Reasoning about Dynamics of Trust and Agent Beliefs

Mehmet A. Orgun
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
Email: mehmet@comp.mq.edu.au

Chuchang Liu
Information Networks Division
Defence Science and Technology Organisation
PO Box 1500, Edinburgh, SA 5111, Australia
Email: Chuchang.Liu@dsto.defence.gov.au

Abstract

Many formal frameworks have been proposed for specify ing and reasoning about the notion of trust and trust mech anisms in agentbased systems. Typed Modal Logic (TML) is a logic of beliefs which is suitable for for the description of, and reasoning about, trust for multiagent systems by formalising trust policies of the system and agent meta beliefs in its security mechanisms. In this paper, by adopting the methodology of Finger and Gabbay for temporalising a logic system, we combine TML with a temporal logic, so that the users can also model evolving theories of trust. In the resulting logic, denoted by TML$^+$, temporal properties of trust and agent beliefs can be expressed in a natural way by combinations of temporal and modal belief operators.

1. Introduction

Multiagent systems (MAS) consist of a collection of agents that interact with each other in dynamic and un predictable environments. A very basic problem regarding security properties of MAS is that whether a message sent by an agent is reliably received by other agents and whether the message received is regarded as reliable in the view of receivers. The problem generally depends on the trust that agents would put in the security mechanisms of a system [2]. It has been argued that, in order to support the analysis of various security mechanisms in MAS, it is necessary to systematically investigate formal representation techniques, and provide more generic tools for the specication, and reasoning about security mechanisms in MAS.

Several logics have been proposed for describing MAS, for example, logics for knowledge and belief [6], logics for goals and intentions [16], etc. These formalisms have successfully been used for dealing with certain aspects of agents in MAS, but they generally ignore some other aspects such as dynamics of trust. Liu [9] proposed a belief logic, called the TML (Typed Modal Logic), which extends first order logic with typed variables and modal belief operators. Based on TML, systemspecific theories of trust can be constructed, and they provide a basis for reasoning about trust in particular environments and systems [9], [12].

The work presented in this article has been supported in part by The Australian Research Council (ARC).

TML is suitable to express static properties, for example, the assertion "Alice believes that Bob has the key (k)" can be formalised in TML as $\mathbb{B}_{alice} has(bob, key)$ where $\mathbb{B}_{alice}$ is the modal belief operator for agent *Alice*. However, trust changes (as time progresses) or evolves dynamically when we gain or lose confidence in other agents. One agent may be trusted by some agents today, but tomorrow the situation may change. For example, we may have the assertion that "Alice believes that Bob has the key today but tomorrow Alice does not believe that Bob has the key." TML may not be able to express such dynamics as it lacks a temporal dimension. 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 in a multiagent system. This leads us to logics suitable for representing temporal properties.

Temporal logics have the ability to deal with dynamics, i.e., the changes in time of a set of properties, and they have been used successfully in many areas such as program verification, modelling concurrent computations, and speci fying communication protocols. In this paper, we combine TML with a temporal logic called TLC [10] using the temporalising technique proposed by Finger and Gabbay [3] which allows for integration of two logics in a certain way. Such a temporalised logic can be applied for reasoning about timedependent properties regarding agent beliefs in multi agent systems [11]. Now the assertion given above can be formalized in TML$^+$ by the following formula:

$$\mathbb{B}_{alice}\ has(bob, key)\ \wedge\ \mathbf{next}\ \neg\mathbb{B}_{alice}\ has(bob, key)$$

where temporal operator **next** refers to "tomorrow". Our work is therefore motivated by the need to provide a logical framework to express timedependent properties of trust for multiagent systems and to develop sufficient techniques for describing the dynamics of trust (that is, evolving theories of trust) within such systems.

The rest of the paper is organized as follows. Section 2 discusses the notions of trust and beliefs in the agent setting. Section 3 gives an introduction to TML, TLC and TML$^+$. To increase its expressive power, we extend TML$^+$ with a kind of a temporal merging operator called **fby** (read "followedby"). Section 5 discusses an application of TML $^+$ by building an evolving trust theory for a Public Key

0-7803-9788-6/06/$20.00 ©2006 IEEE.

Infrastructure (PKI). Section 6 discusses a general form of "temporalising" as a hierarchy combination technique. The last section outlines future research directions.

2. Trust and Belief

The notion of trust is essential for understanding the interactions between agents such as human beings, machines, organizations, and other entities [17]. Linguistically, the notion of "trust" is closely related to "true" and "faithful", with a usual dictionary meaning of "assured reliance on the character, the integrity, justice, etc., of a person, or something in which one places confidence." So, in common English usage "trust" is what one places his confidence in.

Digital communication involves computer systems and networks. A computer system can be regarded as an interconnection of people, hardware, and software, together with its external connections. We view a secure digital communication environment (e.g., the Internet) as a large complex system consisting of a number of agents (that is, a multiagent system). Agents can be a person, a computer, a handheld device or some other entity. Agent interactions are unavoidable and also necessary in many applications. Agents need to trust others if they are to have confidence that such interactions will lead to a desirable outcome. When we say that agent A trusts another agent B, this means that (in some sense) the two agents are situated in a state in which, from A's perspective, certain actions by B will be chosen under certain circumstances. In other words, A may believe that B will truthfully do certain actions which concern A.

In discussing formal descriptions of trust, we need to observe the following features within the notion of trust:

- There is no global trust in a secure digital communication environment. In other words, there are no agents who can be trusted by all others. This is most obvious in distributed systems or mobile systems. Even in a hierarchical system, such as a hierarchical PKI [14], although it is more likely that we may assume that all agents would trust the top Certification Authority, in practice there may still be some agents who do not trust it and may try to check its behavior in a variety of ways. Trust depends on the observer (agent).
- Trust is not static. The trust of an agent can be changed dynamically. For example, for two weeks agent A trusts agent B, but this morning A found that B lied to A, so A no longer trusts B.
- There is no full trust; an agent cannot in general trust all of the statements provided by another agent. So, we choose a limited trust model, where "agent A trusts agent B" means that A will only trust B on some topics.
- A trust relation may lack the properties of transitivity and symmetry.

Let us consider the case in a Public Key Infrastructure (PKI) which manages public keys, where agent *Alice* wants to communicate securely with agent *Bob*, then *Alice* has to obtain *Bob*'s public key first. The PKI provides a mechanism for users to retrieve required certificates, so *Alice* can retrieve any certificate she requires. Once *Alice* has *Bob*'s certificate, in which Bob's public key is bound, if *Alice* believes that the certificate is valid (in other words, she believes that the certificate can be trusted), then she may use the public key in Bob's certificate to send secure messages to Bob.

We may say that each agent asserts the truth of the proposition $\texttt{valid}(C)$ where C is a certificate and the interpretation of $\texttt{valid}(C)$ is that it returns $true$ if and only if C is valid. Such an assertion made by an agent is related to the agent's beliefs. In fact, $Alice$ would use Bob's certificate only when she cannot prove $\texttt{valid}(Bob's\ certificate)$ to be false from her beliefs. More strongly, *Alice* is not prepared to use *Bob*'s certificate unless she can prove $\texttt{valid}(Bob's\ certificate)$ from her own beliefs. To infer $\texttt{valid}(\text{Bob}'s\ certificate)$ from her belief, $Alice$ has to use some assumptions. In our approach, such assumptions will be encapsulated in a notion of trust for the system.

From the above analysis, we may see that reasoning about trust actually involves reasoning about beliefs. Therefore, a theory of trust may be based on a logic that possesses the ability to represent beliefs. As Rangan [13] has pointed out, belief represents a disposition of an agent on a proposition, so a logic of expressing propositional dispositions should be able to expressing the required relations between believers and attitudes. Classical firstorder logic cannot handle such relations well. The modal logic approach is able to enhance propositional and firstorder logics with modal operators to represent agents' beliefs.

As mentioned above, trust is the outcome of observations leading to the belief that the actions of another may be relied upon, without explicit guarantee, to achieve a goal in a risky situation. In the above example, suppose that for one reason or another, Alice has lost her trust in Bob. Trust therefore should be developed over time as the outcome of a series of confirming observations [2]. To put it simply, trust changes (or evolves) dynamically. To capture the evolution of trust, we need to consider a formalism that is not only capable of expressing agent beliefs but also their dynamics.

3. Temporalised Belief Logic

This section introduces the temporalised belief logic and its constituents.the temporalisation of TML is given elsewhere [11].

3.1. Temporal logic with Clocks

TLC has two temporal operators, $\mathbf{first}$ and $\mathbf{next}$, which refer to the initial moment and the next moment in time respectively [10]. These operators are applied to the formulas, not to the terms of TLC. The meaning of TLC formulas are defined with respect to given local clocks (subsequences of a global clock). The global clock is the increasing sequence of natural numbers, i.e., $\langle 0, 1, 2, \ldots \rangle$ and a local clock (or simply, a clock) is defined as an infinite subsequence of the global clock.

In TLC, the assertions such as "Bob has the key initially" and "Alice has the key tomorrow" can be expressed by formulas "$\mathbf{first}\ has(bob, key)$" and "$\mathbf{next}\ has(alice, key)$" respectively. We also introduce a third temporal operator called **fby** (followedby) into TLC to increase its expressive power. The assertion "Bob has the key initially and Alice has the key all the time" can be expressed by the formula: "$has(bob, key)\ \mathbf{fby}\ has(alice, key)$". The **fby** operator can not be expressed in terms of any combinations of boolean connectives and/or basic temporal operators.

In TLC, since past is not unbounded in the underlying timeline, we cannot have a "yesterday" operator that is symmetric to the "tomorrow" operator. However, **fby** can also be used as a kind of a yesterday operator: the assertion "Bob had the key yesterday" can now be expressed as "$false\ \mathbf{fby}\ has(bob, key)$".The meaning of a formula of the form $\varphi\ \mathbf{fby}\ \psi$ is determined by the value of φ at time 0, and by the previous (yesterday's) value of ψ at times greater than 0. Since there is no yesterday of time 0, hence falsity as the first argument to the operator.

Table 1 gives an intuitive explanation of the interpretation of the temporal operators of TLC.

Formula	Truth value								
φ	T	T	F	F	T	F	T	F	...
ψ	F	F	T	T	F	F	T	T	...
first φ	T	T	T	T	T	T	T	T	...
next φ	T	F	F	T	F	T	F	...	...
φ **fby** ψ	T	F	F	T	T	F	F	T	...
Time	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	...

Table 1. Interpretation of temporal operators (φ and ψ are formulas; T represents value **true** and F value **false**)

More formally, we have the following definition of the semantics of atomic formulas of TLC:

Definition 1 (time models): A time model for the logic TLC has the form $\mathbf{c} = \langle C, <, \pi^{(\mathbf{c})} \rangle$, *where* $C = \langle t_0, t_1, t_2, \ldots \rangle$ *is a clock,* $<$ *is the usual ordering relation over* C, *and* $\pi^{(\mathbf{c})}$ *is an assignment function that gives a value* $\pi^{(\mathbf{c})}(t, q) \in \{true, false\}$ *for any any time point* t *in* C *and any atomic formula* q.

Then the semantics of the temporal operators of TLC are given as follows (note that the semantics of the classical/boolean operators of TLC can be given in a straightforward manner and are therefore omitted):

- $\mathbf{c}, t_i \models \mathbf{first}\ \varphi$ iff $t_0 \models \varphi$.
- $\mathbf{c}, t_i \models \mathbf{next}\ \varphi$ iff $t_{i+1} \models \varphi$.
- $\mathbf{c}, t_i \models \varphi\ \mathbf{fby}\ \psi$ iff $\mathbf{c}, t_0 \models \varphi$ when $i = 0$ or $\mathbf{c}, t_{i-1} \models \psi$ when $i \geq 1$.
- satisfaction in the model $\langle C, <, \pi^{(\mathbf{c})} \rangle$ is defined as satisfaction at some time point on C.

The axioms (A0–A7) and inference rules of TLC are given below. Let $\bigtriangledown$ stand for **first** and **next**:

A0. All classical tautologies.
A1. $\bigtriangledown(\neg\varphi) \leftrightarrow \neg(\bigtriangledown\varphi)$.
A2. $\bigtriangledown(\varphi \wedge \psi) \leftrightarrow (\bigtriangledown\varphi) \wedge (\bigtriangledown\psi)$.
A3. $\bigtriangledown(\mathbf{first}\ \varphi) \leftrightarrow \mathbf{first}\ \varphi$.
A4. $\mathbf{first}\ (\varphi\ \mathbf{fby}\ \psi) \leftrightarrow \mathbf{first}\ \varphi$.
A5. $\mathbf{next}\ (\varphi\ \mathbf{fby}\ \psi) \leftrightarrow \psi$.
A6. $(\mathbf{first}\ \varphi)\ \mathbf{fby}\ (\mathbf{next}\ \varphi) \leftrightarrow \varphi$.
A7. $(\varphi\ \mathbf{fby}\ \psi)\ \mathbf{fby}\ \gamma \leftrightarrow \varphi\ \mathbf{fby}\ \gamma$.
US. From $\vdash \varphi(p)$ infer $\vdash \varphi(p/\psi)$, where p is a propositional symbol and ψ any formula and $\varphi(p/\psi)$ is a formula obtained by substituting all appearances of p in φ by ψ. (Uniform Substitution)
MP. From $\vdash \varphi$ and $\vdash \varphi \rightarrow \psi$ infer $\vdash \psi$. (Modus Ponens)
TN. From $\vdash \varphi$ infer $\vdash \bigtriangledown\varphi$. (Temporal Necessitation)
FI. From $\vdash \mathbf{first}\ \varphi$ and $\vdash \psi$, infer $\vdash \varphi\ \mathbf{fby}\ \psi$. (Followed by Introduction)

3.2. Belief Logic

Typed Modal Logic (TML) [9] is an extension of first order logic with typed variables and modal operators to express beliefs of rational agents. It is a variant of the modal logic **KD** of beliefs [15]. In discussing the syntax of TML, we need to distinguish two different concepts, *messages* (in firstorder logic, called terms) and *formulas*. Messages can be names of agents, certificates, public keys, private keys, dates, In the vocabulary of TML, apart from (typed) variables, function and predicate symbols, we have the classical boolean connectives, quantifiers, and modal belief operators: $\mathbb{B}_{a_1}, \ldots, \mathbb{B}_{a_m}$, where we assume that $\mathcal{A} = \{a_1, \ldots, a_m\}$ is the set of agents and $\mathbb{B}_{a_i}$ $(i = 1, \ldots, m)$ stand for "agent a_i believes that".

The semantics for TML is based on the Kripke possible world semantics [8], using the notion of possible global states for the interpretation of agent beliefs. A *classical Kripke model* for TML is defined as a tuple $\mathbf{m} = \langle S, R_1, \ldots, R_n, \pi \rangle$, where S is the set of states; and each R_i, $i = 1, \ldots, n$, is a relation over S, (called the *possibility relation* according to agent a_i), and is defined as follows: R_i is a nonempty set consisting of state pairs (u, v) such that $(u, v) \in R_i$ iff, at state u, agent a_i considers the state v possible (or accessible); and π is the *assignment function*, which gives a value $\pi(u, q) \in \{true, false\}$ for any $u \in S$ and atomic formula q. Formula φ is satisfiable in the model $\mathbf{m}$ if there exists $u \in S$ such that $\mathbf{m}, u \models \varphi$.

Then the semantics of the belief operators of TML are given as follows (we omit that the semantics of the classical/boolean operators of TML):

- $\mathbf{m}, u \models \mathbb{B}_{a_i}\varphi$ iff, for all v such that $(u, v) \in R_i$, $\mathbf{m}, v \models \varphi$.

The axioms (B1–B3) and inference rules (R1–R4) of TML are given below:

B1. All axioms of the classical firstorder logic.
B2. $\mathbb{B}_{a_i}(\varphi \rightarrow \psi) \wedge \mathbb{B}_{a_i}\varphi \rightarrow \mathbb{B}_{a_i}\psi$, $i = 1, \ldots, m$.
B3. $\mathbb{B}_{a_i}(\neg\varphi) \rightarrow \neg(\mathbb{B}_{a_i}\varphi)$, $i = 1, \ldots, m$.
R1. From $\vdash \varphi$ and $\vdash \varphi \rightarrow \psi$ infer $\vdash \psi$ (Modus Ponens)
R2. From $\vdash \forall X\varphi(X)$ infer $\vdash \varphi(Y)$ (Instantiation)
R3. From $\vdash \varphi(X)$ infer $\vdash \forall X\varphi(X)$ (Generalisation)
R4. From $\vdash \varphi$ infer $\vdash \mathbb{B}_{a_i}\varphi$, $i = 1, \ldots, m$. (Necessitation)

3.3. Adding TLC to TML

Having decided to combine TLC and TML, we adopt the method of Finger and Gabbay [3], named as "adding a temporal dimension" (to a logic system). When TLC is *added* to TML, substitutions of formulas of TML for the atoms of TLC are allowed, but substitutions of formulas of TLC for atoms of TML are not allowed. We write $\text{TML}^+ = TML + TLC$ to indicate that a new logic TML^+ is constructed by adding TLC onto TML in this way.

In the resulting logic, there are certain restrictions on the use of temporal and belief operators, because of the hierarchical combination of belief and temporal logics used. Hence in TML^+, we can only express statements about temporal aspects of agent beliefs such as "At the initial moment in time, Alice believes that Bob holds the key": $\mathbf{first}\ \mathbb{B}_{alice}\ holds(bob, k)$.

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 temporalisation 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 definition of any formula of TML^+ would involve a time model and a time reference initially, 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.

The discussion of the semantics in the case of the Kripke models for TML with time models for TLC can be laid out in three levels: using a single time model, or considering a set of time models with the same clock, or based on different clock models. In this paper, to simplify the presentation, we assume the single time model in which all formulas are defined on the same (global) clock. We refer the reader to the literature for the technical details of how TLC is added to TML under this assumption [11].

In the following, we discuss the axioms and rules of inference of TML^+ (inherited from the constituent logics TLC and TML):

The axioms and rules of inference of TLC;

For every formula φ of TML^+, if $\vdash_{TML}\ \varphi$, then $\vdash_{TML^+}\ \varphi$. where $\vdash_{TML}\ \varphi$ means that φ is a theorem of the system TML and $\vdash_{TML^+}\ \varphi$ means that φ is a theorem of the system TML^+.

The item above is a new rule of inference called theoremhood preservation (or **TP**). Therefore, the system TML^+ contains, as rules of inference, **US**, **MP**, **TN**, **FI** and **TP**. The soundness for the logic TML^+ depends on the soundness theorems for belief logic and TLC. The completeness theory for TML^+ can be proved by the techniques used in [3].

4. Developing a Trust Theory

Reasoning about the security properties of a given security mechanism of MAS usually involves: (1) choosing a (combined) logic appropriate for the specific application at hand, (2) defining functions and predicates, and then formalizing our knowledge about the trustbased security mechanisms of MAS as a set of axioms and/or rules in a consistent theory (say Γ_t), and (3) expressing the underlying assumptions as a set of formulas (say Γ_a). Once a consistent theory has been formalised, we can check whether an essential (security or other) property (say ϕ) logically follows from the theory and the assumptions by constructing a proof of the property. In other words, we would like to check whether $\Gamma_t \cup \Gamma_a \vdash \phi$. In the following, we show how to construct a trust theory for an agentbased system.

The Public Key Infrastructure (PKI) enables users of a basically unsecured public network such as the Internet to securely exchange information through the use of public and private cryptographic key pairs that are obtained and shared through a trusted evaluated infrastructure. In the PKI system, the authenticity of public keys is certified by Certification Authorities (CAs). Digital certificates issued by CAs contain information about the holder, the holders public key, an expiration date, and the digital signature of the issuer (the certification authority). We assume that PKI certificates have the form as $\texttt{Cert}[\texttt{I}, \texttt{DS}, \texttt{DE}, \texttt{S}, \texttt{PK}, \texttt{E}, \texttt{Sig}]$, where I is the issuer, DS and DE are the start date and expiry date respectively, S is the subject of the certificate, PK is the value of the public key for S, E is the value of the extension field, and Sig holds the signature of the issuer I.

For any certificate $C = \texttt{Cert}[\texttt{I}, \texttt{DS}, \texttt{DE}, \texttt{S}, \texttt{PK}, \texttt{E}, \texttt{Sig}]$, there are eight projection functions, which are applied to obtain the values of its components. They are: $\overline{\texttt{I}}(C) = \texttt{I}$, $\overline{\texttt{DS}}(C) = \texttt{DS}$, $\overline{\texttt{DE}}(C) = \texttt{DE}$, $\overline{\texttt{S}}(C) = \texttt{S}$, $\overline{\texttt{PK}}(C) = \texttt{PK}$, $\overline{\texttt{E}}(C) = \texttt{E}$, $\overline{\texttt{Sig}}(C) = \texttt{Sig}$ and $\overline{\texttt{tbs}}(C) = (\texttt{I}, \texttt{DS}, \texttt{DE}, \texttt{S}, \texttt{PK}, \texttt{E})$

The other denotations used in building a trust theory for the PKI include: $\{M\}_{\hat{K}}$ – the message M encrypted using $\hat{K}$, and $\langle M\rangle_{\hat{K}}$ – the message M decrypted using $\hat{K}$, where $\hat{K}$ is the public key or the private key of a key pair; and CRL_A – the certificate revoked list of an agent A at the current state.

Based on TML, Liu [9] proposed a systemspecific trust theory applied for the verification of certificates within the PKI system, which contains the following axioms:

T1. $K = (PK, SK) \rightarrow \langle\{M\}_{SK}\rangle_{PK} = M$.
T2. $K = (PK, SK) \rightarrow \langle\{M\}_{PK}\rangle_{SK} = M$.
T3. $\texttt{valid}(C) \rightarrow \texttt{valid}(\overline{\texttt{PK}}(C))$.
T4. $\overline{\texttt{I}}(C) = \overline{\texttt{S}}(C') \wedge \texttt{valid}(\overline{\texttt{PK}}(C')) \wedge h^*(\overline{\texttt{tbs}}(C)) = \langle\overline{\texttt{Sig}}(C)\rangle_{\overline{\texttt{PK}}(C')} \rightarrow \texttt{valid}(\overline{\texttt{Sig}}(C))$.
T5. $\texttt{valid}(\overline{\texttt{Sig}}(C)) \wedge today \geq \overline{\texttt{DS}}(C) \wedge today < \overline{\texttt{DE}}(C) \wedge \neg(C \in CRL_{\overline{\texttt{I}}(C)}) \rightarrow \texttt{valid}(C)$.

Axioms T4 and T5 are directly related to the validity of certificates. The signature part of a certificate C, i.e., the component $\overline{\texttt{Sig}}(C)$, is the digital signature over $\overline{\texttt{tbs}}(C)$.

This is a mechanism that binds the information and guar antees the integrity of the certificate. If any information is changed, then digital signature verification will fail. Digital signature generation involves two steps: hash and signature generation. First, a oneway, collisionfree hash function is applied to $\overline{\texttt{tbs}}(C)$. Then, the output of the hash function and the issuer's private key are provided as input of the signature algorithm. The signature value, i.e., $\overline{\texttt{Sig}}(C)$, is the output of the algorithm. Let h^* be the hash function, then information bound to the certificate C is unmodified if and only if $h^*(\overline{\texttt{tbs}}(C)) = \langle\overline{\texttt{Sig}}(C)\rangle_{\overline{\texttt{PK}}(C')}$, where we assume C' is the certificate held by the issuer of C. Thus, axioms T4 and T5 allow agents to verify the signature of a certificate as well as the certificate based on another certificate whose validity has been established.

We reuse axioms T1–T5 but revise T5 as follows:

T5′. $\texttt{valid}(\overline{\texttt{Sig}}(C)) \wedge val\_period(C,n) \wedge \neg(C \in CRL_{\overline{\texttt{I}}(C)}) \rightarrow \texttt{valid}(C)$.

where n is related to DS and DE, and $val\_period(C,n)$ is intended to represent the fact that (the certificate) C would be valid in the next n days (moments). Further we consider the following axiom schemata:

T6. $\mathbb{B}_{a_i} val\_period(C,n) \rightarrow (n > 1 \rightarrow \textbf{next}\ \mathbb{B}_{a_i}\ val\_period(C,n-1))$

T7. $\mathbb{B}_{a_i} val\_period(C,n) \rightarrow (\neg(n>1) \rightarrow \textbf{next}\ \neg\mathbb{B}_{a_i}\ val\_period(C,n-1))$

T8. $val\_period(C,n) \leftrightarrow (n > 0)$

Then the logic TML$^+$ together with the set of (trust) axioms $\{T1, T2, T3, T4, T5', T6, T7, T8\}$ will form a theory Γ_t. Having such a theory, we are able to reason about beliefs of agents related to specific moments in time.

We also make a few assumptions (supposing that they are formalised in Γ_a). For example, suppose that $charlie$ holds a certificate, c_1, and bob holds a certificate, c_2, which is signed by $charlie$ with his private key corresponding to the public key bound. Consider the case: c_1 is issued at May 5, 2003, and the expiry date is May 15, 2003. At a specific date (for example, May 10, 2003), $alice$ requires bob's certificate, he (always) trusts $charlie$ and he in particular believes that $charlie$'s certificate c_1 is valid, but at that moment, he is not sure whether c_2 is valid. Therefore, in order to use c_2, $alice$ must verify it. Let us take May 5, 2003 as the initial time (day 0) and also assume that $\mathbb{B}_{alice}\ \texttt{valid}(c1)$.

Then we would like to show that

$$\Gamma_t \cup \Gamma_a \vdash \textbf{first}\ \textbf{next}^{(5)}\ \mathbb{B}_{alice}\texttt{valid}(c_2).$$

Here $\textbf{next}^{(n)}$ denotes n applications of **next** and, when $n = 0$, $\textbf{next}^{(n)}p$ denotes p. We outline the proof (verification) procedure below.

Firstly, from the formula $\mathbb{B}_{alice}\ valid(c1)$ and by T3 and T4, if both $\overline{\texttt{I}}(C) = \overline{\texttt{S}}(C')$ and $h^*(\overline{\texttt{tbs}}(C)) = \langle\overline{\texttt{Sig}}(C)\rangle_{\overline{\texttt{PK}}(C')}$ are checked and hold, then we can have

(1) $\mathbb{B}_{alice}\ \texttt{valid}(\overline{\texttt{Sig}}(C))$, and

(2) $\textbf{first}\ \textbf{next}^{(5)}\ \mathbb{B}_{alice}\ \texttt{valid}(\overline{\texttt{Sig}}(C))$.

Secondly, if $\overline{\texttt{DS}}(C)$ and $\overline{\texttt{DE}}(C)$ are checked and correct, it would be true that

(3) $\textbf{first}\ \ \mathbb{B}_{alice}\ val\_period(c2, 10)$.

Thus, from (3) and by T6 and T7, we can have

(4) $\textbf{first}\ \textbf{next}^{(5)}\ \ \mathbb{B}_{alice}\ val\_period(c2, 5)$.

Thirdly, if $\neg(C \in CRL_{\overline{\texttt{I}}(C)})$ is checked and holds, then

(5) $\mathbb{B}_{alice}\ \neg(C \in CRL_{\overline{\texttt{I}}(C)})$, and hence

(6) $\textbf{first}\ \textbf{next}^{(5)}\ \ \mathbb{B}_{alice}\ \neg(C \in CRL_{\overline{\texttt{I}}(C)})$.

Thus, from (3), (5) and (7) and by T5', we finally obtain

(7) $\textbf{first}\ \textbf{next}^{(5)}\ \mathbb{B}_{alice}\ \texttt{valid}(c_2)$,

which is exactly what we want to show.

Note that from (4), we cannot derive the following formula by repeatedly using the rule T6:

(8) $\textbf{first}\ \textbf{next}^{(11)}\ \mathbb{B}_{alice}\ val\_period(c2, 0)$,

However, by rules T6 and T7, we may obtain

(9) $\textbf{first}\ \textbf{next}^{(11)}\ \neg\mathbb{B}_{alice}\ val\_period(c2, 0)$,

which in fact means that at day 11 it is not the case that $alice$ believes $val\_period(c2, 0)$ to be true, therefore, we cannot derive a statement that at day 11 $alice$ believes that bob's certificate c_2 is valid.

In order to increase the expressive power of TLC, we can introduce temporal modalities $\Box$ and $\Diamond$ as :

$$\Box\,\varphi = \textbf{first}\ \varphi \wedge \Box\,(\textbf{next}\ \varphi) \quad \text{and} \quad \Diamond\,\varphi = \neg\,\Box\,\neg\,\varphi$$

As usual, $\Box$ can be read as "always" and $\Diamond$ as "sometimes". Thus, we may have the formulas in the combined logic TML$^+$ as follows:

$$\Box\,\mathbb{B}_{bob}\ \forall M(sends(alice, M) \rightarrow reliable(M))$$
$$\Diamond\,\mathbb{B}_{bob}\ (sends(alice, M) \wedge reliable(M))$$

where the first formula says that it is always true that Bob believes all the messages sent by Alice to be reliable; and the second one says that sometime Bob believes that the message sent by Alice is reliable.

5. Hierarchy Combination of Logics

Combining logics is an emerging research area in logic and formal methods, with many potential applications. In simple terms, the problem of combining logics is: given two logics L_1 and L_2, how may we combine them into a single logic $L_1 + L_2$ that extends the expressive power of each. As an example, this paper showed how to obtain a new logic by adding TLC onto TML (resulting in TML^+). This kind of a combination technique (called temporalising) can be seen as a *hierarchy combination technique*.

In general, the hierarchy combination technique can be formalised as follows. We say that $(S, \succ)$ is a hierarchical structure if the partial order relation $\succ$ defined over the set S satisfies the following properties:

for any $X \in S, (X, X) \notin \succ$ (antireflexive)
for any $X, Y \in S$, if $(X, Y) \in \succ$, then $(Y, X) \notin \succ$ (antisymmetric); and
for any $X, Y, Z \in S$, if $(X, Y) \in \succ$ and $(Y, Z) \in \succ$, then $(X, Z) \in \succ$ (transitive).

We may write $(X, Y) \in \succ$ as $X \succ Y$ and read it as "X is a succession of Y".

Given a hierarchical structure $(\{L_1, \ldots, L_n\}, \succ)$ of modal logics $L_1, \ldots, L_n$, the hierarchy combination of these logics based on the structure is informally defined as a combination that satisfies the condition: For any $L_i, L_j (1 \leq i, j \leq n)$, if $L_i \succ L_j$, then the modal operators of L_i can be within the scope of the modal operators of L_j, but the modal operators of L_j are never within the scope of the modal operators of L_i. We denote the combination (the combined logic) by $(L_1 + \ldots + L_n)_{\succ}$.

Given two logics L_1 and L_2 , the hierarchical relation over $\{L_1, L_2\}$ can be $\{\}$ (the empty set), $L_1 \succ L_2$ or $L_2 \succ L_1$. So, there are normally three distinct hierarchy combined logics from the two logics. The first one, $(L_1 + L_2)_{\{\}}$, is the simplest combination of L_1 and L_2. Informally, we can view this combination as consisting of all formulas in the two languages and the boolean combinations of these formulas. The last two combined logics can be determined as in the example of plussing TLC onto TML, that is, $TML^+ = (TML + TLC)_{\{TML \succ TLC\}}$.

If we would like to be able express agent beliefs about temporal properties, which may be very natural for cer tain applications, TML$^+$ is not the logic we would use. For example, in TML$^+$, we can not express the asser tion that "Alice believes that at the initial time *bob* holds the key k." We would need to be able to write down $\mathbb{B}_{alice}$ **first** $holds(bob, k)$ which would be possible in $TLC^+ = (TML + TLC)_{\{TLC \succ TML\}}$.

Although it imposes certain restrictions on the combined logic, there are several advantages with the hierarchy combi nation technique. We have a standardised pattern for defining the syntax; the semantics of the combined logic is defined by the standard possibleworld semantics approach; and the proof system for any combined logic can be obtained from those of the two logics.

A critical issue in combining logics is what kind of a combination method is chosen to obtain a logic system that is suitable for a particular target application. As this paper shows by an example, logics obtained by the hierarchy com bination technique are generally strong (expressive) enough for use in the analysis of security properties.

Other operations (or combining techniques), which are re cently under investigation, includes the fusion of two mono modal logics [7], fibring technique [4], and the product of modal logics [5]. Fusion is the simplest method of combining two logics whereby the semantics of the two combined logics are placed side by side while product defines the notion of dimension in logics. Fibring is the most powerful combination technique. combination technique For further discussion and motivation on the combination of logics, we refer the reader to the literature [1].

6. Concluding Remarks

This paper defined a simple operation + applied to com bine a belief logic (TML) with a temporal logic (extended TLC). Investigating different techniques for combining logics and studying the properties of resulting logic systems will be covered in future work. With the understanding we gain from the combined logic systems, we could isolate and abstract away the properties that are most important for the application at hand, and, if and where needed, we could consider other and richer applicable logics (such as epistemic logics, deontic logics, fuzzy logics to name a few) to investigate how the combined systems could provide more expressive theories for describing, and reasoning about, security properties within trusted agentbased systems.

7. References

[1] A. CostaLeite. Towards a general theory of the combination of logics. In J.Y. Beziau, A. CostaLeite and A. Facchini. *Aspects of universal logic*, . Travaux de Logique [Works on Logic], Vol.17. Universite de Neuchatel, pp.219–230, 2004.

[2] G. Elofson. Developing trust with intelligent agent: An exploratory study. In *Proceedings of the first International Workshop on Trust*, pages 125–139, 1998.

[3] M. Finger and D. M. Gabbay. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information*, 1:221–237, 1997.

[4] D. Gabbay. *Fibring Logics*, volume 38 of *Oxford Logic Guides*. Oxford Univesity Press, 1998.

[5] D. M. Gabbay and V. Shehtman. Products of modal logics, part 1. *Logic Journal of the IGPL*, 6(1):71–146, 1998.

[6] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence* **54**, pages 319–379, 1992.

[7] M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *The Journal of Symbolic Logic*, 56(4):1469–1485, 1991.

[8] S.Ã. Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[9] C. Liu. Logical foundations for reasoning about trust in secure digital communication. In *AI2001: Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence Vol.2256*, pages 333–344. Springer Verlag, 2001.

[10] C. Liu and M. A. Orgun. Verification of reactive systems using temporal logic with clocks. *Theoretical Computer Science*, Vol.220(2) pp.377408, 1999.

[11] C. Liu, M. Ozols and M. Orgun. A temporalised belief logic for specifying the dynamics of trust for multiagent systems. In *Proceedings of the Ninth Asian Computer Science Conference, Lecture Notes in Computer Science Vol.3321*. SpringerVerlag, 2004.

[12] J. Ma and M. A. Orgun. Trust management and trust theory revision. *IEEE Transactions on Systems, Man and Cybernetics, Part A: System s and Humans*, Vol.36(3), pp.451–460, May 2006.

[13] P. V. Rangan. An Axiomatic basis of trust in distributed systems. *Proceedings of the IEEE Symposium on Security and Privacy*. p.205.

[14] D. R. Kuhn, V. C. Hu, W. T. Polk and S. Chang. Introduction to Public Key Technology and the Federal PKI Infrastructure. National Institute of Standards and Technology, 2001. U.S. Government publication.

[15] P. Smets, E. H. Mandami, D. Dubois, and H. Prade. *NonStandard Logics for Automated Reasoning.* Academic press, 1988.

[16] M. Wooldridge. Coherent social action. In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI94)* , pages 279–283, Amsterdam, The Netherlands, 1994.

[17] R. Yahalom, B. Klein, T. Beth. Trust Relationships in Secure Systems A Distributed Authentication Perspective. *Proceedings of the 1993 IEEE Symposium on Security and Privacy*, p.150, May 2426, 1993 .