a formula, she believes it.[5] Without this requirement, we cannot ensure the validity of R3. Implicit in the notion of honesty is the idea that an agent does not forge "from" fields in messages. To capture this, we need to refine our interpretation of extract, by refining the definition of the function $\mathit{can\_compute}$. For an agent i, define the function $\mathit{can\_compute}_i$ inductively, by the same set of conditions as in the definition of $\mathit{can\_compute}$, except that the condition for $\mathsf{m} = \{\mathsf{m}_1\}_k$ is replaced by

- $\mathsf{m} = \{\mathsf{m}_1 \cdot i\}_k$, with $\mathsf{m}_1 \in \mathit{can\_compute}_i(s)$ and $k \in \mathit{can\_compute}_i(s)$.

Intuitively, this change ensures that when the agent constructs an encrypted message, it includes a "from" field set to its own name. We say that agent i is a *nonforging Dolev-Yao* agent in an interpreted system $\mathcal{I} = (\mathcal{R}, \pi)$ if it satisfies the following refinement of the Dolev-Yao conditions: for all $m \geq 0$, and for all messages m,

[5] To account for the fact that a principal may forward a formula without necessarily believing in it, AT introduces a special notation (quotation) for forwarded message; this does not remove the need for honesty, in that the original sender of the formula still must have believed the formula before sending it. For ease of exposition, we do not deal with forwarded messages here; they do not cause any difficulties. The fact that BAN requires honesty was already pointed out by AT. They avoid the need for honesty by replacing R3, the only rule for which honesty is essential, by other rules.

- $\pi(r, m)(\mathsf{extract}_i(\mathsf{m})) = \mathbf{true}$ iff $\mathsf{m} \in \mathit{can\_compute}_i(r_i(m))$, and
- if $r_i(m+1) = r_i(m) \cdot \mathsf{sent}_i(\mathsf{m})$, then $\mathsf{m} \in \mathit{can\_compute}_i(r_i(m))$.

We say that an agent i is an *honest Dolev-Yao agent* in an interpreted system $\mathcal{I} = (\mathcal{R}, \pi)$ if

1. i is a nonforging Dolev-Yao agent in $\mathcal{I}$, and
2. for all φ and all points (r, m), if i sends a message m at round m (i.e., between times $m-1$ and m), then if $\varphi \sqsubseteq \mathsf{m}$ (i.e., φ is a submessage of m), and $(\mathcal{I}, r, m-1) \models \mathsf{extract}_i(\varphi)$, then $(\mathcal{I}, r, m-1) \models K_i(\bigwedge_{s \leq t} \bigcirc^s \varphi)$.

The intuition for the last condition is that an agent will only say things that it knows will still be true some time in the near future when its message is received. Again, this is parameterized by a time t, which should be taken as the same time parameter used to interpret freshness. Note that, by the assumption of perfect recall implicit in our semantics, $K_i \bigwedge_{s \leq t} \bigcirc^s(\varphi)$ implies $\bigwedge_{s \leq t} \bigcirc^s(K_i \varphi)$, so the recipient is guaranteed that i still knows what it said.

We now want to show that this translation preserves the validity of the BAN inference rules. Strictly speaking, we translate *instances* of those inference rules, where the agent names and formulas appearing in the inference rule are instantiated to particular values. An instance r of a BAN inference rule has the form "from F_1 (and F_2) infer F". We translate this to the formula r^K: $F_1^K (\wedge F_2^K) \Rightarrow F^K$. We use the notation r_{ij}^K to emphasize that the formulas in an instance r refers to agents i and j.

Theorem 3.1: *For every instance r_{ij} of the BAN inference rule Rn, for $n = 1, 2, 3$, the formula r_{ij}^K is valid in systems that model Dolev-Yao agents that have no additional prior information and where agents i and j are honest. For every instance r of the BAN inference rule Rn, for $n \geq 4$, the formula r^K is valid in systems that model Dolev-Yao agents that have no additional prior information.*

The restriction to Dolev-Yao agents is hardwired into the definitions of $\mathbf{said}$ and $\mathbf{sees}$ by AT and SW. We model it using $\mathsf{extract}$ instead. This means that our logic can be used to deal with other adversaries besides those that satisfy the Dolev-Yao properties, without changing the underlying syntax and semantics. Similarly, rather than hardwiring honesty into the definition of $\mathbf{said}$, we build it into the class of structures.

4 An example: the Needham-Schroeder authentication protocol

The logic we introduced in Section 2 provides us with a language in which to express properties of multiagent systems. As we saw in Section 3, this logic can encode an interpretation of BAN based on knowledge. In this section, we study the Needham-Schroeder authentication protocol using that interpretation. This protocol has been well-studied in the literature, and provides a baseline from which to evaluate our BAN translation from Section 3, and the probabilistic interpretation from Section 6. The aim is to examine to what extent we can take specifications of the protocol expressed in the BAN logic, and establish that the appropriate translations of these specifications hold when expressed in our logic against models of the Needham-Schroeder protocol in the presence of various adversaries.

4.1 Description of the protocol

The Needham-Schroeder protocol is a simple protocol introduced by Needham and Schroeder (1978) to perform authentication with public-key cryptography. Consider the following simplified version of the protocol, which we call the NS protocol. It presumes that principals A and B know each other's public keys, denoted k_a and k_b, respectively.

$$\begin{array}{l} A \rightarrow B : \{n_a, A\}_{k_b} \\ B \rightarrow A : \{n_a, n_b\}_{k_a} \\ A \rightarrow B : \{n_b\}_{k_b} \end{array}$$

Here, n_a and n_b are nonces, and k_a and k_b are the public keys of principals A and B, respectively. Thus, initially principal A sends B the nonce n_a and his name A, encoded using B's public key (so that B can decrypt it). B sends A back his nonce, together with another nonce n_b, all encoded using A's public key. Finally, A sends B back B's nonce n_b, encoded with B's public key. The claim is that, at the end of this protocol, A and B know that they have communicated with each other. The protocol also establishes n_a and n_b as secrets known only to A and B.

In order to analyze NS, we need to write down the protocols involved in a way suitable for our framework. The NS exchange actually involves two protocols: the NS Initiator protocol ($P^I_{A,B}$), executed by the principal initiating the authentication request, and the NS Responder protocol $P^R_{A,B}$, executed by the principal responding to the authentication request. Recall that in our framework, a protocol is a function from local states to actions. The Initiator protocol $P^I_{A,B}$ (when the initiator A communicates with B) is

$$P^I_{A,B}(\langle\{\mathtt{n_a}, \mathtt{k_b}, \mathtt{k_a}\}\rangle) = \mathsf{send}(B, \{\mathtt{n_a}, A\}_{\mathtt{k_b}})$$

$$P^I_{A,B}(\langle\{\mathtt{n_a}, \mathtt{k_b}, \mathtt{k_a}\}, \mathsf{recv}(\{\mathtt{n_a}, \mathtt{n_b}\}_{\mathtt{k_a}})\rangle) = \mathsf{send}(B, \{\mathtt{n_b}\}_{\mathtt{k_b}}).$$

Roughly speaking, the Initiator protocol, when the local state is only made up of the initial information, will send the nonce bound to the identifier $\mathtt{n_a}$ (as given by the binding in the initial state of the agent), and when the agent receives the first message prescribed by the NS protocol, it uses pattern matching on the received nonce, binding the identifier $\mathtt{n_b}$ to that value, and sends the value back.

The Responder protocol $P^R_{A,B}$ is similar:

$$P^R_{A,B}(\langle\{\mathtt{n_b}, \mathtt{k_b}, \mathtt{k_a}\}, \mathsf{recv}(\{\mathtt{n_a}, A\}_{\mathtt{k_b}})\rangle) = \mathsf{send}(A, \{\mathtt{n_a}, \mathtt{n_b}\}_{\mathtt{k_a}}).$$

We assume a context where the adversary X is not an insider. In other words, the NS Initiator will never try to authenticate herself to X. The adversary can of course intercept messages, but no message is addressed explicitly to X. (Lowe (1995) shows that the original NS protocol is vulnerable to a man-in-the-middle attack when the adversary is an insider.) In general, A and B are parameters. For the sake of illustration, we consider only a context where there are just two agents, who run the protocol just once. This enables us to treat A and B as constants. We later discuss the question of what is needed to generalize the results to multiple runs and additional agents.

4.2 Analyzing the Needham-Schroeder protocol

In this section, we analyze Needham-Schroeder in the presence of a Dolev-Yao adversary. NS against such an adversary can be analyzed using the knowledge-based interpretation of BAN.

The system We construct a system $\mathcal{I}^{DY}$ as follows. Given nonces n_a and n_b and keys k_a and k_b, let $s_I(n_a, n_b, k_a, k_b) = \langle\{\mathtt{n_a} \mapsto n_a, \mathtt{k_a}^{-1} \mapsto k_a^{-1}, \mathtt{k_a} \mapsto k_a, \mathtt{k_b} \mapsto k_b\}\rangle$. Recall that the notation $\mathtt{n} \mapsto n$ says that the nonce n is bound to the identifier $\mathtt{n}$, and similarly for keys. Intuitively, $s_I(n_a, n_b, k_a, k_b)$ is the Initiator's initial state when its nonce is n_a, the Responder's nonce is n_b, its key is k_a, and the Responder's key is k_b. It knows its nonce and the inverse to its key; it also knows the Responder's public key, but not its inverse. Similarly, let $s_R(n_a, n_b, k_a, k_b) = \langle\{\mathtt{n_b} \mapsto n_b, \mathtt{k_b}^{-1} \mapsto k_b^{-1}, \mathtt{k_a} \mapsto k_a, \mathtt{k_b} \mapsto k_b\}\rangle$ and $s_X(n_a, n_b, k_a, k_b) = \langle\{\mathtt{k_a} \mapsto k_a, \mathtt{k_b} \mapsto k_b\}\rangle$. Let

$$I = \{(s_I(n_a, n_b, k_a, k_b), s_R(n_a, n_b, k_a, k_b), s_X(n_a, n_b, k_a, k_b)) : n_a, n_b \in \mathcal{N}, k_a, k_b \in \mathcal{K}\}.$$

We take $\mathcal{P}^{DY}$ to be the set of all joint protocols of the form $(P^I_{A,B}, P^R_{A,B}, P)$, where P is a deterministic protocol consistent with the Dolev-Yao capabilities of the adversary. Let $\mathcal{I}^{DY}$ be the system $\mathcal{R}(\mathcal{P}^{DY}, I)$. Hence, for every choice of n_a, n_b, k_a, k_b and for every choice of a Dolev-Yao protocol P for the adversary, $\mathcal{I}^{DY}$ contains a run where agent A uses nonce n_a and private key k_a^{-1}, agent B uses nonce n_b and private key k_b^{-1}, and where the adversary starts with the knowledge of the public keys and runs protocol P.

The specifications Using BAN to reason about a protocol involves specifying the assumptions under which the analysis should proceed (the initial beliefs of the principals), and deriving conclusions via the inference rules. Let us talk now about the specifications of Needham-Schroeder using BAN (Burrows, Abadi, and Needham 1990a). We have only considered the specifications that do not involve the server, as we have simplified our account of the protocol to do away with the server.

Say that F is an *assumption of the BAN analysis for NS* if it is one of the following formulas:

$$\begin{array}{ll}
A \textbf{ believes } \overset{k_a}{\mapsto} A & B \textbf{ believes } \overset{k_b}{\mapsto} B \\
A \textbf{ believes } \textbf{fresh}(n_a) & B \textbf{ believes } \textbf{fresh}(n_b) \\
A \textbf{ believes } A \overset{n_a}{\rightleftharpoons} A & B \textbf{ believes } B \overset{n_b}{\rightleftharpoons} B
\end{array}$$

As we shall see, these assumptions (or rather, their translation) are true at the initial states of our systems. Note that the original BAN assumptions for this protocol include A **believes** $A \overset{n_a}{\rightleftharpoons} B$ and B **believes** $A \overset{n_b}{\rightleftharpoons} B$. The intuition was that no one but A and B would ever know n_a and n_b. We do not encode secrecy as "at all future points the secret holds", to allow for a protocol that may reveal secrets. Secrecy at time 0 is assumed only with respect to the original "owner" of the nonce. Hence, $A \overset{n_a}{\rightleftharpoons} A$, in an abuse of notation, indicates that only agent A knows the nonce n_a (and similarly for B and n_b). The original BAN assumptions A **believes** $A \overset{n_a}{\rightleftharpoons} B$ and B **believes** $A \overset{n_b}{\rightleftharpoons} B$ are now properties that we want to hold as conclusions, i.e., the protocol will not reveal those secrets.

Say that F is a *conclusion of the BAN analysis for NS* if it is one of the following formulas:

$$\begin{array}{ll}
A \textbf{ believes } \overset{k_b}{\mapsto} B & B \textbf{ believes } \overset{k_a}{\mapsto} A \\
A \textbf{ believes } B \textbf{ believes } A \overset{n_b}{\rightleftharpoons} B & B \textbf{ believes } A \textbf{ believes } A \overset{n_a}{\rightleftharpoons} B \\
A \textbf{ believes } A \overset{n_a}{\rightleftharpoons} B & B \textbf{ believes } A \overset{n_b}{\rightleftharpoons} B \\
 & B \textbf{ believes } A \textbf{ believes } B \textbf{ believes } A \overset{n_b}{\rightleftharpoons} B
\end{array}$$

While assumptions are meant to hold at the initial state of the system, before the protocol executes, the conclusions are meant to hold at the points where the protocol has finished executing. At least, this is one intuition that is worth pursuing, and as we shall see, is indeed an appropriate intuition *when belief is interpreted as knowledge*. In Section 6, when we interpret belief probabilistically, we will need to refine this view.

Formal results As far as $\mathcal{I}^{DY}$ is concerned, it is straightforward to show that the translation of the initial assumptions of an NS protocol interaction, according to the original BAN analysis, do indeed hold before the protocol is executed in our framework.

Proposition 4.1: *Let F be an assumption of the BAN analysis for NS. For all runs r in $\mathcal{I}^{DY}$ such that* $\mathtt{n_a} \mapsto n_a$ *and* $\mathtt{k_a^{-1}} \mapsto k_a^{-1}$ *are in* $r_A(0)$, $\mathtt{n_b} \mapsto n_b$ *and* $\mathtt{k_b^{-1}} \mapsto k_b^{-1}$ *are in* $r_B(0)$, $n_a \neq n_b$ *and* $k_a \neq k_b^{-1}$, *we have* $(\mathcal{I}^{DY}, r, 0) \models F^K$ *where we interpret freshness with respect to a time interval of* $t \geq 1$.

Furthermore, we can show that our translation validates the goals of authentication for NS as given by BAN. Following the previous discussion, conclusions are meant to hold at the end of a protocol execution. In other words, if F is a conclusion, then the formula $end \Rightarrow F^K$ should hold at every point of the system, where end identifies the end of the protocol.

Proposition 4.2: *Let F be a conclusion of the BAN analysis for NS. For all runs r in $\mathcal{I}^{DY}$ such that* $\mathtt{n_a} \mapsto n_a$ *and* $\mathtt{k_a^{-1}} \mapsto k_a^{-1}$ *are in* $r_A(0)$, $\mathtt{n_b} \mapsto n_b$ *and* $\mathtt{k_b^{-1}} \mapsto k_b^{-1}$ *are in* $r_B(0)$, $n_a \neq n_b$, *and* $k_a \neq k_b^{-1}$, *we have* $(\mathcal{I}^{DY}, r, 0) \models \Box(\mathsf{recv}_B(\{n_b\}_{k_b}) \Rightarrow F^K)$.

The results of this section were established directly, by checking that the appropriate formulas hold at appropriate points of the models. In particular, we have not tried to derive the conclusions deductively from the assumptions. Note also that we have worked directly from the original description of the protocol: we have not made use of the step (generally considered questionable) of "protocol idealization"; we have not transformed the messages of the protocol, as is done in (Burrows, Abadi, and Needham 1990a), nor have we captured the idealization step in formulas, as in some later approaches (Syverson and van Oorschot 1994; Wedel and Kessler 1996). Nor do we need to make use of the honesty assumption (as no formulas are being sent in the protocol!) Our results suggest that, although the means of BAN logic reasoning are open to doubt, it is possible to give the ends a rigorous justification. The problem with BAN is not that it is based on epistemic notions.

5 Discussion

While the results of the previous section are encouraging, we need to be careful about their interpretation. The system $\mathcal{I}^{DY}$ misses out on some of the key intuitions underlying the structure of the NS protocol. Intuitively, what the responder should learn from the first message of the protocol is that some agent is attempting to authenticate itself as A. The remaining two messages are designed to convince the responder that this agent is in fact A. The intuitive reason that the first message does not suffice is that it could have been constructed by the adversary. However, this is not reflected in our definition of a Dolev-Yao adversary, which prevents the adversary from attempting to authenticate himself to B as A directly, because it can send only messages constructed from what it has been able to extract, and it is initially unable to extract any nonce. Indeed, with respect to our model, we can prove the rather odd fact that, after receiving the first message, B *knows* that A sent the message! This problem would persist even if we were to enrich our model to allow additional agents, engaging in multiple executions of the NS protocol. On the assumption (which has been argued by some to be inherent to NS) that no agent attempts to run the protocol with the adversary, B learns from any message of the form $\{n_x, X\}_{k_b}$ that X has sent this message at some time. This is so simply because the adversary would never be able to extract any nonce, hence can only be replaying a genuine initiation message.

This problem points to some inherent limitations of the Dolev-Yao model as we have defined it. The issues, it turns out, run deeper than the above remarks suggest, and affect not just our results but other work in the literature as well. We show this by an examination of Paulson's work (Paulson 1998). We believe that similar issue arise in other works based on the Dolev-Yao model, although we have not examined them carefully.

Paulson's approach to cryptographic protocol verification is to formally describe the traces of a protocol that are obtained when a set of agents operates in the context of the Dolev-Yao adversary model, and to use induction to prove properties of these traces. Paulson's formulation of the Dolev-Yao model, like ours, enables the spy (that is, the adversary, in Paulson's setting) to use a nonce as part of the process of constructing a message only if it has been able to extract that nonce from messages seen or decrypted using keys that it knows or has extracted. (The model does have a notion of "guessable number", but this plays no role in the NS protocol analysis.) The properties Paulson proves are temporal, i.e., are properties only of the single trace in question, rather than properties that relate different traces. This means that any epistemic content of the results is left implicit.

We illustrate this point by considering one of the results Paulson establishes. Paulson considers the version of the Needham-Schroeder protocol defined above, within a context where multiple agents may execute the protocol multiple times. Concerning the first two messages of the protocol, Paulson proves a lemma that he glosses as follows: "The guarantees for A are that her nonce remains secret—from the spy—and that B is present. The latter follows from the former, for if the spy does not know n_a then he could not have sent message 2." (Note that this informal explanation is expressed in terms of knowledge.) For the detailed statement of what is proved one needs to look in Paulson's framework files,[6] where one finds the following formulation of the result:

```
(*Authentication for A: if she receives message 2 and has used NA
  to start a run, then B has sent message 2.*)

theorem A_trusts_NS2:
     "[Says A  B (Crypt(pubK B) {Nonce NA, Agent A}) <in> set evs;
       Says B' A (Crypt(pubK A) {Nonce NA, Nonce NB}) <in> set evs;
       A <notin> bad;  B <notin> bad;  evs <in> ns_public]
      --> Says B A (Crypt(pubK A) {Nonce NA, Nonce NB}) <in> set evs"
```

Using our notation, if A and B are not spies, A has in fact sent to B the message $\{n_a, A\}_{k_b}$, and receives from some agent B' (the agents and nonces in the above are implicitly universally quantified) the message $\{n_a, n_b\}_{k_a}$, then this message was in fact sent by B at some time. (The agent B' might still be the spy, relaying B's message.)

The proof of this result relies crucially on some assumptions that Paulson builds into the model. One of the critical assumptions is that nonces are unique. That is, if a protocol step calls for the generation of a nonce, then the nonce produced is generated just once in the history. Any other instance of nonce generation, by the same agent, or by another, produces a different nonce. This assumption is necessary, because if n_a is repeated, the spy might be replaying a message of the form $\{n_a, n_b\}_{k_a}$ sent by some other agent C in a previous run of the protocol.

However, nonce uniqueness is not enough on its own to enable us to safely interpret this result as stating that A is assured of B's presence. For, suppose that the agents use the following "name+timestamp" approach to construct nonces: in a message sent at time k, agent X uses the nonce (X, k). This guarantees the uniqueness property. However, an adversary could easily exploit this behaviour to predict the message that A is expecting in response to its initial message to B. If A's message is sent at time k, the adversary could construct the message $\{(A, k), (B, k')\}_{k_a}$ without B's participation, and this would pass the test that A performs on receiving the message. What this illustrates is that the theorem relies crucially on another of the assumptions built into the model: that the spy does not guess nonces.

How adequate are these assumptions? The uniqueness assumption, taken literally, seems somewhat problematic. A single agent might, at some cost, keep a record of nonces used, but to scale this to multiple agents would seem to require a globally trusted uniqueness service, which is implausible. As for the assumption that the adversary uses only nonces that it is able to extract from messages seen, this seems an

[6] Available at `http://www.cl.cam.ac.uk/Research/HVG/Isabelle/library/HOL/Auth/index.html`.

extremely severe assumption. In the context of nonces implemented using the timestamp approach discussed above, we would hardly expect the adversary to permit itself to be so hampered.

What this points to is the fact that the Dolev-Yao model brushes under the carpet a crucial *epistemic* assumption: that the adversary does not have access to a source of information that would enable it to make reliable predictions about the nonces being used by the agents.[7] The answer to both objections above is, of course, that the intention of the protocol is that nonces be randomly generated. This ensures that nonce repetitions, while not impossible, are a low probability event. Moreover, randomness provides a way of preventing intelligent guessing of the kind illustrated in the timestamp nonce example. Randomization is a way of decreasing the amount of *information* available to the spy.

We see that even before we consider the subtle issues surrounding the capabilities of the adversary to extract information from encrypted messages, there are subtleties regarding the information carried by nonces that need to be understood. Note also that Paulson's work has not circumvented the epistemic intuitions underlying BAN. To fully understand what has been proved by Paulson's analysis, and what it means for the security of the protocol, it seems to be necessary to reinstate epistemic language.

6 A probabilistic interpretation

In Section 3 we saw that we could use our logic to interpret the BAN operators. In this translation, we interpreted the BAN $\mathbf{believes}$ operator as knowledge. Intuitively, if the adversary cannot guess, then the information that she can derive is restricted to what she sees in the intercepted messages. AT point out that this knowledge interpretation is inappropriate if the adversary is allowed guesses. AT goes on to define belief as a form of defeasible knowledge. In this section, we use a much more straightforward approach: we define belief as knowing with high probability. To accommodate this view, we need to introduce probabilities in our framework.

6.1 A probabilistic interpretation of BAN

To capture probability, we consider probabilistic systems, where there are probability distributions on the set $\mathcal{R}$ of runs of the system. We do not assume a single probability distribution on $\mathcal{R}$, since that would require us to put a probability on the possible protocols that the adversary is using. Rather, we partition $\mathcal{R}$ into subsets, called cells; intuitively, each cell corresponds to factoring out all the choices we would like to view as nondeterministic (such as the choice of protocol, or the keys used by the participants in a protocol). We assume that there is a probability μ_C on the runs $\mathcal{R}_C$ of each cell C. (See (Halpern and Tuttle 1993) for further discussion of this approach.)

We augment our logic with formulas of the form $\mathrm{Pr}_i(\varphi) \geq \alpha$, where $\alpha \in [0,1]$, for reasoning about probabilities. Intuitively, $\mathrm{Pr}_i(\varphi) \geq \alpha$ means that the formula φ holds with probability at least α, according to agent i.

To give semantics to formulas of the form $\mathrm{Pr}_i(\varphi) \geq \alpha$ at a point (r,m), we use the following approach, taken from (Halpern and Tuttle 1993). Let $\mathcal{K}_i(r,m)$ be the set of points that agent i cannot distinguish from (r,m), that is, the set $\{(r',m') \mid (r,m) \sim_i (r',m')\}$. For every run r, there is a unique cell C_r with $r \in C_r$. Let $\mathcal{C}(r)$ be the set of points from the runs in $\mathcal{R}_{C_r}$, the points in the runs in the same cell as r. The prior probability μ_C on the runs $\mathcal{R}_C$ of cell C induces a probability $\mu_{r,m,i}$ on the points in $\mathcal{K}_i(r,m) \cap \mathcal{C}(r)$ in a straightforward way. If $U \subseteq \mathcal{K}_i(r,m) \cap C(r)$, define

$$\mu_{r,m,i}(U) = \frac{\mu_C(\{r' : (r',m) \in U\})}{\mu_C(\{r' : (r',m) \in \mathcal{K}_i(r,m) \cap \mathcal{C}(r)\})}.$$

[7]To some extent the phrase "having no prior additional information" we used above is suggestive of what is being assumed, but the definitions above are too weak to capture what is needed.

The satisfaction of a formula $\Pr_i(\varphi) \geq \alpha$ becomes:

$$(\mathcal{I}, r, m) \models \Pr_i(\varphi) \geq \alpha \text{ iff } \mu_{r,m,i}(\{(r', m') \mid (\mathcal{I}, r', m') \models \varphi\} \cap \mathcal{K}_i(r, m) \cap \mathcal{C}(r)) \geq \alpha$$

Intuitively, this rule checks that the probability that the current point (r, m) is a world where φ holds (among the points that agent i cannot distinguish, *in the same cell as* r) is at least α.

We define the probabilistic knowledge operator $K_i^\alpha \varphi$ as an abbreviation for $K_i(\Pr_i(\varphi) \geq \alpha)$. This operator simply means that, essentially, no matter which cell C the agent thinks the current point is in, the probability of φ in that cell is at least α.

We can now define a translation of BAN formulas that interprets belief as probabilistic knowledge $K_i^\alpha \varphi$. More precisely, we define $F^{P,\alpha}$ to be the result of translating the formula F according to this probabilistic translation. The translation is exactly the same as F^K, except that we now use $K_i^\alpha(F^{P,\alpha})$ at every place in the translation where $K_i(F^K)$ occurs, that is, in the translation of **believes** and **controls**. Note that **controls**, under this translation, no longer corresponds to the proposition $true$.

The translation $(\cdot)^{P,\alpha}$ captures BAN in a more general setting than in the previous section. We now allow agents (especially adversaries) to make initial guesses at values of nonces and keys. Hence, what really changes is the assumption that agents have no prior information. The guesses can be thought of as representing prior information. We say that agents *have no additional prior information beyond guesses* in an interpreted system $\mathcal{I}$ if the initial states of all agents include only public keys, their own private keys, the nonces required by their protocol (in the case of nonadversary agents), and a finite set of other keys or nonces they have guessed.[8] We also require, following the discussion at the end of Section 5, the principals' nonces to be chosen at random according to the uniform distribution. This therefore presupposes that the space $\mathcal{N}$ of nonces is finite. (The purpose of choosing nonces at random is to decrease the amount of information available to the adversary. The uniform distribution is the one that minimizes this information.)

As in the last section, we need to somehow capture a notion of honesty in the system, to ensure the validity of R3. Honesty was defined with respect to knowledge K_i. We can in fact weaken this for our present translation, so that an agent need only believe what it says. We say that an agent i is an *essentially honest Dolev-Yao agent* in an interpreted system $\mathcal{I} = (\mathcal{R}, \pi)$ if

1. i is a nonforging Dolev-Yao agent in $\mathcal{I}$, and
2. for all φ and all points (r, m), if i sends a message m at round m (i.e., between times $m-1$ and m), then if $\varphi \sqsubseteq \mathsf{m}$ (i.e., φ is a submessage of m), and $(\mathcal{I}, r, m-1) \models \mathsf{extract}_i(\varphi)$, then $(\mathcal{I}, r, m-1) \models K_i^\alpha(\bigwedge_{s \leq t} \bigcirc^s \varphi)$.

Again, this is parameterized by a time t, which should be taken as the same time parameter used to interpret freshness.

Theorem 6.1: *For every instance r_{ij} of the BAN inference rule Rn, for $n = 1, 2, 3$, the formula $r_{ij}^{P,1}$ is valid in probabilistic systems that model Dolev-Yao agents that have no additional prior information beyond guesses and where agents i and j are essentially honest. For every instance r of the BAN inference rule Rn, for $n \geq 4$, the formula $r^{P,1}$ is valid in probabilistic systems that model Dolev-Yao agents that have no additional prior information beyond guesses.*

Theorem 6.1 states that the BAN inference rules are valid in our logic when BAN belief is interpreted as knowing with probability 1. One can check that validity fails for lesser values. Essentially, this is because

[8] We could generalize this setup by allowing the adversary guesses at arbitrary points, not simply at the initial state. This would require the introduction of a new event $\mathsf{guess}(\mathsf{m})$. Our results should hold in this generalized setting, as long as we allow only finitely many guesses at any point of the system. We follow the simpler approach for the sake of presentation.

K_i^α satisfies the following property: $K_i^{1-r}\varphi \wedge K_i^{1-s}\psi \Leftrightarrow K_i^{1-(r+s)}(\varphi \wedge \psi)$. It is only when $\alpha = 1$ that we have $K_i^\alpha\varphi \wedge K_i^\alpha\psi \Leftrightarrow K_i^\alpha(\varphi \wedge \psi)$, the property needed for validity. This seems overly restrictive, as we would really like to take BAN belief as meaning "knowing with high probability, close to 1". Details remain to be worked out, but there is a sense in which we can actually take this interpretation. Roughly speaking, we can show that if, from a collection Σ of formulas involving belief (interpreted as holding with probability 1), we can conclude a new fact φ about belief (again, interpreted as holding with probability 1), then for any $\epsilon > 0$, by taking the beliefs in Σ to hold with probability sufficiently close to 1, we can conclude φ probability at least $1 - \epsilon$. This argument allows us to reason qualitatively without having to worry about, for example, the length of an encryption key. It tells us that we can get conclusions to hold with arbitrarily high probability as long as, for example, keys are sufficiently long so that the assumptions in Σ hold with sufficiently high probability.

6.2 A probabilistic analysis of the Needham-Schroeder protocol

To analyze NS in the context of a Dolev-Yao adversary which is allowed some amount of guessing, we derive a system $\mathcal{I}^{GDY}$ modeling executions of the NS protocol where as before A and B use nonces n_a and n_b respectively and public keys k_a and k_b, but also where the adversary is able to guess nonces. (For simplicity, we do not allow the adversary to guess keys.) The only difference from the nonguessing case is the initial states. Specifically, the difference is in the state of the adversary, which needs to include his guesses. Formally, let $s_X(n_a^X, n_b^X, k_a, k_b) = \langle\{\mathtt{n}_\mathtt{a}^\mathtt{X} \mapsto n_a^X, \mathtt{n}_\mathtt{b}^\mathtt{X} \mapsto n_b^X, \mathtt{k}_\mathtt{a} \mapsto k_a, \mathtt{k}_\mathtt{b} \mapsto k_b\}\rangle$. The initial global states of the system are given by the set:

$$I^+ = \{\big(s_I(n_a, n_b, k_a, k_b), s_R(n_a, n_b, k_a, k_b), s_X(n_a^X, n_b^X, k_a, k_b)\big) : n_a, n_b, n_a^X, n_b^X \in \mathcal{N}, k_a, k_b \in \mathcal{K}\}.$$

Let $\mathcal{I}^{GDY}$ be the system $\mathcal{R}(\mathcal{P}^{DY}, I^+)$, where $\mathcal{P}^{DY}$ is the set of joint protocols defined above; let $\mathcal{I}^{GDY}_{P,n_a^X,n_b^X}$ be the subsystem consisting of all runs in which the adversary uses protocol P and nonces n_a^X and n_b^X.

To turn the system $\mathcal{I}^{GDY}$ into a probabilistic system, we define a distribution on the runs of each subsystem $\mathcal{I}^{GDY}_{P,n_a^X,n_b^X}$. This distribution is that generated from the distribution on the set I corresponding to taking each of the variables n_a and n_b to be uniformly distributed. Thus, we take the adversary's choice of protocol and nonces to be nondeterministic. Intuitively, this corresponds to the assumption that the adversary has complete freedom of choice (unconstrained by any distribution) concerning which attack it is going to mount. Note that we have taken the agents' choice of nonce to be random. Intuitively, this captures the fact that the agents are using random choices to protect themselves against the unknown but fixed adversary. (An analogous situation in game theory is the fact that a mixed strategy is often the best strategy, since it corresponds to maximal hiding of information from the adversary.)

In Section 4.2, we showed that the system $\mathcal{I}^{DY}$ corresponding to two principals running the Needham-Schroeder protocol in the presence of a Dolev-Yao adversary satisfied (the translations of) the BAN specifications of the protocol (Propositions 4.1 and 4.2). For $\mathcal{I}^{GDY}$, similar results hold, except using the probabilistic interpretation of belief. Note that the probability in the interpretation of belief depends on the size of the space of nonces $\mathcal{N}$.

Proposition 6.2: *Let F be an assumption of the BAN analysis for NS. For all runs r in $\mathcal{I}^{GDY}$ such that $\mathtt{n}_\mathtt{a} \mapsto n_a$ and $\mathtt{k}_\mathtt{a}^{-1} \mapsto k_a^{-1}$ are in $r_A(0)$, $\mathtt{n}_\mathtt{b} \mapsto n_b$ and $\mathtt{k}_\mathtt{b}^{-1} \mapsto k_b^{-1}$ are in $r_B(0)$, $n_a \neq n_b$ and $k_a \neq k_b^{-1}$, we have $(\mathcal{I}^{GDY}, r, 0) \models F^{P,\alpha}$, where $\alpha = (|\mathcal{N}| - 1)/|\mathcal{N}|$, and freshness is interpreted with respect to a time interval of $t \geq 1$.*

It is clear that assumptions are meant to hold at the initial state of the system, before the protocol executes. What about the conclusions? There are really two competing intuitions. The first is that these beliefs should hold at the points where the protocol has finished executing. This is the interpretation we took in Section 4.2. For example, at the end of the protocol, principal A should believe that n_a is a secret between A and B. The second intuition is that these beliefs are initial beliefs about *what will hold when the protocol terminates*. Under this interpretation, the BAN conclusion A **believes** $A \overset{n_a}{\rightleftharpoons} B$ means that principal A initially believes (before the protocol executes) that once the protocol terminates, n_a is a secret between A and B. It turns out that if we analyze NS against a Dolev-Yao adversary, when we can interpret BAN belief as knowledge, then these two interpretations coincide, which is why we did not raise this distinction in Section 4.2. However, if we interpret belief probabilistically, these interpretations differ dramatically.

The following example should illustrate the difference. Alice uses a protocol to communicate with Bob. Assume that there is very little chance for Trudy, the attacker, to guess the messages used by Alice and Bob during their exchange. Furthermore, assume that whenever a guess made by Trudy is wrong, Alice and Bob notice it immediately. Hence, before the protocol begins, Alice and Bob believe that with high probability, they will be able to communicate successfully (or catch Trudy in the act), since the chance of Trudy guessing successfully is very low. This is an initial belief about the outcome of the protocol. Now, suppose that Alice and Bob know *without any uncertainty* that Trudy is attacking. The initial belief still hold: with high probability, either they will communicate securely or they will detect Trudy. What happens if they manage to communicate without detecting Trudy? It is not the case that they believe with high probability that everything went fine. Quite the contrary! They know that Trudy was attacking, and if they did not detect her, that *must* mean that Trudy's attack was successful. In other words, they *know* that the protocol failed. Thus, the initial beliefs and the final beliefs can be quite different in a probabilistic setting.

As we shall see, the appropriate interpretation of the BAN conclusions—at least in a context when the adversary is allowed guesses—is best captured by treating these beliefs as initial beliefs in the results of the protocol. This interpretation is in fact quite common in the probabilistic verification literature (de Alfaro 1998) and in the definition of zero-knowledge protocols (Halpern, Moses, and Tuttle 1988). On the other hand, it is debatable whether specifications for *security protocols* should really be of the form of initial beliefs about conclusions. As our discussion above indicates, even strong initial beliefs about conclusions cannot avoid the situation where the principals involved reach a point where they are sure that something bad has happened. The stronger guarantees given by final beliefs would seem preferable in a security setting.

Following this discussion, we need to interpret a conclusion such as A **believes** F as $K_A(\Box(end \Rightarrow \varphi))$ where φ is the translation of F and end identifies the end of the protocol. This complicates the statement of the next proposition. Given a BAN conclusion F of the form i_1 **believes** $\ldots i_k$ **believes** G, where G is not a belief formula, let the *believers* of F be the sequence of agents $i_1, \ldots, i_k$, and the *body* of F be the formula G. With this terminology, we can state the following:

Proposition 6.3: *Let F be a conclusion of the BAN analysis for NS. For all runs r in $\mathcal{I}^{GDY}$ such that $\mathtt{n_a} \mapsto n_a$ and $\mathtt{k_a^{-1}} \mapsto k_a^{-1}$ are in $r_A(0)$, $\mathtt{n_b} \mapsto n_b$ and $\mathtt{k_b^{-1}} \mapsto k_b^{-1}$ are in $r_B(0)$, $n_a \neq n_b$, and $k_a \neq k_b^{-1}$, we have $(\mathcal{I}^{GDY}, r, 0) \models K_{i_1}^{\alpha} \ldots K_{i_k}^{\alpha}(\Box(\mathsf{recv}_B(\{n_b\}_{k_b}) \Rightarrow G^{P,\alpha}))$, where $i_1, \ldots, i_k$ are the believers of F, G is the body of F, $\alpha = (|\mathcal{N}| - 1)/|\mathcal{N}|$.*

The results of this section were established directly, by checking that the appropriate formulas hold at appropriate points of the models. In particular, we have not tried to derive the conclusions deductively from the assumptions. Note that for the probabilistic interpretation of belief, Theorem 6.1 guarantees the validity of the BAN inference rules only when belief is interpreted as knowing with probability 1, which would not allow us to directly derive the results in Proposition 6.3. This indicates that it might be difficult to formally account for the Hoare-style reasoning advocated by BAN in the presence of guessing adversaries.

7 Conclusion

We have introduced in this paper a simple multimodal logic to reason about security protocols, based on well-understood modal operators for knowledge, probability, and time. We have shown how those primitive notions can be used to capture the more high-level operators of BAN logic, helping us to understand the intuitions underlying BAN. More specifically, we have shown how assumptions about the adversary, whether it can perform guesses or not, affect the interpretation of the BAN operators.

A question comes out of our analysis, specifically from the fact that the goals of the BAN style analysis hold both in the nonprobabilistic (Propositions 4.1 and 4.2) and probabilistic case (Propositions 6.2 and 6.3). Are there conditions under which it is possible to "reduce" the analysis of a protocol under probabilistic assumptions to the nonprobabilistic assumptions. Why we would want such a result is clear: while the probabilistic models are more accurate and hence more realistic, the nonprobabilistic models are much simpler to generate and analyze than the probabilistic models. Such an investigation is in the spirit of the work of Abadi and Rogaway (2000), who show that under certain assumptions, formal reasoning about security protocols under a Dolev-Yao model of adversary is sound with respect to actual cryptosystems characterized by complexity-theoretic assumptions.

Our investigation of the interpretation of BAN belief in terms of probabilities revealed an interesting subtlety in the specification of protocols using belief, in a context where an adversary can perform guesses. There is a difference, when belief is viewed probabilistically, between an agent believing a fact φ at the end of a protocol run, and the agent initially believing that once a protocol ends φ holds. Our analysis highlights that the latter interpretation is the appropriate one for the beliefs making up BAN conclusions, in the presence of a guessing adversary. The implications of this observation for security protocol analysis in general, beyond the interpretation of BAN, is a topic of future work.

In our formalization of BAN via translation, we have only accounted for the actual BAN formulas. We have not attempted to account for the protocol analysis process advocated by BAN and its successors. This process involves reasoning deductively about the formulas that are true after every message sent during the protocol. The protocol analysis process is intimately connected to the semantics of the logic (although, interestingly, this connection is never completely formalized in the literature). For instance, the process relies on formulas being *stable*, that is, once true, they remain true. This dramatically restricts the formulas that can be used in the analysis. It may be possible to derive a similar, Hoare-style protocol analysis process for our logic, despite this lack of stability, and we are exploring this issue.

Our ultimate goal is to design a logic that will be useful for reasoning about all aspects of security, based on the logic we introduced here. In this paper, we have focused on high-level issues concerning the expressibility of high-level operators using well-understood primitive concepts. Other aspects of reasoning about security protocols that we have hinted at need to be further investigated. One such aspect is how to deal with quantitative belief. Ultimately, the most we can hope to prove is that, with high probability, a protocol is secure. Being able talk about this kind of probabilistic security, both qualitatively and quantitatively, is important. Another aspect is how to deal with the issue of what an adversary can compute; in this paper, we have sidestepped the problem using the proposition `extract`. We are investigating more general approaches to deal with this issue.

A BAN inference rules

The following inferences rules are used in BAN to derive new formulas from others.

R1. $$\frac{i \textbf{ believes } j \overset{k}{\leftrightarrow} i \quad i \textbf{ sees } \{F^l\}_k \quad l \neq i}{i \textbf{ believes } j \textbf{ said } F}$$

R2. $$\frac{i \textbf{ believes } j \overset{k}{\leftrightarrow} i \quad i \textbf{ sees } \{F^l\}_{k^{-1}} \quad l \neq i}{i \textbf{ believes } j \textbf{ said } F}$$

R3. $$\frac{i \textbf{ believes fresh}(F) \quad i \textbf{ believes } (j \textbf{ said } F)}{i \textbf{ believes } j \textbf{ believes } F}$$

R4. $$\frac{i \textbf{ believes } j \textbf{ controls } F \quad i \textbf{ believes } j \textbf{ believes } F}{i \textbf{ believes } F}$$

R5. $$\frac{i \textbf{ sees } F \cdot F'}{i \textbf{ sees } F}$$

R6. $$\frac{i \textbf{ believes } j \overset{k}{\leftrightarrow} i \quad i \textbf{ sees } \{F\}_k}{i \textbf{ sees } F}$$

R7. $$\frac{i \textbf{ believes } \overset{k}{\mapsto} i \quad i \textbf{ sees } \{F\}_k}{i \textbf{ sees } F}$$

R8. $$\frac{i \textbf{ believes } \overset{k}{\mapsto} j \quad i \textbf{ sees } \{F\}_{k^{-1}}}{i \textbf{ sees } F}$$

R9. $$\frac{i \textbf{ believes fresh}(F)}{i \textbf{ believes fresh}(F \cdot F')}$$

B Generating multiagent systems from protocols

In this section, we give a formal description of how we generate multiagent systems from protocols. The approach is derived from (Fagin, Halpern, Moses, and Vardi 1995, Chapter 5).

Intuitively, a protocol for agent i is a description of what actions i may take as a function of her local state. Actions, for our purposes, will consists of sending messages (although we could incorporate other actions, such as choosing keys, or tossing coins to randomize protocols).

We can formalize this intuition as follows. Fix a set L_i of local states for agent i (the local states that arise in some system) and a set A_i of possible actions that agent i can take. For nonadversary agents, we let A_i consists of the actions $\mathsf{no\text{-}op}$ (the null action) and $\mathsf{send}(j, \mathsf{m})$ for agent j and message m. For adversary agents, we let A_i consists of these same actions, along with the actions $\mathtt{intercept}(i)$, for agent i. Intuitively, these last actions indicate that the adversary intercepts any message sent by agent i at that time.

A *protocol* P_i for agent i is a function that associates with every local state in L_i a nonempty subset of actions in A_i. Intuitively, $P_i(s)$ is the set of actions that agent i may perform in local state s. Notice that i's actions can depend only on her local state. If P_i prescribes a unique action for i at each local state, then P_i is said to be *deterministic*.

Typically, each honest principal runs its prescribed protocols, while adversaries run one of a set of possible adversary protocols. The adversary protocols may involve guessing keys and nonces, or attempts to launch attacks by forwarding messages. To consider the effect of all the agents' protocols on the global state of the system, we define a *joint protocol* $(P_e, P_1, \ldots, P_n)$, consisting of a protocol for each of the agents, that associates with each global state a subset of possible *joint actions*, a subset of $A \subseteq A_e \times A_1 \times \cdots \times A_n$.

Joint actions transform global states. To capture their effect, associate with every joint action $(\mathbf{a}_1, \ldots, \mathbf{a}_n)$ a function from global states to global states. Formally, we let $\tau(\mathbf{a}_1, \ldots, \mathbf{a}_n)$ be a function taking a global state $(s_1, \ldots, s_n)$ to $(s'_1, \ldots, s'_n)$, where

$$s'_j = \begin{cases} s_j \cdot \mathsf{send}(i, \mathsf{m}) & \text{if } \mathbf{a}_j = \mathsf{send}(i, \mathsf{m}) \\ s_j \cdot \mathsf{recv}(\mathsf{m}) & \text{if } \mathbf{a}_i = \mathsf{send}(i', \mathsf{m}) \\ & \text{and } \mathbf{a}_j = \mathsf{intercept}(i) \text{ for some } j \\ s_j \cdot \mathsf{recv}(\mathsf{m}) & \text{if } \mathbf{a}_i = \mathsf{send}(j, \mathsf{m}) \\ & \text{and } \mathbf{a}_{i'} \neq \mathsf{intercept}(i) \text{ for all } i' \\ s_j & \text{otherwise.} \end{cases}$$

We use $\cdot$ to denote adding an element to the sequence of events making up a local state.

Given a joint protocol (and an associated interpretation τ for joint actions) and a set of initial global states, we can generate a system in a straightforward way. Intuitively, the system consists of all the runs that are obtained by running the joint protocol from one of the initial global states. More formally, say that run r is *consistent with joint protocol* P if it could have been generated by P, that is, for all m, $r(m+1)$ is the result of applying a joint action $\mathbf{a}$ that could have been performing according to protocol P to $r(m)$. (More precisely, there exists a joint action $\mathbf{a} = (\mathbf{a}_e, \mathbf{a}_1, \ldots, \mathbf{a}_n)$ such that $\mathbf{a}_i \in P_i(r_i(m))$ and $r(m+1) = \mathbf{a}(r(m))$.) Given a set I of global states, and a set $\mathcal{P}$ of joint protocols, the system $\mathcal{R}(\mathcal{P}, I)$ consists of all the runs consistent with some joint protocol in $\mathcal{P}$ that start at some initial global state in I.

Given protocols for the honest principals and a set of possible protocols for the adversaries, let $\mathcal{P}$ consist of all joint protocols where the honest principals follow their prescribed protocols and the adversaries use some allowed protocol. (We will not be concerned in this paper with the environment protocol; we will basically take the environment to be inactive.) Given an appropriate set I of initial states, we can then construct the system $\mathcal{R}(\mathcal{P}, I)$. We remark that, typically, finding an appropriate set I of initial states is easy; however, characterizing an appropriate set of possible protocols for the adversaries can be quite nontrivial.

References

Abadi, M. and P. Rogaway (2000). Reconciling two views of cryptography (the computational soundness of formal encryption). In *Proceedings of the IFIP International Conference on Theoretical Computer Science (IFIP TCS2000)*, Volume 1872 of *Lecture Notes in Computer Science*, pp. 3–22. Springer-Verlag.

Abadi, M. and M. R. Tuttle (1991). A semantics for a logic of authentication. In *Proc. 10th ACM Symp. on Principles of Distributed Computing*, pp. 201–216.

Aumann, R. J. (1999). Interactive epistemology I: knowledge. *International Journal of Game Theory 28*(3), 263–301.

Boyd, C. and W. Mao (1993). On a limitation of BAN logic. In *EUROCRYPT '93*, Volume 765 of *Lecture Notes in Computer Science*, pp. 240–247. Springer-Verlag.

Brandenburger, A. (1989). The role of common knowledge assumptions in game theory. In F. Hahn (Ed.), *The Economics of Information, Games, and Missing Markets*. Oxford, U.K.: Oxford University Press.

Burrows, M., M. Abadi, and R. Needham (1990a). A logic of authentication. *ACM Transactions on Computer Systems 8*(1), 18–36.

Burrows, M., M. Abadi, and R. M. Needham (1990b). Rejoinder to Nessett. *ACM Operating Systems Review 24*(2), 39–40.

de Alfaro, L. (1998). *Formal Verification of Probabilistic Systems*. Ph. D. thesis, Stanford University. Available as Technical Report STAN-CS-TR-98-1601.

Dolev, D. and A. C. Yao (1983). On the security of public key protocols. *IEEE Transactions on Information Theory 29*(2), 198–208.

Dwork, C. and Y. Moses (1990). Knowledge and common knowledge in a Byzantine environment: crash failures. *Information and Computation 88*(2), 156–186.

Fagin, R., J. Y. Halpern, Y. Moses, and M. Y. Vardi (1995). *Reasoning about Knowledge*. The MIT Press.

Gong, L., R. Needham, and R. Yahalom (1990). Reasoning about belief in cryptographic protocols. In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pp. 234–248. IEEE Computer Society Press.

Halpern, J. Y. and Y. Moses (1998). Using counterfactuals in knowledge-based programming. In *Theoretical Aspects of Rationality and Knowledge: Proc. Seventh Conference*, pp. 97–110. San Francisco, Calif.: Morgan Kaufmann.

Halpern, J. Y., Y. Moses, and M. R. Tuttle (1988). A knowledge-based analysis of zero knowledge. In *Proc. 20th ACM Symp. on Theory of Computing*, pp. 132–147.

Halpern, J. Y. and R. Pucella (2002). Modeling adversaries in a logic for reasoning about security protocols. In *Proceedings of FASec'02*. Available as Royal Holloway Department of Computer Science Technical Report CSD-TR-02-13.

Halpern, J. Y. and M. R. Tuttle (1993). Knowledge, probability, and adversaries. *Journal of the ACM 40*(4), 917–962.

Kripke, S. (1963). A semantical analysis of modal logic I: normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 9*, 67–96.

Lowe, G. (1995). An attack on the Needham-Schroeder public-key authentication protocol. *Information Processing Letters 56*, 131–133.

Lowe, G. (1998). Casper: A compiler for the analysis of security protocols. *Journal of Computer Security 6*, 53–84.

Lowe, G. (2002). Analysing protocols subject to guessing attacks. In *Proceedings of the Workshop on Issues in the Theory of Security (WITS'02)*.

Mitchell, J., M. Mitchell, and U. Stern (1997). Automated analysis of cryptographic protocols using Murφ. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pp. 141–151. IEEE Computer Society Press.

Moses, Y. and M. R. Tuttle (1988). Programming simultaneous actions using common knowledge. *Algorithmica 3*, 121–169.

Needham, R. M. and M. D. Schroeder (1978). Using encryption for authentication in large networks of computers. *Communications of the ACM 21*(12), 993–999.

Nessett, D. M. (1990). A critique of the Burrows, Abadi and Needham logic. *ACM Operating Systems Review 24*(2), 35–38.

Paulson, L. C. (1998). The inductive approach to verifying cryptographic protocols. *Journal of Computer Security 6*(1/2), 85–128.

Pnueli, A. (1977). The temporal logic of programs. In *Proc. 18th IEEE Symp. on Foundations of Computer Science*, pp. 46–57.

Stubblebine, S. and R. Wright (1996). An authentication logic supporting synchronization, revocation, and recency. In *Proc. Third ACM Conference on Computer and Communications Security*, pp. 95–105.

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

Thayer, F. J., J. C. Herzog, and J. D. Guttman (1999). Strand spaces: Proving security protocols correct. *Journal of Computer Security 7*(2/3), 191–230.

van Oorschot, P. C. (1993a). An alternate explanation of two BAN logic 'failures'. In *EUROCRYPT '93*, Volume 765 of *Lecture Notes in Computer Science*, pp. 443–447. Springer-Verlag.

van Oorschot, P. C. (1993b). Extending cryptographic logics of belief to key agreement protocols. In *Proceedings of the First ACM Conference on Computer and Communications Security*, pp. 232–243. ACM Press.

Wedel, G. and V. Kessler (1996). Formal semantics for authentication logics. In *Proceedings of 4th European Symposium on Research in Computer Security (ESORICS'96)*, Volume 1146 of *Lecture Notes in Computer Science*, pp. 219–241. Springer-Verlag.