Analysis of Authentication Protocols in Agent-Based Systems Using Labeled Tableaux

Ji Ma, Mehmet A. Orgun, *Senior Member, IEEE*, and Abdul Sattar

***Abstract*—The study of multiagent systems (MASs) focuses on systems in which many intelligent agents interact with each other using communication protocols. For example, an authentication protocol is used to verify and authorize agents acting on behalf of users to protect restricted data and information. After authentication, two agents should be entitled to believe that they are communicating with each other and not with intruders. For specifying and reasoning about the security properties of authentication protocols, many researchers have proposed the use of belief logics. Since authentication protocols are designed to operate in dynamic environments, it is important to model the evolution of authentication systems through time in a systematic way. We advocate the systematic combinations of logics of beliefs and time for modeling and reasoning about evolving agent beliefs in MASs. In particular, we use a temporal belief logic called TML$^+$ for establishing trust theories for authentication systems and also propose a labeled tableau system for this logic. To illustrate the capabilities of TML$^+$, we present trust theories for several well-known authentication protocols, namely, the Lowe modified wide-mouthed frog protocol, the amended Needham–Schroeder symmetric key protocol, and Kerberos. We also show how to verify certain security properties of those protocols. With the logic TML$^+$ and its associated modal tableaux, we are able to reason about and verify authentication systems operating in dynamic environments.**

***Index Terms*—Agent-based systems, authentication protocols, labeled tableaux, temporal belief logic, trust theory.**

I. Introduction

A. Motivation

THE study of multiagent systems (MASs) focuses on systems in which many intelligent agents interact with each other using communication protocols. For example, an authentication protocol provides a set of standard rules specifying the data representation, signaling, authentication, and error detection required to send messages over a communication channel. After authentication, two agents should be entitled to believe that they are communicating with each other and not with intruders. Therefore, it is important to precisely express such beliefs and to capture the reasoning that leads to them [8].

Trust is a critical issue for authentication protocols. Trust influences not only the specification of authentication protocols but also the techniques needed to implement and manage such protocols [30]. In a trust model proposed in [26], it is assumed that for security considerations, agents may initially not trust anyone but the security mechanisms (as special agents) of a system. The initial trust or metabeliefs of agents in the system can be represented as a set of rules (axioms) in a chosen logical framework. These rules together with those of the logic form a theory of trust for the system [24]. Such a theory provides a foundation for reasoning about agent beliefs, as well as security properties that the system may satisfy.

Since authentication protocols operate in dynamic environments, it is important to study the evolution of systems through time. In this paper, we propose a formal method to establish trust theories for communication protocols using a temporal belief logic called TML$^+$ and propose a labeled tableau system for TML$^+$; with its temporal dimension, the labeled tableaux can verify both the static and dynamic aspects of systems. Our approach can be used to model, reason about, and verify agent beliefs that may vary through time.

B. Related Works

A basic problem regarding the security properties of an authentication system is whether a message received through the system is reliable. The problem generally depends on the security mechanisms of a system and the trust that agents would put in the mechanisms. Reasoning about trust involves specifying and reasoning about agent beliefs. Modal logics can naturally be used to express necessity, possibility, belief, knowledge, temporal, and other modalities [18]. There are also many studies discussing how to use a kind of belief logic to describe communication systems [2], [8], [15], [38], [43].

Reference [8] proposed a logic called BAN to describe agent beliefs and the evolution of those beliefs. Many researchers have developed extensions of the BAN logic to overcome its deficiencies by generalizing its assumptions, such as the GNY logic [15], the AT logic [2], the VO logic [38], and the SVO Logic [43]. The GNY belief logic extends the BAN logic to require fewer assumptions and offers a few extra features. The SVO logic corrects flaws found in the BAN logic and other similar logics; it is shown that the SVO logic is easier to use and more expressive than the BAN logic. All those works are attempted to generate concise proofs and find flaws in many protocols.

Manuscript received December 8, 2007; revised April 4, 2008. First published May 5, 2009; current version published July 17, 2009. This work was supported in part by the Australian Research Council under Discovery Grant DP0452628 and in part by Macquarie University under a Research Development Grant. This paper was recommended by Associate Editor Y. Wang.

J. Ma is with the Department of Computing, Macquarie University, Sydney, NSW 2109, Australia, and also with Griffith University, Brisbane 4111, Australia (e-mail: jma@comp.mq.edu.au).

M. A. Orgun is with the Department of Computing, Macquarie University, Sydney, NSW 2109, Australia (e-mail: mehmet@comp.mq.edu.au).

A. Sattar is with the Institute for Integrated and Intelligent Systems, Griffith University, Brisbane 4111, Australia (e-mail: a.sattar@griffith.edu.au).

Digital Object Identifier 10.1109/TSMCB.2009.2019263

Some other belief logics include the following: the logic in [35] is used for reasoning about beliefs of protocol participants and trust changes, the Yahalom *et al.* system [46] is used for analyzing the trust that protocol participants must have in each other, the logic in [19] is used for analyzing the accountability of communication protocols, the logic in [21] can be used for analyzing the accountability of electronic commerce protocols, and the logic in [45] is used for analyzing the soundness of authentication protocols. Based on these logics (and many others like them), system-specific theories of trust can be constructed, and hence, they provide a basis for reasoning about trust in particular environments and systems.

Furthermore, belief logics are primarily suitable for expressing static properties, for example, the assertion "Alice believes that Bob is honest" can be formalized (adopting the modal logic notation) as $\mathbf{B}_{Alice}$ IsHonest (*Bob*), where $\mathbf{B}_{Alice}$ is the modal belief operator for agent *Alice*. However, trust dynamically changes. Therefore, there is a pressing need for building a logical framework, in which we are able to describe both the "statics" and "dynamics" of trust and agent beliefs. This leads us to logics suitable for representing dynamic properties. Temporal logics have the ability to deal with dynamic properties. Several researchers have recently proposed approaches based on temporal epistemic logics for analyzing security protocols [9], [29].

In parallel to the development of temporal belief logics, it has been argued that any logical system used for modeling active agents should be a combined system of logics of knowledge, belief, time, and norms [14] since these are among the essential concepts to be reasoned about. There have been several methods and techniques proposed for combining logics, such as fibring, products, and temporalization [13]. Temporalization [11] is a methodology for combining logics whereby an arbitrary logic system can be enriched with temporal features to generate a new logic system. The resulting logic is a kind of temporal belief logic, and such logics are particularly suitable for specifying and reasoning about the dynamics of trust for communication systems.

Formal methods have been used to verify the correctness of communication protocols. Many researchers have proposed and developed general-purpose verification tools for modeling and verifying authentication protocols, such as Petri nets [5], [10], [23], finite-state machines [3], and model checkers [6], [13], [42]. Some researchers have developed special-purpose expert systems [20], [28], [41], [44], [47]. Those expert systems have been developed specifically for the analysis of agent systems. Therefore, they are more successful in their task than general-purpose tools. Some researchers have developed algebraic methods [1], [7], [40] for modeling protocols with a collection of rules for transforming and reducing algebraic expressions representing messages.

C. Aims and Contributions

We advocate the systematic combinations of logics of belief, time, and other notions for specifying and verifying trust theories for communication protocols. In particular, we use a temporal belief logic called TML$^+$ [27], which is combined by a belief logic with a temporal logic using the temporalizing technique proposed by Finger and Gabbay [11]. This technique allows a hierarchy integration of two logics in a systematic way. TML$^+$ can be applied for reasoning about time-dependent properties regarding trust and agent beliefs in communication protocols. In this paper, we propose a labeled tableau proof system for the temporal belief logic TML$^+$. With the logic and its associated modal tableau, we are able to specify, reason about, and verify communication systems operating in dynamic environments.

Focusing on the dynamics of trust, we first show how to establish theories of trust for authentication systems in TML$^+$ and then discuss how to verify the security properties of those theories once they have been established.

The main contributions of this paper are listed as follows.

1) It provides a formal approach for establishing theories of trust for authentication protocols, which can be used to express agent beliefs within authentication systems. Such trust theories also provide a foundation for reasoning about evolving agent beliefs.
2) It provides a labeled tableau inference system for TML$^+$, which can be used to analyze the correctness of MASs.
3) It provides several tableau inference techniques, which include label unification and tableau closure, and demonstrates the advantages of the labeled tableau inference system and its associated inference techniques for verifying the security properties of an authentication protocol for MASs.

The rest of this paper is organized as follows. Section II introduces the temporal belief logic TML$^+$. Section III proposes a labeled tableau system for TML$^+$. Section IV proposes a generic method for establishing dynamic trust theories for authentication protocols and gives three examples: the Lowe modified wide-mouthed frog protocol, the amended Needham–Schroeder symmetric key protocol, and Kerberos. Furthermore, we show how to verify those theories using the labeled tableau system. Section V concludes the paper with a brief summary.

II. Temporal Belief Logic TML$^+$

For reasoning about agent beliefs, [24] proposed the typed modal logic (TML), which extends a first-order logic with typed variables and belief operators. TML has a strong expressive power for describing static properties. However, without the introduction of a temporal dimension, TML may not be able to express dynamic agent beliefs. In a later work, [27] used the temporalizing technique proposed in [11] to add a temporal dimension to TML by combining it with a discrete-time temporal logic called simple linear-time temporal logic (SLTL) [25]. The resulting logic, i.e., TML$^+$, can be applied for reasoning about time-dependent properties regarding agent beliefs in secure systems.

A. Syntax and Semantics of TML$^+$

The temporalized belief logic TML$^+$ has two classes of modal operators: 1) belief operators and 2) temporal operators.

TABLE I
INTERPRETATION OF TEMPORAL OPERATORS (A IS A FORMULA; T REPRESENTS VALUE **true**, AND F REPRESENTS VALUE **false**)

Formula	Truth value								
A	T	T	F	F	T	F	T	F	...
first A	T	T	T	T	T	T	T	T	...
next A	T	F	F	T	F	T	F	...	...
Time	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	...

The belief operators are those of TML, whereas the temporal operators are those of SLTL. Each agent has an associated belief operator, for instance, the belief operator $\mathbf{B}_{Bob}$ is intended to denote "Bob believes that." The temporal operators contained in TML$^+$ are $\mathbf{first}$ and $\mathbf{next}$, which refer to the initial moment and the next moment in time, respectively. Classical temporal operators such as *always* and *sometimes* can also be introduced into SLTL using recursive definitions; however, we do not consider them in this paper. Since the logic TML$^+$ is obtained by using temporalization, which is a hierarchy combination technique for combining logics, the temporal operators $\mathbf{first}$ and $\mathbf{next}$ can never appear within the scope of a belief operator $\mathbf{B}_i$. Hence, in TML$^+$, we can only express statements about the temporal aspects of agent beliefs, such as "At the initial moment in time, Alice believes that Bob is honest": $\mathbf{first}\ \mathbf{B}_{Alice}$ IsHonest(*Bob*).

SLTL is a (first-order) discrete linear-time logic where the underlying collection of time points is the set of natural numbers with its usual ordering relation $<$. A time model for the logic SLTL has the form $\mathcal{M}_{\mathrm{sltl}} = \langle ck, <, \pi \rangle$, where $ck = \langle t_0, t_1, t_2, \ldots \rangle$ is a clock, $<$ is the usual ordering relation over ck, and π is an assignment function that gives a value $\pi(t, p) \in \{true, false\}$ for any time point t in ck and any atomic formula p. The assignment function π interprets all the terms given in atomic formula p in a standard way. In the following, we only give the semantics of the temporal operators of SLTL.

1) $\mathcal{M}_{\mathrm{sltl}}, t_i \models \mathbf{first}\ \varphi$ iff $\mathcal{M}_{\mathrm{sltl}}, t_0 \models \phi$.
2) $\mathcal{M}_{\mathrm{sltl}}, t_i \models \mathbf{next}\ \varphi$ iff $\mathcal{M}_{\mathrm{sltl}}, t_{i+1} \models \phi$.

Table I gives an explanation of the interpretation of the temporal operators of SLTL.

TML is a variant of the modal logic $\mathbf{KD}$ of beliefs [17]. A *classical Kripke model* [22] for TML is defined as a tuple $\mathcal{M}_{\mathrm{tml}} = \langle S, R_1, \ldots, R_n, \pi \rangle$, where S is the set of states or possible worlds, and each R_i, $i = 1, \ldots, n$, is a serial relation over S (called the *accessibility relation* according to agent i). R_i is a nonempty set consisting of state pairs (s, t) such that $(s, t) \in R_i$ iff at state s, agent i considers the state t to be accessible, and π is the *assignment function*, which gives a value $\pi(s, p) \in \{true, false\}$ for any $s \in S$ and atomic formula p. Again, the assignment function π interprets all the terms given in atomic formula p in a standard way. Formula ϕ is satisfiable in the model $\mathcal{M}_{\mathrm{tml}}$ if there exists $s \in S$ such that $\mathcal{M}_{\mathrm{tml}}, s \models \phi$. The domain objects (values) are classified into types, and each term can only have values of a certain type. In the following, we give the semantics of the belief operators of TML.

- $\mathcal{M}_{\mathrm{tml}}, s \models \mathbf{B}_i\ \phi$ iff for all t, $(s, t) \in R_i$, $\mathcal{M}_{\mathrm{tml}}, t \models \phi$.

For combining TML and SLTL, we first stipulate that SLTL and TML share the same vocabulary (predicates, function symbols, and variables).

Let $\mathcal{L}_{\mathrm{tml}}$ represent the set of all formulas in TML defined as follows.

1) $\phi \in \mathcal{L}_{\mathrm{tml}}$ if ϕ is an atomic formula of TML.
2) If $\phi \in \mathcal{L}_{\mathrm{tml}}$ and $\psi \in \mathcal{L}_{\mathrm{tml}}$, then $\neg\phi \in \mathcal{L}_{\mathrm{tml}}$, and $(\phi \wedge \psi) \in \mathcal{L}_{\mathrm{tml}}$.
3) $\mathbf{B}_i\phi \in \mathcal{L}_{\mathrm{tml}}$, for all $i (1 \leq i \leq n)$, if $\phi \in \mathcal{L}_{\mathrm{tml}}$.

Let $\mathcal{L}_{\mathrm{tml}^+}$ be the set of formulas of TML$^+$. Then, $\mathcal{L}_{\mathrm{tml}^+}$ is defined as follows.

1) If $\phi \in \mathcal{L}_{\mathrm{tml}}$, then $\phi \in \mathcal{L}_{\mathrm{tml}^+}$.
2) If $\phi \in \mathcal{L}_{\mathrm{tml}^+}$ and $\psi \in \mathcal{L}_{\mathrm{tml}^+}$, then $\neg\phi \in \mathcal{L}_{\mathrm{tml}^+}$, and $(\phi \wedge \psi) \in \mathcal{L}_{\mathrm{tml}^+}$.
3) If $\phi \in \mathcal{L}_{\mathrm{tml}^+}$, then $\mathbf{first}\ \phi \in \mathcal{L}_{\mathrm{tml}^+}$ and $\mathbf{next}\ \phi \in \mathcal{L}_{\mathrm{tml}^+}$.
4) If $\phi(X) \in \mathcal{L}_{\mathrm{tml}^+}$ and X is any variable, then $(\forall X)\phi(X) \in \mathcal{L}_{\mathrm{tml}^+}$.

To be able to interpret a formula of TML$^+$ whose main operator is a temporal operator, we need to use the meaning of the temporal operators with a time reference. To be able to interpret a formula whose main operator is a belief operator, similarly, we need to use the meaning of the belief operators with a state reference. The temporalization method combines the semantics of the constituent logics using a mapping that associates each moment in time with a classical Kripke model in such a way that any formula of TML$^+$ is interpreted in its proper context. The meaning of a TML$^+$ formula initially involves a time model and a time reference; however, as soon as a belief operator is encountered, the current time reference would be mapped to the classical Kripke model under which the meaning of the subformula (to which the belief operator was applied) would be decided.

Let $\mathcal{K}^+$ be a class of Kripke models [22] of the logic TML of the form $\mathcal{M}_{\mathrm{tml}} = \langle S, R_1, \ldots, R_n, \pi, s \rangle$, where s is the current world, from which the observation is made [11]. Consider a time frame $(CK, <)$, where $CK = \langle t_0, t_1, t_2, \ldots \rangle$, and a function $v : CK \rightarrow \mathcal{K}^+$, mapping moments in time on the clock CK to a model in the class $\mathcal{K}^+$. We fix a global assignment α, which interprets all the terms of TML$^+$ (based on rigid designators over any constant domain $\mathbf{D}$). A model of TML$^+$ is then a quadruple $\langle CK, <, v, \alpha \rangle$, denoted by $\mathcal{M}_{\mathrm{tml}^+}$. The semantics of TML$^+$ formulas are given as follows.

1) $\mathcal{M}_{\mathrm{tml}^+}, t_i \models \phi$, where $\phi \in \mathcal{L}_{\mathrm{tml}}$, iff $v(t_i) = \mathcal{M}_{\mathrm{tml}}$ and $\mathcal{M}_{\mathrm{tml}} \models \phi$, where the assignment function π in $\mathcal{M}_{\mathrm{tml}}$ agrees with the global assignment α in $\mathcal{M}_{\mathrm{tml}^+}$ on the values of all the terms.
2) $\mathcal{M}_{\mathrm{tml}^+}, t_i \models \neg\phi$ iff $\mathcal{M}_{\mathrm{tml}^+}, t_i \not\models \phi$.
3) $\mathcal{M}_{\mathrm{tml}^+}, t_i \models \phi \wedge \psi$ iff $\mathcal{M}_{\mathrm{tml}^+}, t_i \models \phi$ and $\mathcal{M}_{\mathrm{tml}^+}, t_i \models \phi$.
4) $\mathcal{M}_{\mathrm{tml}^+}, t_i \models \mathbf{first}\ \phi$ iff $\mathcal{M}_{\mathrm{tml}^+}, t_0 \models \phi$.
5) $\mathcal{M}_{\mathrm{tml}^+}, t_i \models \mathbf{next}\ \phi$ iff $\mathcal{M}_{\mathrm{tml}^+}, t_{i+1} \models \phi$.
6) $\mathcal{M}_{\mathrm{tml}^+}, t_i \models (\forall X)\phi(X)$ iff $\mathcal{M}_{\mathrm{tml}^+}[X/d], t_i \models \phi(X)$ for all $d \in \mathbf{D}$, where the notation $\mathcal{M}_{\mathrm{tml}^+}[X/d]$ refers to the model that may differ from the model $\mathcal{M}_{\mathrm{tml}^+}$ only in the assignment to the variable X.

We assume the standard definitions of logical connectives such as $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$. The semantics of quantifiers $\forall$ and $\exists$ are defined in the standard way with respect to TML models based on rigid domains for terms [12].

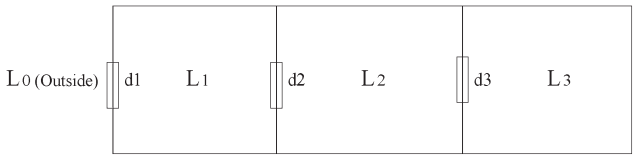

Fig. 1. Secured location of an organization.

B. Example Theory

In the following, we give an example adopted from our previous work [30] to demonstrate how TML$^+$ can be used to specify and reason about evolving agent beliefs.

Suppose that there is a secured location of an organization. For entering the location, anyone who is currently outside (location L_0) must first pass through door d_1 to get into location L_1, then pass through door d_2 to get into location L_2, and finally pass through door d_3 to get into location L_3, as shown in Fig. 1. Doors d_1, d_2, and d_3 are controlled by agents a_1, a_2, and a_3, respectively. The methods of authentication adopted for d_1, d_2, and d_3 are assumed to be m_1, m_2, and m_3, respectively. Each agent allows a person to pass through the door it controls only if the agent believes that the identity of the person is authenticated.

To establish a theory of trust for the system, we define three predicates as follows.

1) At (x, l): x is at the location l, where x represents a person ranging over the set of agents, and l represents a location ranging over $\{l_0, l_1, l_2, l_3\}$.
2) RequestsToEnter(x, l): x requests to enter the location l.
3) Authenticated(x, m): The identity of x is authenticated by m, where m represents the authentication method ranging over $\{m_1, m_2, m_3\}$.

We now define the following rules that describe the functional (behavioral) properties of the authentication system. Here, x is assumed to be universally quantified over agents.

S1) $\text{At}(x, l_{i-1}) \wedge \text{RequestsToEnter}(x, l_i) \rightarrow (\mathbf{next}\ \text{At}(x, l_i) \leftrightarrow \mathbf{B}_{a_i}\ \text{Authenticated}(x, m_i))$, for $i = 1, 2, 3$.
S2) $\text{At}(x, l_0) \rightarrow \mathbf{next}(\text{At}(x, l_1) \vee \text{At}(x, l_0))$.
S3) $\text{At}(x, l_1) \rightarrow \mathbf{next}(\text{At}(x, l_2) \vee \text{At}(x, l_1) \vee \text{At}(x, l_0))$.
S4) $\text{At}(x, l_2) \rightarrow \mathbf{next}(\text{At}(x, l_3) \vee \text{At}(x, l_2) \vee \text{At}(x, l_1))$.
S5) $\text{At}(x, l_3) \rightarrow \mathbf{next}(\text{At}(x, l_3) \vee \text{At}(x, l_2))$.
S6) $\text{At}(x, l_i) \wedge \mathbf{next}\ \text{At}(x, l_i) \rightarrow \mathbf{next}^2\ \text{At}(x, l_{i-1})$, for $i = 1, 2$.

Here, $\mathbf{next}^n$ denotes the n-fold application of $\mathbf{next}$. Rule S1 specifies the authentication procedure. It says that if currently, a person x is at location l_{i-1} and requests to enter the next location l_i, then person x is at location l_i at the next moment in time iff agent a_i believes the client is authenticated by authentication method m_i. Here, a_i represents an authentication agent ranging over $\{a_1, a_2, a_3\}$, and m_i represents an authentication method ranging over $\{m_1, m_2, m_3\}$. Rules S2–S5 specify location restrictions, and rule S6 specifies the timing restriction.

III. Labeled Tableaux for TML$^+$

In this section, we present a labeled tableau system for TML$^+$, which can be used to automatically check the security properties of a system and reason about agent beliefs. Tableaux are refutation systems. Tableaux use a kind of tree structure; the formulas of the labeled tableau system we present are in the form $\phi : w$, where ϕ is a formula of the logic, and w is a label. The interpretation of $\phi : w$ is that ϕ is true at the world w. Various semantic tableau proof systems have been proposed in the literature [12], [16], [33], [39]; we adopt a variant of the free-variable tableau system [16], which suits combinations of logics very well. In this system, the labels can be either atomic labels (either constants or variables) or complex labels obtained by sequences of atomic labels.

The reasoning process is based on inference rules. The labeled tableau system for the logic TML$^+$ consists of the following rules.

The following are the conjunctive rules:

$$\frac{\phi \wedge \psi : w}{\begin{array}{c}\phi : w \\ \psi : w\end{array}} \quad \frac{\neg(\phi \vee \psi) : w}{\begin{array}{c}\neg\phi : w \\ \neg\psi : w\end{array}} \quad \frac{\neg(\phi \rightarrow \psi) : w}{\begin{array}{c}\phi : w \\ \neg\psi : w\end{array}} \quad \frac{\phi \leftrightarrow \psi : w}{\begin{array}{c}\phi \rightarrow \psi : w \\ \psi \rightarrow \phi : w\end{array}}.$$

These conjunctive rules are linear branch expansion rules.

The following are disjunctive rules:

$$\frac{\phi \vee \psi : w}{\phi : w \mid \psi : w} \quad \frac{\neg(\phi \wedge \psi) : w}{\neg\phi : w \mid \neg\psi : w}$$

$$\frac{\phi \rightarrow \psi : w}{\neg\phi : w \mid \psi : w} \quad \frac{\neg(\phi \leftrightarrow \psi) : w}{\neg(\phi \rightarrow \psi) : w \mid \neg(\psi \rightarrow \phi) : w}.$$

These disjunctive rules generate new branches.

The following are the quantifier rules:

$$\frac{\forall x \Phi(x) : w}{\Phi(c) : w} \quad \frac{\neg\exists x \Phi(x) : w}{\neg\Phi(c) : w}$$

$$\frac{\exists x \Phi(x) : w}{\Phi(c) : w} \quad \frac{\neg\forall x \Phi(x) : w}{\neg\Phi(c) : w}.$$

In both universal and existential rules, x is a variable, and c is a constant or any variable.

The following is the substitution rule:

$$\frac{\phi \leftrightarrow \psi : w_i \qquad \phi : w_j}{\psi : w_j}.$$

For the substitution rule, if $\phi \leftrightarrow \psi : w_i$ and $\phi : w_j$ both occur on a tableau branch, then $\psi : w_j$ can be added to the end of the branch.

In the following, we present the rules for modal operators. The temporal rules are given as follows:

$$\frac{\mathbf{first}\ \phi : w}{\phi : (t_0, w)} \quad \frac{\neg\,\mathbf{first}\ \phi : w}{\neg\phi : (t_0, w)} \quad \frac{\mathbf{next}\ \phi : w}{\phi : (t_i, w)}$$

$$\frac{\neg\mathbf{next}\ \phi : w}{\neg\phi : (t_i, w)} \quad \frac{\mathbf{first}\ \phi : (t_i, w)}{\phi : (t_0, w)} \quad \frac{\neg\mathbf{first}\ \phi : (t_i, w)}{\neg\phi : (t_0, w)}$$

$$\frac{\mathbf{next}\ \phi : (t_i, w)}{\phi : (t_{i+1}, (t_i, w))} \quad \frac{\neg\mathbf{next}\ \phi : (t_i, w)}{\neg\phi : (t_{i+1}, (t_i, w))}.$$

For temporal rules, we need to introduce the concept of a time-free world. A label (world) may or may not relate to time. If a world does not relate to time, i.e., it does not contain a time reference, we say that it is a time-free world such as w; otherwise, it is a time-related world such as (t_0, w), (t_i, w), and $(t_{i+1}, (t_i, w))$

The belief rules are given as follows:

$$\frac{\mathbf{B}_a\phi : w}{\phi : (w_i^a, w)} \quad \frac{\neg\mathbf{B}_a\phi : w}{\neg\phi : (w_i^a, w)} \quad \frac{\mathbf{B}_a\phi : (t_j, \ldots, (t_i, w)\ldots)}{\phi : (t_j, \ldots, (t_i, (w_i^a, w))\ldots)}$$

$$\frac{\neg\mathbf{B}_a\phi : (t_j, \ldots, (t_i, w)\ldots)}{\neg\phi : (t_j, \ldots, (t_i, (w_i^a, w))\ldots)}.$$

For belief rules, the notation (w_i^a, w) represents the world accessible via R_a from w, where R_a is the accessibility relation associated with agent a; (w_i^a, w) is also a time-free world, where w_i^a in the rules for $\mathbf{B}_a$ is a variable atomic label. In the rules for $\neg\mathbf{B}_a$, it is a constant atomic label not previously occurring in the proof.

The following unification rule states that two labeled formulas are σ_{tml^+}-complementary when the two formulas are complementary and their labels σ_{tml^+}-unify

$$\frac{\phi(x) : u \qquad \neg\phi(y) : v}{\times}.$$

Here, variables x and y unify as usual. $[u; v]\sigma_{\mathrm{tml}^+}$ holds iff u and v can be unified with the unifier σ_{tml^+}, i.e., $u\sigma_{\mathrm{tml}^+} = v\sigma_{\mathrm{tml}^+}$. We define $[u; v]\sigma_{\mathrm{tml}^+}$ as follows.

1) $[u; v]\sigma_{\mathrm{tml}^+} = u$ if $u = v$ in the case $\sigma_{\mathrm{tml}^+} = \{\}$.
2) $[u; v]\sigma_{\mathrm{tml}^+} = u$ if v is a variable in the case $\sigma_{\mathrm{tml}^+} = \{u/v\}$. Symmetrically, $[u; v]\sigma_{\mathrm{tml}^+} = v$ if u is a variable in the case $\sigma_{\mathrm{tml}^+} = \{v/u\}$.
3) $[u; v]\sigma_{\mathrm{tml}^+} = ([h(u); h(v)]\sigma^{(h)}_{\mathrm{tml}^+}, [t(u); t(v)]\sigma^{(t)}_{\mathrm{tml}^+})$ if $[h(u); h(v)]\sigma^{(h)}_{\mathrm{tml}^+}$ and $[t(u); t(v)]\sigma^{(t)}_{\mathrm{tml}^+}$ in the case $\sigma_{\mathrm{tml}^+} = \sigma^{(h)}_{\mathrm{tml}^+} \cup \sigma^{(t)}_{\mathrm{tml}^+}$, where h stands for head, t stands for tail, $u = (h(u), t(u))$, and $v = (h(v), t(v))$.
4) Otherwise, u and v cannot be unified.

Logical consequence is the relation that holds between a set of formulas and a formula; we write

$$\mathcal{S} \models_{\mathrm{TML}^+} \mathcal{U} \Rightarrow \phi$$

which denotes that formula ϕ is derived from $\mathcal{S}$ and $\mathcal{U}$, where $\mathcal{S}$ is the global assumption set, and $\mathcal{U}$ is the local assumption set. ϕ is a logical consequence that follows from $\mathcal{S}$ and $\mathcal{U}$.

In some stages of a reasoning process, one may need to introduce assumptions. The following assumption rules should be added into the tableau system:

$$\frac{\phi \in \mathcal{S}}{\phi : W} \quad \frac{\phi \in \mathcal{U}}{\phi : w_0}$$

where W is variable that represents any world, and w_0 is a specific world.

The tableau system is sound and complete for the logic presented in this paper. As usual with tableaux systems, a proof of ϕ is a closed tableau for $\neg\phi$. A tableau system is sound and complete for a particular logic if it is able to generate closed tableaux for the negation of a valid formula and open tableaux (models) for all satisfiable formulas. For more details on the soundness and completeness of a tableau system, we refer the reader to the literature [12], [16].

IV. Establishing Theories of Trust for Authentication Protocols

In this section, we propose a generic method for establishing trust theories and give three examples to show how to construct a trust theory for a given authentication protocol.

A. Generic Method for Establishing Trust Theories

A trust theory for a given authentication system consists of a set of rules that can be used for reasoning about agent beliefs and security properties that the system may satisfy [32]. Establishing a trust theory for a given authentication system involves the following steps:

1) analyzing how the communication system works;
2) analyzing the security mechanisms of the system and identifying the agents in it and the initial security assumptions;
3) defining appropriate predicates to express agent beliefs;
4) defining rules that describe the functions and behaviors of the system.

In this paper, we use the notations A, B, and S to denote agents, I denotes an intruder, X and Y denote messages, T_a and T_s denote specific timestamps, N_a and N_b denote specific nonces, and k_{ab}, k_{as}, and k_{bs} denote specific keys.

To establish a theory of trust for authentication systems, we define the following standard predicates in our vocabulary.

1) Send(a, b, msg): Agent a sends a message msg to another agent b.
2) Receive(a, msg): Agent a receives a message msg.
3) Secure(k): Key k is secure.
4) Fresh(t): Timestamp t is fresh.
5) Duplicated(t): Timestamp t is duplicated.
6) Reliable(msg): Message msg is reliable.
7) Reject(msg): Message msg is rejected.
8) Client(c): c is a registered client.
9) ServiceGranted(a, s): Agent a is granted a service s.

Now, we discuss the standard communication axioms for authentication systems. In the following axioms, all variables

are assumed to be universally quantified. The axioms are valid for any given pairs of agents a and b.

R1) $\mathbf{B}_a$ Secure$(k) \wedge$ Receive$(a, \{X\}k) \leftrightarrow$ $\mathbf{B}_a$ Reliable(X).
R2) $\mathbf{B}_a$ Secure$(k) \wedge$ Receive$(a, \{X[T]\}k) \wedge$ $\mathbf{B}_a$ Fresh$(T) \leftrightarrow \mathbf{B}_a$ Reliable$(X[T])$.
R3) Receive$(a, \{X[T]\}_k) \wedge \mathbf{B}_a$ Duplicated(T) $\leftrightarrow$ Reject$(X[T])$.
R4) Send $(a, b, X) \leftrightarrow$ Receive(b, X).
R5) Receive$(a, (X, Y)) \leftrightarrow$ Receive$(a, X) \wedge$ Receive(a, Y).
R6) $\mathbf{B}_a$ Fresh$((X, Y)) \leftrightarrow \mathbf{B}_a$ Fresh$(X) \wedge \mathbf{B}_a$ Fresh(Y).
R7) $\mathbf{B}_a$ Reliable$((X, Y)) \leftrightarrow$ $\mathbf{B}_a$ Reliable$(X) \wedge \mathbf{B}_a$ Reliable(Y).

Here, $\{X\}k$ means that message X is encrypted with key k. The meaning of these axioms are given as follows. Rule R1 says that if agent a believes that encryption key k is secure and receives a message X encrypted with key k, then agent a believes that the message X is reliable. Rule R2 says that if agent a believes that encryption key k is secure and receives a message X that contains a timestamp T (denoted by $X[T]$), the message is encrypted with key k, and agent a believes that the timestamp is fresh, then agent a believes that the message is reliable. Rule R3 says that if agent a receives a message X that contains a timestamp T and agent a believes that the timestamp is duplicated, then the message will be rejected. Rule R4 assumes that sending and receiving a message simultaneously happen. The meanings of Rules R5–R7 are straightforward.

The security properties of this protocol are based on the confidentiality of encryption keys and synchronized clocks.

1) Encryption keys: If an intruder does not know those encryption keys used to encrypt messages, in other words, if encryption keys are not compromised, then the intruder cannot learn those messages.
2) Synchronized clocks: Synchronized clocks are used to guarantee the freshness of timestamps.

We give the following protocol examples to show how to establish a specific trust theory for a given system.

B. Lowe Modified Wide-Mouthed Frog Protocol

The Lowe modified wide-mouthed frog protocol is a symmetric key management protocol involving a trusted third party. The protocol can be specified as follows in security protocol notation, where an agent A is authenticating itself to another agent B using a server S [4], [8], [29]:

1) A $\rightarrow$ S: $A, \{T_a, K_{ab}, B\}_{K_{as}}$.
2) S $\rightarrow$ B: $\{T_s, K_{ab}, A\}_{K_{bs}}$.
3) B $\rightarrow$ A: $\{N_b\}_{K_{ab}}$.
4) A $\rightarrow$ B: $\{N_b + 1\}_{K_{ab}}$.

The set of the security mechanisms of the system, denoted by $\mathcal{M}_w$, is defined as follows:

$$\mathcal{M}_w = \{k_{ab},\ k_{as},\ k_{bs}\}.$$

Then, we have the following security assumptions:

$$\begin{array}{ll} \mathbf{B}_a \text{ Secure}(k_{ab}) & \mathbf{B}_a \text{ Secure}(k_{as}) \\ \mathbf{B}_s \text{ Secure}(k_{as}) & \mathbf{B}_s \text{ Secure}(k_{bs}) \\ \mathbf{B}_b \text{ Secure}(k_{bs}) & \mathbf{B}_b \text{ Secure}(k_{ab}) \end{array}$$

These assumptions are about the security of encryption keys.

We have the following rules that describe the authentication procedure of the protocol:

W1) $\mathbf{B}_s$ Reliable$(\{T_a, K_{ab}, B\}_{K_{as}}) \leftrightarrow$ **next** Send$(S, B, \{T_s, K_{ab}, A\}_{K_{bs}})$.
W2) $\mathbf{B}_b$ Reliable$(\{T_s, K_{ab}, A\}_{K_{bs}}) \leftrightarrow$ **next** send$(B, A, \{N_b\}_{K_{ab}})$.
W3) $\mathbf{B}_a$ Reliable$(\{N_b\}_{K_{ab}}) \leftrightarrow$ **next** send$(A, B, \{N_b + 1\}_{K_{ab}})$.

These authentication rules are defined based the sequence of the communication protocol. Now, we have established a theory $\mathrm{T}_w = \{\mathrm{R1},\ \mathrm{R2},\ \mathrm{R3},\ \mathrm{R4},\ \mathrm{R5},\ \mathrm{R6},\ \mathrm{R7},\ \mathrm{W1},\ \mathrm{W2},\ \mathrm{W3}\}$ for the Lowe modified wide-mouthed frog protocol. In this theory, rules R1–R7 describe agent beliefs with authentication systems, and rules W1–W3 describe possible actions of agents based on their beliefs. Since agents trust the security mechanisms of the protocol, the theory is the foundation for reasoning about agent beliefs within the system.

We have the trust theory T and $\mathcal{S}$ as the global assumption set. To show that a security property (a formula) is valid in a theory, we need to prove it is a logical consequence of $\mathrm{T} \cup \mathcal{S}$. That is, in the general case, we need to prove that

$$\mathrm{T} \cup \mathcal{S} \models_{\mathrm{TML}^+} \mathcal{U} \Rightarrow \phi.$$

With the assumption that Send $(A, S, \{msg\}k_{as})$, we can prove $\mathbf{B}_s$ Reliable(msg). In other words, when an agent sends a message to server S, the server would believe that the message is reliable. That is

$$\mathrm{T}_w \cup \mathcal{S} \models_{\mathrm{TML}^+} \{\text{Send}\,(A, S, \{msg\}k_{as})\} \Rightarrow \mathbf{B}_s \text{ Reliable}(msg).$$

The proof is shown in Fig. 2.

For proving $\mathbf{B}_s$Reliable(msg), we begin with its negation $\neg\mathbf{B}_s$Reliable(msg), which is labeled by w_0 (a specific world). In the proof process, when we introduce a local assumption, we also label it by w_0; when we introduce a global assumption, we label it by a new variable such as W_1 and W_2.

When a branch reaches a node with $\times$, then the branch is closed. In this tableau, node 14 is obtained by nodes 9 and 1 with the unifier $\{W_2/w_0\}$. We apply the unification rule without applying expansion rules in suitable cases. A closure of a tableau immediately generates a closure for all branches that a node could have. This technique simplifies the proof procedure.

This proof has all branches closed, so we have proved the property $\mathbf{B}_s$ Reliable(msg). This concludes the analysis of this security property of the wide-mouthed frog protocol.

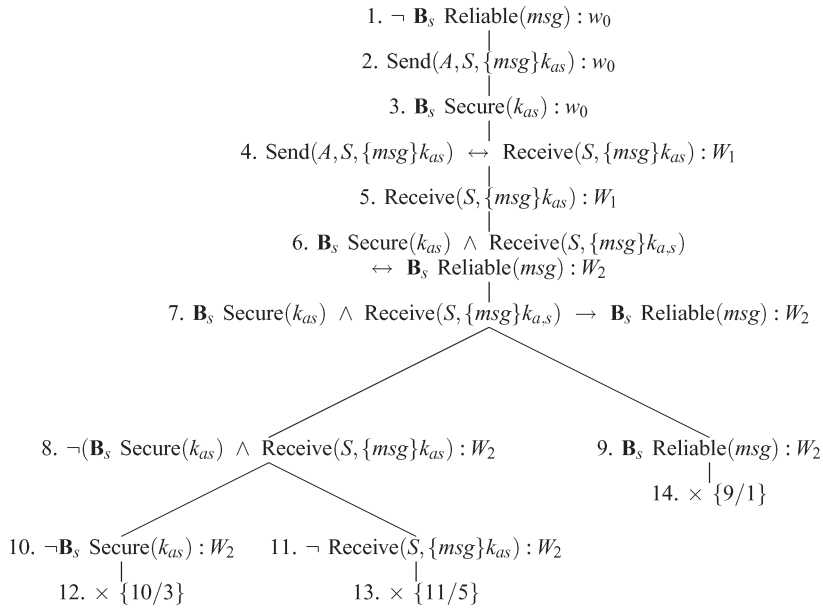

Fig. 2. Lowe modified wide-mouthed frog protocol.

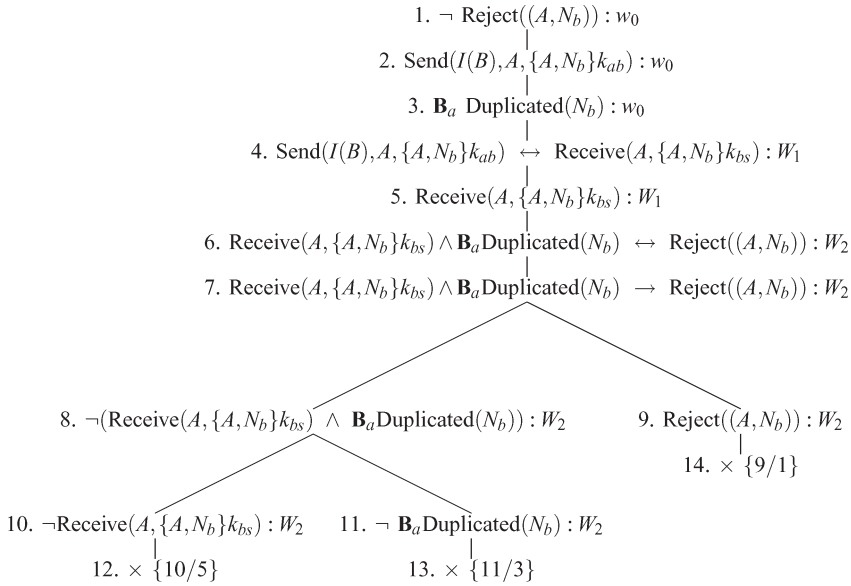

Fig. 3. Intruder detection.

C. Needham–Schroeder Symmetric Key Protocol

The amended version of the Needham–Schroeder symmetric key protocol is based on a symmetric encryption algorithm. It forms the basis for the Kerberos protocol. This protocol establishes a session key between two parties over a communication channel to protect further communication [36], [37].

1) A → B: A
2) B → A: $\{A, N_b\}_{K_{bs}}$
3) A → S: $A, B, N_a, \{A, N_b\}_{K_{bs}}$
4) S → A: $\{N_a, K_{ab}, B, \{K_{ab}, N_b, A\}_{K_{bs}}\}_{K_{as}}$
5) A → B: $\{K_{ab}, N_b, A\}_{K_{bs}}$
6) B → A: $\{N_b\}_{K_{ab}}$
7) A → B: $\{N_b - 1\}_{K_{ab}}$

The set of the security mechanisms of the system, denoted by $\mathcal{M}_n$, is defined as follows:

$$\mathcal{M}_n = \{k_{ab},\ k_{as},\ k_{bs}\}.$$

Then, we have the following security assumptions:

$$\mathbf{B}_a\ \text{Secure}(k_{ab}) \qquad \mathbf{B}_a\ \text{Secure}(k_{as})$$
$$\mathbf{B}_s\ \text{Secure}(k_{as}) \qquad \mathbf{B}_s\ \text{Secure}(k_{bs})$$
$$\mathbf{B}_b\ \text{Secure}(k_{bs}) \qquad \mathbf{B}_b\ \text{Secure}(k_{ab})$$

These assumptions are about the security of encryption keys.

As with the wide-mouthed frog protocol, we start with the assumption of the standard axioms R1–R7 given above and then

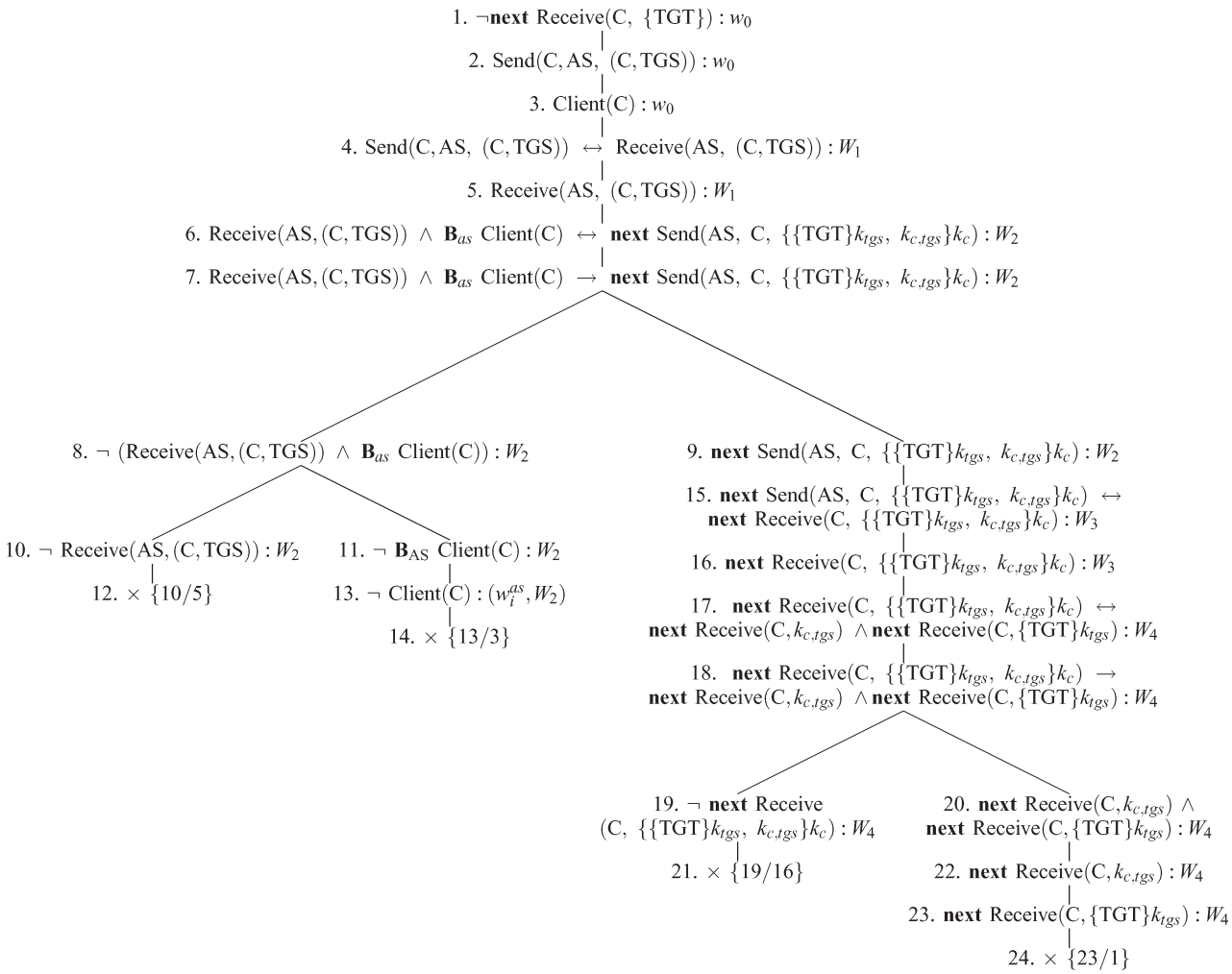

Fig. 4. Granting a TGT.

develop the specific axioms for the Needham–Schroeder protocol. We have the following rules that describe the authentication procedure.

N1) $\text{Receive}(B,(A)) \leftrightarrow \textbf{next}\ \text{Send}(B,A,\{A,N_B\}_{K_{bs}})$.
N2) $\mathbf{B}_A\ \text{Reliable}(\{A,N_B\}_{K_{bs}}) \leftrightarrow$ $\textbf{next}\ \text{Send}(A,S,(A,B,N_a,\{A,N_b\}_{K_{bs}}))$.
N3) $\mathbf{B}_s\ \text{Reliable}((A,B,N_a,\{A,N_b\}_{K_{bs}})) \leftrightarrow$ $\textbf{next}\ \text{Send}(S,A,\{N_a,K_{ab},B,\{K_{ab},N_b,A\}_{K_{bs}}\}_{K_{as}})$.
N4) $\mathbf{B}_a\ \text{Reliable}(\{N_a,K_{ab},B,\{K_{ab},N_b,A\}_{K_{bs}}\}_{K_{as}}) \leftrightarrow$ $\textbf{next}\ \text{Send}(A,b,\{K_{ab},N_b,A\}_{K_{bs}})$.
N5) $\mathbf{B}_b\ \text{Reliable}(\{K_{ab},N_b,A\}_{K_{bs}}) \leftrightarrow$ $\textbf{next}\ \text{Send}(B,A,\{N_b\}_{K_{ab}})$.
N6) $\mathbf{B}_a\ \text{Reliable}(\{N_b\}_{K_{a,b}}) \leftrightarrow$ $\textbf{next}\ \text{Send}(A,B,\{N_b-1\}_{K_{ab}})$.

Now, we have established a theory $\mathrm{T}_n = \{\mathrm{R1},\mathrm{R2},\mathrm{R3},\mathrm{R4},\mathrm{R5},\mathrm{R6},\mathrm{R7},\mathrm{N1},\mathrm{N2},\mathrm{N3},\mathrm{N4},\mathrm{N5},\mathrm{N6}\}$ for the amended Needham–Schroeder symmetric key protocol.

Assume that agent b sent a message $\{A,N_b\}k_{bs}$ to agent a. Later, it is possible that an intruder I masquerading as agent b (denoted by $I(B)$) may duplicate the message $\{A,N_b\}k_{bs}$ and forward it to agent a, but he/she is unable to create a new nonce if he/she does not know key k_{bs}. Furthermore, the message will be rejected because of the duplicated nonce. This prompts an error message, which is sent to agent b. As a result, agent b will be notified that an intruder tried to masquerade his/her identity. With the assumption that $\text{Send}(I(B),A,\{A,N_b\}k_{bs})$ and $\mathbf{B}_a\text{Duplicated}(N_b)$, we can prove $\text{Reject}((A,N_b))$. That is,

$$\mathrm{T}_n \cup \mathcal{S} \models_{\mathrm{TML}^+} \{\text{Send}\,(I(B),A,\{A,N_b\}k_{bs})\,, \mathbf{B}_a\ \text{Duplicated}(N_b)\} \Rightarrow \text{Reject}\,((A,N_b))\,.$$

The proof is shown in Fig. 3. In this tableau, node 14 is obtained by nodes 9 and 1 with the unifier $\{W_2/w_0\}$. This proof has all branches closed, so we have proved $\text{Reject}((A,N_b))$. This concludes the analysis of the attack detection process.

D. Kerberos Protocol

Kerberos is a trusted third-party authentication system [34]. For the analysis of the Kerberos system, we use the notations C, AS, and TGS to denote client, authentication server, and ticket granting server, respectively, TGT and SGT denote specific tickets; T_c denotes client's timestamp. k_c, k_{serv}, k_{tgs}, $k_{c,tgs}$, $k_{c,serv}$ denote specific keys. The Kerberos authentication process is given as follows.

1) C → AS: C, TGS.
2) AS → C: $\{\{\text{TGT}\}K_{tgs},K_{c,tgs}\}K_c$.

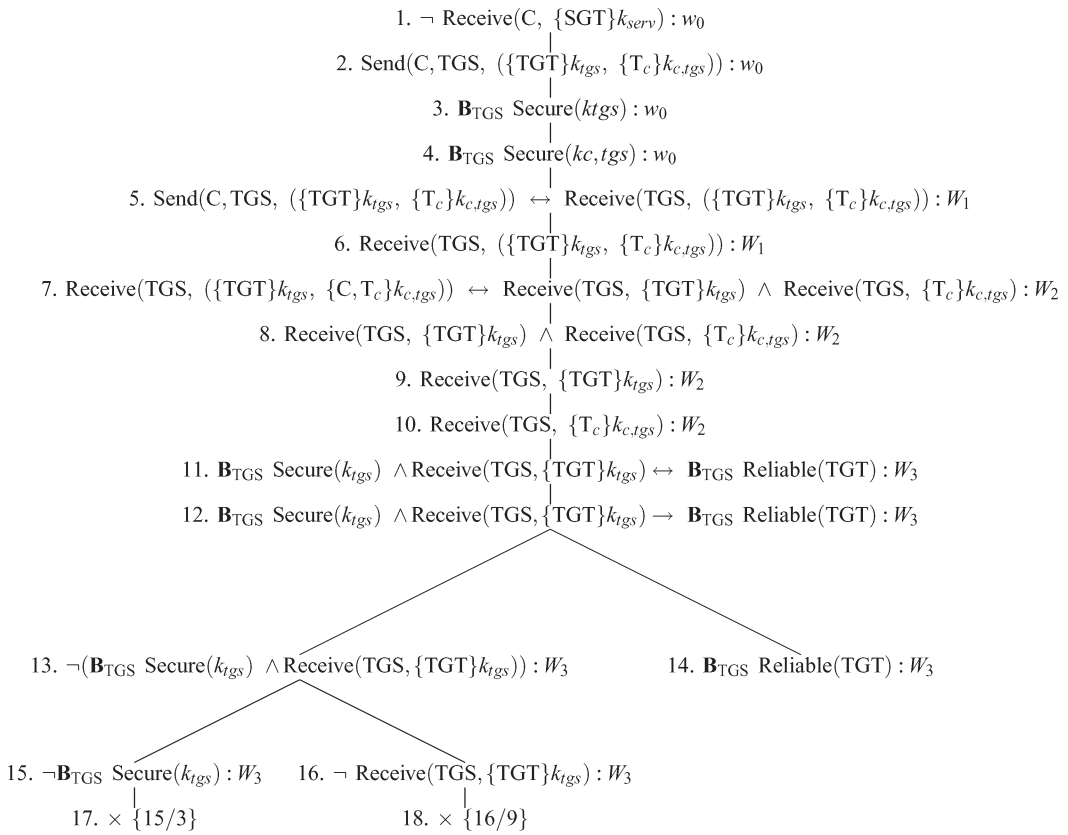

Fig. 5. Granting an SGT, part I. Verify the TGT.

3) C → TGS: Serv, $\{\text{TGT}\}K_{tgs}$, $\{\text{T}_c\}K_{c,tgs}$.
4) TGS → C: $\{\{\text{SGT}\}K_{serv}, K_{c,serv}\}K_{c,tgs}$.
5) C → Serv: $\{\text{SGT}\}K_{serv}, \{\text{T}_c\}K_{c,serv}$.
6) Serv → C: $\{\text{T}_c + 1\}K_{c,serv}$.

The set of the security mechanisms of the system, denoted by $\mathcal{M}_k$, is defined as follows:

$$\mathcal{M}_k = \{k_c,\ k_{serv},\ k_{tgs},\ k_{c,tgs},\ k_{c,serv}\}.$$

Then, we have the following security assumptions:

$\mathbf{B}_c$ Secure(k_c)	$\mathbf{B}_c$ Secure($k_{c,tgs}$)
$\mathbf{B}_c$ Secure($k_{c,serv}$)	$\mathbf{B}_{as}$ Secure(k_c)
$\mathbf{B}_{tgs}$ Secure(k_{tgs})	$\mathbf{B}_{tgs}$ Secure($k_{c,tgs}$)
$\mathbf{B}_{serv}$ Secure(k_{serv})	$\mathbf{B}_{serv}$ Secure($k_{c,serv}$)

These assumptions are about the security of encryption keys.

In our previous work [31], we have proposed a trust theory for the Kerberos authentication protocol, which includes 13 rules. In this paper, we propose a simplified theory. We have the following rules that describe the authentication procedure.

K1) Receive(AS, (C, TGS)) ∧ $\mathbf{B}_{as}$Client(C) ↔
next Send(AS, C, $\{\{\text{TGT}\}k_{tgs}, k_{c,tgs}\}k_c$).
K2) $\mathbf{B}_c$ Reliable($\{\{\text{TGT}\}k_{tgs}, k_{c,tgs}\}k_c$) ↔
next Send(C, TGS, ($\{\text{TGT}\}k_{tgs}$, $\{\text{T}_c\}k_{c.tgs}$)).
K3) $\mathbf{B}_{tgs}$ Reliable((TGT, T$_c$)) ↔
next Send(TGS, C, $\{\{\text{SGT}\}K_{serv}, K_{c,serv}\}K_{c,tgs}$).
K4) $\mathbf{B}_c$ Reliable($\{\{\text{SGT}\}K_{serv}, K_{c,serv}\}K_{c,tgs}$) ↔
next Send(C, Serv, ($\{\text{SGT}\}k_{serv}$, $\{\text{T}_c\}k_{c,serv}$)).
K5) $\mathbf{B}_{serv}$ Reliable((SGT, T$_c$)) ↔
next ServiceGranted(C, s).

Now, we have established a theory $\text{T}_k = \{\text{R1}, \text{R2}, \text{R3}, \text{R4}, \text{R5}, \text{R6}, \text{R7}, \text{K1}, \text{K2}, \text{K3}, \text{K4}, \text{K5}\}$ for the Kerberos protocol.

Example 1—Granting a TGT: With the assumptions that Client(C), Send(C, AS(C, TGS)), we can prove **next** Receive(C, $\{\text{TGT}\}k_{tgs}$). In other words, if a registered client sends a request to AS, then AS would issue a TGT. That is,

$$\text{T}_k \cup \mathcal{S} \models_{\text{TML}^+} \{\text{Client(C)}, \text{Send(C, AS(C, TGS))}\} \Rightarrow \mathbf{next}\ \text{Receive(C, }\{\text{TGT}\}k_{tgs}\text{)}.$$

The proof is shown in Fig. 4. In this tableau, node 24 is obtained by nodes 23 and 1 with the unifier $\{W_4/w_0\}$. This proof has all branches closed, so we have proved **next** Receive(C, $\{\text{TGT}\}k_{\text{tgs}}$). This concludes the analysis of the TGT granting process.

Example 2—Granting a SGT: With the assumption that Send(C, TGS, ($\{\text{TGT}\}k_{tgs}$, $\{\text{T}_c\}k_{c,tgs}$)), we can prove **next** Receive(C, $\{\text{SGT}\}k_{serv}$). In other words, if an agent sends a valid request to TGS, then TGS would issue an SGT. That is, $\text{T}_k \cup \mathcal{S} \models_{TML^+} \{\text{Send(C, TGS, (}\{\text{TGT}\}k_{tgs}, \{\text{T}_c\}k_{c,tgs}\text{))}\} \Rightarrow$ **next** Receive(C, $\{\text{SGT}\}k_{serv}$)

The proof is shown in Figs. 5–8. In this tableau, node 46 is obtained by nodes 45 and 1 with the unifier $\{W_8/w_0\}$.

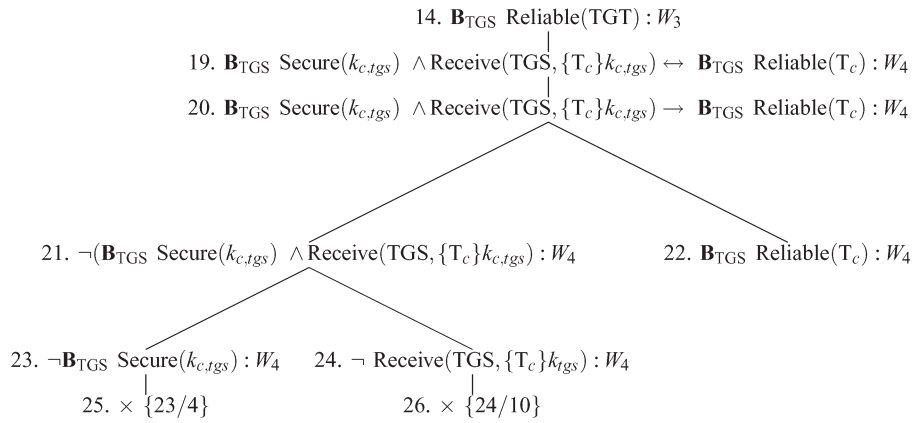

Fig. 6. Granting an SGT, part II. Verify the timestamp.

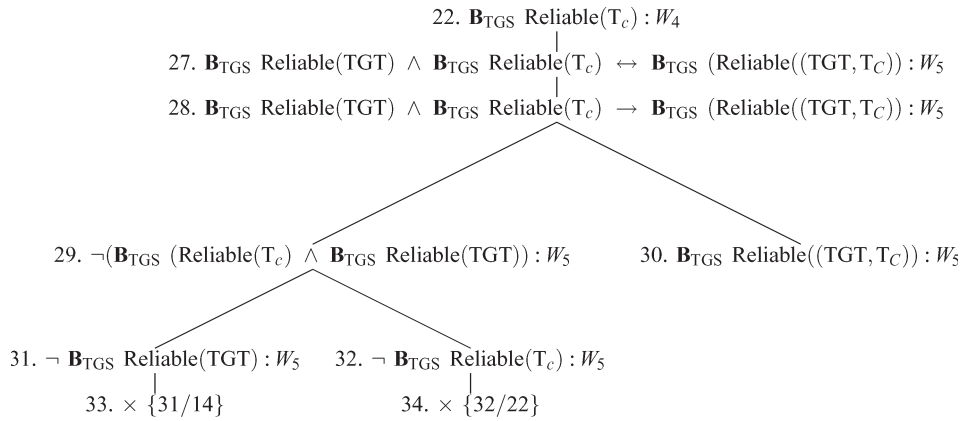

Fig. 7. Granting an SGT, part III. Verify a request.

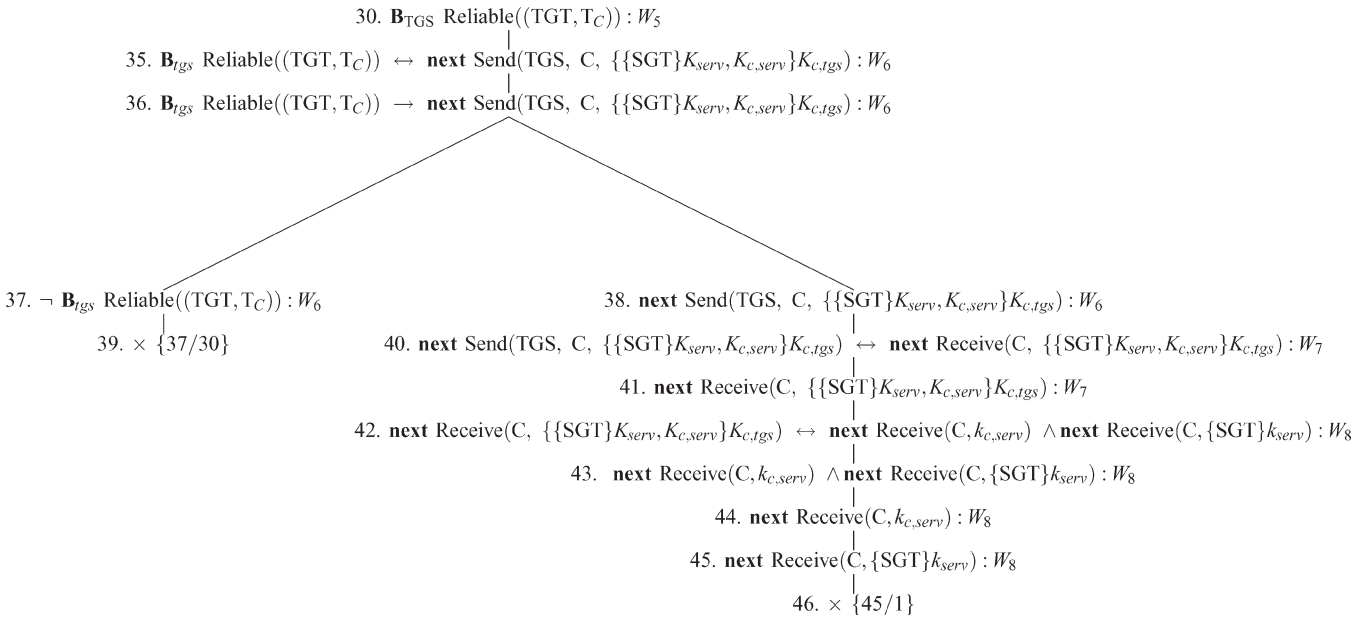

Fig. 8. Granting an SGT, part VI. Granting an SGT.

This proof has all branches closed, so we have proved **next** Receive(C, {SGT}k_{serv})). This concludes the analysis of the SGT granting process.

The above example has shown a whole procedure of establishing a theory of trust for a given communication system. Correctly identifying the security mechanisms of a given system is the basis for constructing a valid theory for the system. Defining appropriate predicates would be helpful to simplify the procedure of constructing the theory and so that the theory developed would easily be understood. To guarantee

that a theory is sound, we need to check every rule to make sure that it is consistent with the theory. Completeness for a theory of trust is also an important issue. It is involved in the analysis of security requirements and the fulfillment of these requirements.

V. Conclusion

We have shown how to use the labeled tableau system to verify the correctness of authentication systems. As we have pointed out before, there are a number of methods that have been developed for specifying and reasoning about agents beliefs; in particular, the BAN logic family has widely been discussed and applied for the analysis of authentication protocols. Since the BAN logic does not attempt to model knowledge, it can only be used to prove authentication systems. Furthermore, the BAN logic is neither a temporal belief logic nor a multiagent belief logic, so it is not suitable to express the evolution of agent beliefs and not suitable to express multiagent beliefs. Comparing with the BAN logic family and other belief logics, we have used a systematic and general approach to combine a temporal logic with a belief logic, so our approach is more flexible and adaptable. Furthermore, the advantages of our labeled tableau approach include the following: 1) it is simpler to generate a proof, and the security properties of protocols can easily be captured through the proof procedure; 2) it generates concise proofs, and the correctness of those proofs can be guaranteed; and 3) it can easily be extended for analyzing other security protocols.

The temporalization technique has certain implications on the expressive power of a combined logic. If we would like to be able to express agent beliefs about temporal properties, for certain real-life applications, TML^+ may not be the best choice. For example, in TML^+, we cannot express the assertion that "Alice believes that at the initial moment in time Bob is honest." We would need to be able to write down $\mathbf{B}_{Alice}$ **first** IsHonest(Bob), which would be possible if we added TML to SLTL (resulting in, e.g., SLTL^+). This leads to the conclusion that the choice of the logics to combine and the way in which they are combined are critical issues for whatever applications we have in mind for the combined logics. As this paper shows by several example trust theories, logics obtained by the temporalization technique are generally strong (expressive) enough for use in the analysis of security properties.

References

[1] M. Abadi and A. D. Gordon, "A calculus for cryptographic protocols: The spi calculus," *Inf. Comput.*, vol. 148, no. 1, pp. 1–70, Jan. 1999.

[2] M. Abadi and M. R. Tuttle, "A semantics for a logic of authentication (extended abstract)," in *Proc. 10th Annu. ACM Symp. Principles Distrib. Comput.*, 1991, pp. 201–216.

[3] V. D. Agrawal, R. D. Blanton, and M. Damiani, "Synthesis of self-testing finite state machines from high-level specifications," in *IEEE Int. Test Conf.*, 1996, pp. 757–766.

[4] R. J. Anderson and R. M. Needham, "Programming Satan's computer," in *Computer Science Today*. Berlin, Germany: Springer-Verlag, 1995, pp. 426–440.

[5] M. D. Backer, "On the verification of web services compatibility: A Petri net approach," in *On The Move Workshops*, 2004, pp. 810–821.

[6] D. A. Basin, S. Modersheim, and L. Vigano, "OFMC: A symbolic model checker for security protocols," *Int. J. Inf. Secur.*, vol. 4, no. 3, pp. 181–208, 2005.

[7] C. Bodei, P. Degano, R. Focardi, and C. Priami, "Primitives for authentication in process algebras," *Theor. Comput. Sci.*, vol. 283, no. 2, pp. 271–304, Jun. 2002.

[8] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, Feb. 1990.

[9] C. Dixon, M. C. F. Gago, M. Fisher, and W. V. D. Hoek, "Using temporal logics of knowledge in the formal verification of security protocols," in *Proc. 11th Int. Symp. Temporal Representation Reasoning*, 2004, pp. 148–151.

[10] O. B. Driss, P. Yim, O. Korbaa, and K. Ghédira, "Reachability search in timed Petri nets using constraint programming," in *IEEE Int. Conf. Syst., Man, Cybern.*, 2004, vol. 5, pp. 4923–4928.

[11] M. Finger and D. M. Gabbay, "Adding a temporal dimension to a logic system," *J. Logic, Lang. Inf.*, vol. 1, no. 3, pp. 203–233, Sep. 1992.

[12] M. Fitting and R. L. Mendelsohn, *First-Order Modal Logic*. Norwell, MA: Kluwer, 1999.

[13] M. Franceschet, A. Montanari, and M. de Rijke, "Model checking for combined logics with an application to mobile systems," *Autom. Softw. Eng.*, vol. 11, no. 3, pp. 289–321, Jun. 2004.

[14] D. M. Gabbay, "Fibring logics," *J. Logic, Lang. Inf.*, vol. 9, no. 4, pp. 511–513, Oct. 2000.

[15] L. Gong, R. Needham, and R. Yahalom, "Reasoning about belief in cryptographic protocols," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, 1990, pp. 234–248.

[16] G. Governatori, "Labelled tableaux for multi-modal logics," in *Proc. 4th Int. Workshop Theorem Proving With Analytic Tableaux Related Methods*, 1995, pp. 79–94.

[17] G. E. Hughes and M. J. Cresswell, *A New Introduction to Modal Logic*. London, U.K.: Routledge, 1996.

[18] A. J. I. Jones and B. S. Firozabadi, "On the characterisation of a trusting agent—Aspects of a formal approach," in *Trust and Deception in Virtual Societies*. Norwell, MA: Kluwer, 2001, pp. 157–168.

[19] R. Kailar, "Accountability in electronic commerce protocols," *IEEE Trans. Softw. Eng.*, vol. 22, no. 5, pp. 313–328, May 1996.

[20] R. A. Kemmerer, C. Meadows, and J. K. Millen, "Three systems for cryptographic protocol analysis," *J. Cryptol.*, vol. 7, no. 2, pp. 79–130, Jun. 1994.

[21] V. Kessler and H. Neumann, "A sound logic for analyzing electronic commerce protocols," in *Proc. ESORICS*, 1998, pp. 345–360.

[22] S. A. Kripke, "Semantical considerations on modal logic," *Acta Philos. Fenn.*, vol. 16, pp. 83–94, 1963.

[23] Y. Lin, "Fair control of Petri nets," in *IEEE Int. Conf. Syst., Man, Cybern.*, 2004, vol. 2, pp. 1672–1677.

[24] C. Liu, "Logical foundations for reasoning about trust in secure digital communication," in *Proc. AI: Advances Artif. Intell.* Berlin, Germany: Springer-Verlag, 2001, vol. 2256, pp. 333–344.

[25] C. Liu and M. A. Orgun, "Executing specifications of distributed computations with Chronolog(MC)," in *Sac*, 1996, pp. 393–400.

[26] C. Liu and M. A. Ozols, "Trust in secure communication systems—The concept, representations, and reasoning techniques," in *Proc. AI: Advances Artif. Intell.* Berlin, Germany: Springer-Verlag, 2002, vol. 2557, pp. 60–70.

[27] C. Liu, M. A. Ozols, and M. A. Orgun, "A temporalised belief logic for specifying the dynamics of trust for multi-agent systems," in *Proc. 9th Asian Comput. Sci. Conf.* Berlin, Germany: Springer-Verlag, 2004, vol. 3321, pp. 142–156.

[28] D. Longley and S. Rigby, "An automatic search for security flaws in key management schemes," *Comput. Secur.*, vol. 11, no. 1, pp. 75–89, Mar. 1992.

[29] G. Lowe, "A family of attacks upon authentication protocols," Dept. Math. Comput. Sci., Univ. Leicester, Leicester, U.K., Tech. Rep. 1997/5, 1997.

[30] J. Ma and M. A. Orgun, "Trust management and trust theory revision," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 451–460, May 2006.

[31] J. Ma and M. A. Orgun, "Specifying agent beliefs for authentication systems," in *Proc. 4th Eur. Conf. Universal Multiservice Netw.*, 2007, pp. 410–418.

[32] J. Ma and M. A. Orgun, "Formalising theories of trust for authentication protocols," *Inf. Syst. Frontiers*, vol. 10, no. 1, pp. 19–32, Mar. 2008.

[33] W. May, "A tableau calculus for a temporal logic with temporal connectives," in *Proc. Int. Conf. Automated Reasoning With Analytic Tableaux Related Methods*. Berlin, Germany: Springer-Verlag, 1999, vol. 1617, pp. 232–246.

[34] S. P. Miller, C. Neuman, J. I. Schiller, and J. H. Saltzer, "Kerberos authentication and authorization system," in *Project Athena Technical Plan*. E.2.1. Cambridge, MA: MIT, 1987.

[35] L. Moser, "A logic of knowledge and belief for reasoning about computer security," in *Proc. Comput. Secur. Found. Workshop II*, 1989, pp. 57–63.
[36] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Commun. ACM*, vol. 21, no. 12, pp. 993–999, Dec. 1978.
[37] M. Nesi and G. Rucci, "Formalizing and analyzing the Needham-Schroeder symmetric-key protocol by rewriting," *Electron. Notes Theor. Comput. Sci.*, vol. 135, no. 1, pp. 95–114, Jul. 2005.
[38] P. C. V. Oorschot, "Extending cryptographic logics of belief to key agreement protocols," in *Proc. 1st ACM Conf. Comput. Commun. Secur.*, 1993, pp. 233–243.
[39] M. A. Orgun, G. Governatori, and C. Liu, "Modal tableaux for verifying stream authentication protocols," *Auton. Agents Multi Agent Syst.*, vol. 19, no. 1, pp. 53–75, Aug. 2009.
[40] P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, and B. Roscoe, *Modelling and Analysis of Security Protocols*. Reading, MA: Addison-Wesley, 2001.
[41] S. T. Sarasamma, Q. A. Zhu, and J. Huff, "Hierarchical Kohonenen net for anomaly detection in network security," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 2, pp. 302–312, Apr. 2005.
[42] D. X. Song, S. Berezin, and A. Perrig, "Athena: A novel approach to efficient automatic security protocol analysis," *J. Comput. Secur.*, vol. 9, no. 1/2, pp. 47–74, 2001.
[43] P. F. Syverson and P. C. V. Oorschot, "A unified cryptographic protocol logic," in "NRL Publication," Naval Res. Lab., Washington DC, 1996.
[44] Y. Wang, "A cognitive informatics reference model of autonomous agent systems (AAS)," *Int. J. Cogn. Informatics Natural Intell.*, vol. 3, no. 1, pp. 1–16, 2009.
[45] G. Wedel and V. Kessler, "Formal semantics for authentication logics," in *Proc. ESORICS*. Berlin, Germany: Springer-Verlag, 1996, vol. 1146, pp. 219–241.
[46] R. Yahalom, B. Klein, and T. Beth, "Trust relationships in secure systems—A distributed authentication perspective," in *Proc. IEEE Symp. Res. Secur. Privacy*, 1993, pp. 150–164.
[47] Z. Yu, J. J. P. Tsai, and T. J. Weigert, "An automatically tuning intrusion detection system," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 373–384, Apr. 2007.

Ji Ma received the Bachelor of Computer and Information Science (with Honours) degree from the University of South Australia, Adelaide, Australia, in 2004 and the Ph.D. degree in computer science from Macquarie University, Sydney, Australia, in 2009.

He is currently a Research Fellow with Griffith University, Brisbane, Australia. He is also with the Department of Computing, Macquarie University. His research interests include multi-agent systems, formal methods, system modeling, and system verification.

Dr. Ma is a recipient of the prestigious Research Areas and Centres of Excellence (RAACE) Scholarship from Macquarie University during his Ph.D. studies. He has served on the Program Committees of the First and Second IEEE International Workshops on Software Engineering for Context Aware Systems and Applications in 2008 and 2009.

Mehmet A. Orgun (SM'96) received the Ph.D. degree in computer science from the University of Victoria, Victoria, Canada, in 1991.

In 1991, he was a Postdoctoral Fellow with the University of Victoria. Since 1992, he has been with the Department of Computing, Macquarie University, Sydney, Australia, where he was a Lecturer from 1992 to 1995, was a Senior Lecturer from 1996 to 1999, and has been an Associate Professor since 2000. He is a Cofounder of the Intelligent Systems Group, Macquarie University. He has been serving on the editorial boards of *The Journal of Universal Computer Science*, *The Open Cybernetics and Systemics Journal*, and *Expert Systems: The Journal of Knowledge Engineering*. His current research interests include data mining, multiagent systems, reactive and distributed systems, and temporal reasoning.

Dr. Orgun has served on the Program and Organizing Committees of numerous international conferences and workshops, including the IEEE Computer Society Signature Conference on Computer, Software and Applications (COMPSAC) in 2008 and 2009, the IEEE International Conference on Cognitive Informatics (ICCI) in 2007, and the IEEE International Conference on Information Reuse and Integration (IRI) in 2007 and 2009. He was a Cochair of the Second, Third, and Fourth IEEE International Workshops on Engineering Semantic Agent Systems in 2007, 2008, and 2009 and a Program Cochair of the Australian Joint Conference on Artificial Intelligence in 2007. He is currently a Conference Cochair of the Second International Conference on Security of Information and Networks (2009).

Abdul Sattar received the Ph.D. degree in artificial intelligence from the University of Alberta, Edmonton, Canada, in 1992.

He was a Lecturer in physics in Rajasthan, India, from 1980 to 1982, and a Research Scholar with Jawaharlal Nehru University, New Delhi, India, from 1982 to 1985, the University of Waterloo, Waterloo, ON, Canada, from 1985 to 1987, and the University of Alberta, Edmonton, AB, Canada, from 1987 to 1991. Since 1992, he has been with Griffith University, Brisbane, Australia, where he was a Lecturer from 1992 to 1995, was a Senior Lecturer from 1996 to 1999, has been a Professor of computer science and artificial intelligence since 2000, and is the founding Director of the Institute for Integrated and Intelligent Systems. He is also a Research Leader with the National ICT Australia, Queensland Research Laboratory. He is a member of the editorial/review board of leading journals. His research interests include knowledge representation and reasoning, constraint satisfaction, intelligent scheduling, rational agents, propositional satisfiability, temporal reasoning, temporal databases, and bioinformatics.

Dr. Sattar is a Life Member of the Association for the Advancement of Artificial Intelligence. He has served as a member of the Program Committee or Senior Program Committee of numerous national and international conferences. He has served as the Program or Conference Committee Chair for the Australian Joint Conference on Artificial Intelligence (AI) in 1997, 2006, and 2007, the Pacific Rim International Conference on Artificial Intelligence (PRICAI) in 2002 and 2010, and the International Symposium on Temporal Representation and Reasoning and International Conference on Temporal Logic (TIME/ICTL) in 2003.