

# Modal tableaux for verifying security protocols

Mehmet A. Orgun<sup>1</sup>, Guido Governatori<sup>2</sup> and Chuchang Liu<sup>3</sup>

<sup>1</sup> Department of Computing, Macquarie University, Sydney, NSW 2109, Australia

<sup>2</sup> School of ITEE, The University of Queensland, Brisbane, QLD 4072, Australia

<sup>3</sup> Information Networks Division, DSTO, Edinburgh, SA 5111, Australia

**Abstract.** To develop theories to specify and reason about various aspects of multi-agent systems, many researchers have proposed the use of modal logics such as belief logics, logics of knowledge, and logics of norms. As multi-agent systems operate in dynamic environments, there is also a need to model the evolution of multi-agent systems through time. In order to introduce a temporal dimension to a belief logic, we combine it with a linear-time temporal logic using a powerful technique called fibring for combining logics. We describe a labelled modal tableaux system for a fibred belief logic (FL) which can be used to automatically verify correctness of inter-agent stream authentication protocols. With the resulting fibred belief logic and its associated modal tableaux, one is able to build theories of trust for the description of, and reasoning about, multi-agent systems operating in dynamic environments.

## 1 Introduction

Multi-agent systems (MASs for short) consist of a collection of agents that interact with each other in dynamic and unpredictable environments. Agents communicate with one another by exchanging messages, and they have the ability to cooperate, coordinate and negotiate with each other to achieve their objectives. In order to develop theories to specify and reason about various aspects of multi-agent systems, many researchers have proposed the use of modal logics such as belief logics [4,6] and logics of knowledge [5,12]. As multi-agent systems operate in dynamic environments, there is also a need to model the evolution of multi-agent systems through time.

In order to introduce a temporal dimension to a belief logic, Liu *et al.* [16] have proposed a temporalized logic that provides a logical framework for users to specify the dynamics of trust and model evolving theories of trust for multi-agent systems. However, in this logic there are certain restrictions on the use of temporal and belief operators because of the hierarchical combination of belief and temporal logics used. Temporal operators can never be within the scope of a belief operator, hence we cannot express a statement asserting that some agent believes an event to happen at some time, e.g., the logic does not have a formula such as  $\mathbf{B}_{john} \mathbf{first} \mathit{holds}(bob, k)$ , which could be used to express an assertion that *John* believes that at the initial time *Bob* holds the key *k*. Such kind of assertions are often needed, for example, in analysing stream authentication protocols; we therefore consider a more powerful combination technique called fibring [8] that treats temporal operators and belief operators equally.

In this paper, we combine, using the fibring technique, the logic TML, a variant of the modal logic KD of belief [16], with the temporal logic SLTL which is suitable

for specifying events that may run on different clocks (time-lines) of varying rates of progress [15]. We show that in the resulting fibred belief logic (FL) we can specify and reason about not only agent beliefs but also the timing properties of a system effectively. We describe a labelled modal tableaux system for FL which can be used to automatically verify correctness of inter-agent stream authentication protocols. With this logical system one is able to build theories of trust for the description of, and reasoning about, multi-agent systems.

In the rest of the paper, Section 2 introduces the TESLA stream authentication protocol. Section 3 briefly discusses logics SLTL and TML. Section 4 presents the fibring technique as specifically applied for combining TML with SLTL, and provides an axiomatisation for the fibred logic called FL. Section 5 adapts KEM [2,9], a labelled modal tableaux system to reason with FL. Section 6 develops a theory of trust in FL for specifying the TESLA protocol and discusses its correctness.

## 2 The TESLA Protocol

Multi-agent systems, typically real world systems, need to employ application specific protocols for transferring data, such as video, audio and sensory data, among agents. Such protocols are often different from the standard class of authentication protocols previously analysed by many researchers using belief logics and/or model checking techniques [4,5,6]. As an example, we consider the TESLA protocol, a multicast stream authentication protocol of Perrig *et al.* [19]. In TESLA, authentication is based on the timing of the publication of keys and the indirect relation of each new key to an original key commitment. The process for verifying data packets received to be authentic depends on trust of the receiver in the sender, and belief on whether an intruder can have prior knowledge of a key before it is published by the protocol.

We consider a basic scheme for the TESLA Protocol, called the PCTS scheme, in which each message  $M_i$  is sent in a packet  $P_i$ , along with additional authentication information [3,19]. The sender issues a signed commitment to a key. The key is only known to the sender. To send message  $M_i$ , the sender uses that key to compute a MAC (Message Authenticating Code) on a packet  $P_i$ , and later discloses the key in packet  $P_{i+1}$ , which enables the receiver (or receivers, when multiple receivers are involved) to verify the commitment and the MAC of packet  $P_i$ . A successful verification will imply that packet  $P_i$  is authenticated and trusted. We assume that, apart from the initial contact messages between the sender and the receiver, for all  $i \geq 2$ , the packet  $P_i$  from the sender to receiver has the standard form  $\langle D_i, MAC(K'_i, D_i) \rangle$ , where  $D_i = \langle M_i, f(K_{i+1}), K_{i-1} \rangle$ ,  $K'_j = f'(K_j)$  for  $j \geq 1$ , and  $f$  and  $f'$  are two different pseudo-random functions.

In analysing the protocol it is assumed that [19]:

- The sender is honest and works correctly, following all requirements of the protocol strictly.
- The receiver accepts packet  $P_i$  as authentic only when it believes the key commitment and the MAC of the packet have been successfully verified.
- The intruder has the ability to capture, drop, resend, delay, and alter packets, can access to a fast network with negligible delay, and can perform efficient computations, such as computing a reasonable number of pseudo-random function appli-

cations and MACs with negligible delay. Nonetheless, the intruder cannot invert a pseudo-random function with non-negligible probability.

The security property for the TESLA protocol we need to guarantee is that the receiver does not believe any packet  $P_i$  to be authenticated unless the  $M_i$  it contains was actually sent by the sender. To prevent any successful attack by an intruder, the receiver only needs to be sure that all packets  $P_i$  arrive safely such that the intruder has no time to change the message and commitment in  $P_i$  and forge the subsequent traffic.

### 3 Two Logics: SLTL and TML

We now give a brief introduction to the logics SLTL and TML.

#### 3.1 SLTL: Simple Linear-time Temporal Logic

SLTL offers two operators, **first** and **next**, which refer to the initial moment and the next moment in time respectively. The formulas of SLTL are built with the usual formation rules from standard connectives and quantifiers of classical first order logic, and the temporal operators **first** and **next**.

The collection of moments in time is the set of natural numbers. We define the global clock as the increasing sequence of natural numbers, i.e.,  $\langle 0, 1, 2, \dots \rangle$ , and a local clock is an infinite subsequence of the global clock. Thus, we have

**Definition 1 (time models)** *A time model for the logic SLTL has the form  $\mathbf{c} = \langle C, <, \nu \rangle$ , where  $C = \langle t_0, t_1, t_2, \dots \rangle$  is a clock,  $<$  is the usual ordering relation over  $C$  and  $\nu$  is an assignment function giving a value  $\nu(t, q) \in \{true, false\}$  for any atomic formula  $q$  at time  $t$  in  $C$ .*

We write  $\mathbf{c}, t \models A$  to stand for “ $A$  is true at time  $t$  in the model  $\mathbf{c}$ ”. Then the semantics of the temporal operators with the notion of satisfaction in SLTL is given as follows:

- $\mathbf{c}, t_i \models \mathbf{first} A$  iff  $\mathbf{c}, t_0 \models A$ .
- $\mathbf{c}, t_i \models \mathbf{next} A$  iff  $\mathbf{c}, t_{i+1} \models A$ .
- satisfaction in the model  $\mathbf{c} = \langle C, <, \nu \rangle$  is defined as satisfaction at some point on  $C$ .

A minimal axiomatic system for the propositional temporal logic consists of the following axioms (axiom schemata). We let  $\nabla$  stand for **first** or **next**.

- A0. all classical tautologies.
- A1.  $\nabla(\mathbf{first} A) \leftrightarrow \mathbf{first} A$ .
- A2.  $\nabla(\neg A) \leftrightarrow \neg(\nabla A)$ .
- A3.  $\nabla(A \wedge B) \leftrightarrow (\nabla A) \wedge (\nabla B)$ .

Apart from the generic substitution rule, SLTL has two rules of inference defined as follows:

- MP. From  $\vdash A$  and  $\vdash A \rightarrow B$  infer  $\vdash B$  (Modus Ponens)
- TG. From  $\vdash A$  infer  $\vdash \nabla A$  (Temporal Generalisation)

The soundness and completeness of the axiomatisation system for SLTL with respect to the class  $\mathcal{C}$  consisting of all local clocks are straightforward [15].

### 3.2 TML: Typed Modal Logic

We assume that there are  $n$  agents  $a_1, \dots, a_n$  and, correspondingly,  $n$  modal operators  $\mathbf{B}_1, \dots, \mathbf{B}_n$  in the logic, where  $\mathbf{B}_i$  ( $1 \leq i \leq n$ ) stands for “agent  $a_i$  believes that”.

We assume the fixed-domain approach to quantification, that is, the domain of quantification is the same in all possible worlds. This means that we have the standard first-order logic semantics for the  $\forall$  and  $\exists$  quantifiers. We also employ rigid denotations for terms, that is, the only dynamic objects are predicates. We also assume that in TML all the wffs built according to the usual formation rules are correctly typed.

A *classical Kripke model* [14] for the logic TML is a tuple  $\mathbf{m} = \langle S, R_1, \dots, R_n, \pi \rangle$ , where  $S$  is the set of states or possible worlds; and each  $R_i$  ( $1 \leq i \leq n$ ) is a relation over  $S$ , consisting of state pairs  $(s, t)$  such that  $(s, t) \in R_i$  iff, at state  $s$ , agent  $a_i$  considers the state  $t$  possible; and  $\pi$  is the *assignment function*, which gives a value  $\pi(s, q) \in \{\text{true}, \text{false}\}$  for any  $s \in S$  and atomic formula  $q$ . Each  $R_i$  called the *possibility relation* according to agent  $a_i$ . We write  $\mathbf{m}, s \models \varphi$  to stand for “ $\varphi$  is true at the state  $s$  in the model  $\mathbf{m}$ ” or “ $\varphi$  holds at  $s$  in  $\mathbf{m}$ ”. The semantics definition for the belief operators with the notion of satisfaction in TML is given as follows:

- $\mathbf{m}, s \models \mathbf{B}_i \varphi$  iff, for all  $t$  such that  $(s, t) \in R_i$ ,  $\mathbf{m}, t \models \varphi$ .
- A formula  $\varphi$  is satisfiable in a model  $\mathbf{m}$  if there exists  $s \in S$  such that  $\mathbf{m}, s \models \varphi$ .

In preparation for fibring TML with SLTL, we now consider monadic models for TML defined as follows:

**Definition 2 (monadic models)** *A monadic model for TML is a structure  $\mathbf{m} = \langle S, R_1, \dots, R_n, \pi, u \rangle$  where  $\langle S, R_1, \dots, R_n, \pi \rangle$  is a classical model for TML and  $u \in S$  is called the actual world.  $\varphi$  is satisfiable in the monadic model  $\mathbf{m}$  if and only if  $\mathbf{m}, u \models \varphi$ .*

We define  $\mathcal{K}_{tml}$  as a class of monadic models of the form  $\langle S, R_1, \dots, R_n, \pi, u \rangle$ , where

$$(1) S = \{x \mid \exists R_{i_1} \dots R_{i_k} u R_{i_1} \circ \dots \circ R_{i_k} x, R_{i_1}, \dots, R_{i_k} \in \{R_1, \dots, R_n\}\},$$

where  $R_i \circ R_j$  represents the relative product (or composition) of  $R_i$  and  $R_j$ . Furthermore, using the notation  $\mathbf{m}$  for a model in  $\mathcal{K}_{tml}$ , we write  $\mathbf{m} = \langle S^{(\mathbf{m})}, R_1^{(\mathbf{m})}, \dots, R_n^{(\mathbf{m})}, \pi^{(\mathbf{m})}, u^{(\mathbf{m})} \rangle$ . In addition we assume P:

- (2) if  $\mathbf{m}_1 \neq \mathbf{m}_2$ , then  $S^{(\mathbf{m}_1)} \cap S^{(\mathbf{m}_2)} = \emptyset$ .
- (3)  $\mathbf{m}_1 = \mathbf{m}_2$  iff  $u^{(\mathbf{m}_1)} = u^{(\mathbf{m}_2)}$ .

Assumption (2) indicates that all sets of possible worlds in  $\mathcal{K}_{tml}$  are all pairwise disjoint, and that there are infinitely many isomorphic (but disjoint) copies of each model; assumption (3) means that a model in  $\mathcal{K}_{tml}$  can in fact be identified by the actual world in it.

TML has the following axiom schemata and inference rules:

- B0. all axioms of the classical first-order logic.
- B1.  $\mathbf{B}_i(\varphi \rightarrow \psi) \wedge \mathbf{B}_i \varphi \rightarrow \mathbf{B}_i \psi$  for all  $i$  ( $1 \leq i \leq n$ ).
- B2.  $\mathbf{B}_i(\neg \varphi) \rightarrow \neg(\mathbf{B}_i \varphi)$  for all  $i$  ( $1 \leq i \leq n$ ).

- B3.  $\forall X \mathbf{B}_i \varphi(X) \rightarrow \mathbf{B}_i \forall X \varphi(X)$  for all  $i$  ( $1 \leq i \leq n$ ).
- I1. From  $\varphi$  and  $\varphi \rightarrow \psi$  infer  $\psi$ . (Modus Ponens)
- I2. From  $\forall X \varphi(X)$  infer  $\varphi(Y)$ . (Instantiation)
- I3. From  $\varphi(X)$  infer  $\forall X \varphi(X)$ . (Generalisation)
- I4. From  $\varphi$  infer  $\mathbf{B}_i \varphi$  for all  $i$  ( $1 \leq i \leq n$ ). (Necessitation)

The soundness and completeness of the axiomatisation system for TML can be proved in a standard pattern [13].

## 4 FL: Fibred Logic

In this section, we discuss how the logic FL is obtained through the use of fibring technique for combining the logics TML and SLTL. Let  $\mathcal{O} = \{\mathbf{B}_1, \dots, \mathbf{B}_n, \mathbf{first}, \mathbf{next}\}$  be the set of modal connectives of FL. Then the formulas of FL are obtained from the usual formation rules. As before we assume that in FL all the wffs are correctly typed.

The discussion of the fibred semantics in the case of the Kripke monadic models for TML with time models for SLTL 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 we restrict ourselves to the first level. Following Gabbay [8], we define the fibred semantics arising from the Kripke models for TML with a single time model based on *simplified fibred models* (simply, *sfm models*) defined as follows:

**Definition 3 (sfm models)** A simplified fibred model or sfm model is a tuple  $\langle W, W_t, W_b, R_0, R_1, \dots, R_n, \pi, \mathbf{F}, w_0 \rangle$  where

1.  $W$  is a set of worlds,  $w_0 \in W_t \cup W_b$
2.  $W_b \subseteq W$ , and  $W_t$  is a set of natural numbers, we also have  $W_t \subseteq W$ .
3. For  $s \in W_b$ , let  $S^{(s)} = \{x \mid sR_{i_1} \circ \dots \circ R_{i_k} x, \text{ for some } R_{i_1}, \dots, R_{i_k} \in \{R_1, \dots, R_n\}\}$ , then (1) for all  $s \in W_b$ ,  $S^{(s)} \cap W_t = \emptyset$ ; (2) for all  $s, r \in W_b$ , if  $s \neq r$ , then  $S^{(s)} \cap S^{(r)} = \emptyset$ ; and (3)  $W = (\bigcup_{s \in W_b} S^{(s)}) \cup W_t$ .
4.  $R_0 = \{(x, y) \mid x, y \in W_t \text{ \& } x < y, \text{ for all } x, y \in W_t\}$ .
5. For all  $u \in W_t$ , the model  $\mathbf{c} = (C, R_0, \pi^{(c)})$  satisfies the condition that  $u \in W_t$  iff  $u$  is a time point in the clock  $C$ , and is in the semantics of SLTL.
6. For all  $u \in W_b$ , the model  $\mathbf{m}^{(u)} = (S^{(u)}, R_1 \upharpoonright S^{(u)} \times S^{(u)}, \dots, R_n \upharpoonright S^{(u)} \times S^{(u)}, u, h \upharpoonright S^{(u)})$  is in the semantics of  $\mathcal{H}_{iml}$  of the logic TML.
7.  $\mathbf{F}$  is the fibred function consisting of two folds,  $\mathbf{F}_b$  and  $\mathbf{F}_t$ . It satisfies the following conditions: (1) For all connectives  $\nabla \in \mathcal{O}$  and all worlds  $w \in W$ ,

$$\mathbf{F}(\nabla, w) = \begin{cases} \mathbf{F}_t(w) & \text{if } \nabla \text{ is first or next} \\ \mathbf{F}_b(w), & \text{otherwise.} \end{cases}$$

- (2) If  $x \in S^{(u)}$  and  $u \in W_b$ , then  $\mathbf{F}_b(x) = x$ ; if  $x \in W_t$ , then  $\mathbf{F}_b(x) \in W_b$ ; if  $x \in W_t$ , then  $\mathbf{F}_t(x) = x$ ; and if  $x \notin W_t$ , then  $\mathbf{F}_t(x) \in W_t$ .

**Definition 4 (semantics)** The semantics of formulas for the logic FL is defined inductively with respect to an sfm-model  $\langle W, W_c, W_b, R_0, R_1, \dots, R_n, \pi, \mathbf{F}, w_0 \rangle$ . For any  $w \in W$ ,

1. for any atomic formula  $q$ ,  $w \models q$  iff  $\pi(w, q) = \text{true}$ .
2.  $w \models \neg\varphi$  iff it is not the case that  $w \models \varphi$ .
3.  $w \models (\varphi \wedge \psi)$  iff  $w \models \varphi$  and  $w \models \psi$ .
4.  $w \models \forall X\varphi(X)$  iff, for all  $d \in \mathcal{T}$ ,  $w \models \varphi(d)$ , where  $\mathcal{T}$  is the type of  $X$ .
5.  $w \models \nabla\varphi$  iff  $\mathbf{F}(\nabla, w) \models \nabla\varphi$ .
6.  $w \models \mathbf{first}\ \varphi$  when  $w \in W_t$  iff  $\min\{t \mid t \in W_t\} \models \varphi$ .
7.  $w \models \mathbf{next}\ \varphi$  when  $w \in W_t$  iff  $\min\{t \mid wR_0t\} \models \varphi$ .
8.  $w \models \mathbf{B}_i\varphi$  when  $w \notin W_t$  and  $1 \leq i \leq n$  iff, for all  $s$  such that  $wR_is$ ,  $s \models \varphi$ , assuming  $s \in S^{(\mathbf{m})}$  and  $\mathbf{m} \in \mathcal{K}_{bl}$ .

With the sfm model  $\langle W, W_c, W_b, R_0, R_1, \dots, R_n, \pi, \mathbf{F}, w_0 \rangle$  we say that it satisfies the formula  $\varphi$  iff  $w_0 \models \varphi$ . Furthermore,  $\mathbf{m} = \langle W, W_c, W_b, R_0, R_1, \dots, R_n, \pi, \mathbf{F} \rangle$  is called a *regular fibred semantics model* for the logic FL. We say  $\varphi$  is valid in the model  $\mathbf{m}$ , and written as  $\mathbf{m} \models \varphi$ , if, for all  $w_0 \in W_t \cup W_b$ , the model  $\langle W, W_c, W_b, R_0, R_1, \dots, R_n, \pi, \mathbf{F}, w_0 \rangle$  satisfies  $\varphi$ ; we say that  $\varphi$  is satisfied in the model  $\mathbf{m}$  if, for some  $w_0 \in W_t \cup W_b$ , the model  $\langle W, W_c, W_b, R_0, R_1, \dots, R_n, \pi, \mathbf{F}, w_0 \rangle$  satisfies  $\varphi$ . Let  $\mathcal{K}_{fl}$  be the set of regular fibred semantics models which defines the fibred logic FL, then we say  $\varphi$  is valid in the logic FL if, for all  $\mathbf{m} \in \mathcal{K}_{fl}$ ,  $\mathbf{m} \models \varphi$ .

The axiom set of FL consists of the combination of the axioms for SLTL and TML and their inference rules. The soundness for the logic FL depends on the soundness theorems for logics TML and SLTL, and is not difficult to prove; the completeness can be proved by the techniques used in Gabbay [8].

## 5 Labelled Tableaux for FL

In this section we show how to adapt KEM, a labelled modal tableaux system, to reason with FL. The system can be used to automatically check for formal properties of security protocols, in particular for TESLA, in FL.

A tableaux system is a semantic based refutation method that systematically tries to build a (counter-)model for a set of formulas. A failed attempt to refute (invalidate) a set of formulas generates a model where the set of formulas is true. To show that a property  $A$  follows from a theory (a protocol)  $B_1, \dots, B_n$  we verify whether a model for  $\{B_1, \dots, B_n, \neg A\}$  exists. If it does not then  $A$  is a consequence of the protocol.

In labelled tableaux systems, the object language is supplemented by labels meant to represent semantic structures (possible worlds in the case of modal and temporal logics). Thus the formulas of a labelled tableaux system are expressions of the form  $A : i$ , where  $A$  is a formula of the logic and  $i$  is a label. The intuitive interpretation of  $A : i$  is that  $A$  is true at (the possible world(s) denoted by)  $i$ .

KEM is a labelled tableaux for logics admitting possible world semantics whose inferential engine is based on a combination of standard tableaux linear expansion rules and natural deduction rules supplemented by an analytic version of the cut rule. In addition it utilises a sophisticated but powerful label formalism that enables the logic to deal with a large class of (quantified) modal and non-classical logics. Furthermore the label mechanism corresponds to fibring thus it is possible to define tableaux systems for multi-modal logic by a seamless combination of the (sub)tableaux systems for the component logics of the combination.

## 5.1 Label Formalism

KEM uses *Labelled Formulas* ( $L$ -formulas for short), where an  $L$ -formula is an expression of the form  $A : i$ , where  $A$  is a wff of the logic, and  $i$  is a label. For FL we have a type of labels to various modalities for each agent (belief) plus a type of labels for the temporal modalities. The set of atomic labels is

$$\Phi = \Phi_T \cup \bigcup_{i \in \text{Agt}} \Phi^i,$$

where  $\Phi_T = \{t_0, t_1, \dots\}$  and every  $\Phi^i$  is partitioned into (non-empty) sets of variables and constants:  $\Phi^i = \Phi_V^i \cup \Phi_C^i$  where  $\Phi_V^i = \{W_1^i, W_2^i, \dots\}$  and  $\Phi_C^i = \{w_1^i, w_2^i, \dots\}$ . Finally we add a sets of auxiliary unindexed atomic labels  $\Phi^A = \Phi_V^A \cup \Phi_C^A$  where  $\Phi_V^A = \{W_1, W_2, \dots\}$  and  $\Phi_C^A = \{w_1, w_2, \dots\}$ .  $\Phi^A$  will be used in unifications and proofs.  $\Phi_C$  and  $\Phi_V$  denote the set of constants and the set of variables. The set of labels is denoted by  $\mathfrak{S}$ .

**Definition 5 (labels)** A label is either a (i) an element of the set  $\Phi_C$ , or (ii) an element of the set  $\Phi_V$ , or (iii) a path term  $(u', u)$  where (iiia)  $u' \in \Phi_C \cup \Phi_V$  and (iiib)  $u \in \Phi_C$  or  $u = (v', v)$  where  $(v', v)$  is a label.

As an intuitive explanation, we may think of a label  $u \in \Phi_C$  as denoting a world (a *given* one), and a label  $u \in \Phi_V$  as denoting a set of worlds (*any* world) in some Kripke model. A label  $u = (v', v)$  may be viewed as representing a path from  $v$  to a (set of) world(s)  $v'$  accessible from  $v$  (the world(s) denoted by  $v$ ).

For any label  $u = (v', v)$  we shall call  $v'$  the *head* of  $u$ ,  $v$  the *body* of  $u$ , and denote them by  $h(u)$  and  $b(u)$  respectively. Notice that these notions are recursive (they correspond to projection functions): if  $b(u)$  denotes the body of  $u$ , then  $b(b(u))$  will denote the body of  $b(u)$ , and so on. We call each of  $b(u)$ ,  $b(b(u))$ , etc., a *segment* of  $u$ . The length of a label  $u$ ,  $\ell(u)$ , is the number of world-symbols in it.  $s^n(u)$  will denote the segment of  $u$  of length  $n$  and we shall use  $h^n(u)$  as an abbreviations for  $h(s^n(u))$ . Notice that  $h(u) = h^{\ell(u)}(u)$ . Let  $u$  be a label and  $u'$  an atomic label. We use  $(u'; u)$  as a notation for the label  $(u', u)$  if  $u' \neq h(u)$ , or for  $u$  otherwise.

For any label  $u$ ,  $\ell(u) > n$ , we define the *counter-segment- $n$*  of  $u$ , as follows (for  $n < k < \ell(u)$ ):

$$c^n(u) = h(u) \times (\dots \times (h^k(u) \times (\dots \times (h^{n+1}(u), w_0))))$$

where  $w_0$  is a dummy label, i.e., a label not appearing in  $u$  (the context in which such a notion occurs will tell us what  $w_0$  stands for). The counter-segment- $n$  defines what remains of a given label after having identified the segment of length  $n$  with a 'dummy' label  $w_0$ . The appropriate dummy label will be specified in the applications where such a notion is used. However, it can be viewed also as an independent atomic label in the set of auxiliary labels.

So far we have provided definitions about the structure of the labels without regard of the elements they are made of. The following definitions will be concerned with the type of world symbols occurring in a label.

We say that a label  $u$  is  $\tau$ -preferred iff  $\tau = i$  and  $h(u) \in \Phi^i$ , or  $\tau = t$  and  $h(u) \in \Phi_T$ ; a label  $u$  is  $\tau$ -pure iff each segment of  $u$  of length  $n > 1$  is  $\tau$ -preferred. With  $\mathfrak{S}^i$  we denote the set of  $i$ -preferred labels where  $i \in \text{Agt}$ .

## 5.2 Label Unifications

One of the key features of KEM is its logic dependent label unification mechanism. In the same way as each modal logic is characterised by a combination of modal axioms (or semantic conditions on the model), KEM defines a unification for each modality and axiom/ semantic condition and then combines them in a recursive and modular way. In this case for SLTL we have to provide a characterisation of the two modalities **first** and **next** in terms of relationships over labels. In particular we use what we call unification to determine whether the denotation of two labels have a non empty intersection, or in other terms whether two labels can be mapped to the same possible world in the possible worlds semantics.

The second key issue is the ability to split labels and to work with parts of labels. The mechanism permits the encapsulation of operations on sub-labels. This is an important feature that, in the present context, allows us to correlate unifications and fibring functions. Given the modularity of the approach the first step of the construction is to define unifications (pattern matching for labels) corresponding to the single modality in the logic we want to study.

Every unification is built from a basic unification defined in terms of a substitution  $\rho : \Phi \mapsto \mathfrak{S}$  such that:

$$\begin{aligned} \rho &: \mathbf{1}_{\Phi_C} \\ \Phi_V^i &\mapsto \mathfrak{S}^i \text{ for every } i \in \text{Agt} \\ \Phi_V^A &\mapsto \mathfrak{S} \end{aligned}$$

This means that a substitution  $\rho$  replaces a constant with the same constant; a variable of type  $i$  can be replaced by any  $i$ -preferred label, while an auxiliary variable can be freely replaced by any label. This is in agreement with the intuitive meaning of labels that a constant stands for a possible world, and a variable stands for a set of possible world (of the appropriate type). Accordingly, we have that two atomic (“world”) labels  $u$  and  $v$   $\sigma$ -unify iff there is a substitution  $\rho$  such that  $\rho(u) = \rho(v)$ . We shall use  $[u;v]\sigma$  both to indicate that there is a substitution  $\rho$  for  $u$  and  $v$ , and the result of the substitution. The  $\sigma$ -unification is extended to the case of composite labels (path labels) as follows:

$$[u;v]\sigma = z \text{ iff } \exists \rho : h(z) = \rho(h(u)) = \rho(h(v)) \text{ and } b(z) = [b(u);b(v)]\sigma$$

Clearly  $\sigma$  is symmetric, i.e.,  $[u;v]\sigma$  iff  $[v;u]\sigma$ . Moreover this definition offers a flexible and powerful mechanism: it allows for an independent computation of the elements of the result of the unification, and variables can be freely renamed without affecting the result of a unification.

We are now ready to introduce the unifications corresponding to the modal operators at hand. For these unification we assume that the labels involved are  $\tau$ -pure. The first unification is that for **first**.

$$[u;v]\sigma^{\text{first}} = (t_0; [h^1(u);h^1(v)]\sigma) \text{ iff } h(u) = h(v) = t_0 \text{ and } [h^1(u);h^1(v)]\sigma$$

The unification for **first** ( $\sigma^{\text{first}}$ -unification) corresponds to a constant function (the initial time is unique for the model). Accordingly if two labels end with the same atomic label ( $t_0$ ) then the two labels denote the same time instant, namely the start of the clock.

For the unification for **next** we will use the fact that the time line is a discrete total order, thus two labels denote the same time instant if they have the same length.

$$[u; v]\sigma^{\text{next}} = u \text{ iff } \ell(u) = \ell(v), [h^1(u); h^1(v)]\sigma \text{ and } c^1(u), c^1(v) \text{ do not contain } t_0.$$

The unification for the logic SLTL is defined by the combination of the unifications for **first** and **next**. Formally

$$[u; v]\sigma_{SLTL} = \begin{cases} [u; v]\sigma^{\text{first}} \\ [c^n(u); c^m(v)]\sigma^{\text{next}}, \quad w_0 = [s^n(u); s^m(v)]\sigma_{SLTL} \end{cases}$$

The belief logic can be understood as the combination of multiple **KD** modal logics, one for each agent  $i \in \text{Agt}$ . Thus we first give the unification for each of such logics and then we combine in a single unification to be used with the unification for SLTL for FL.

$$[u; v]\sigma^{TML_i} = [u; v]\sigma$$

where  $u$  and  $v$  are  $i$ -pure. Notice that using the mechanism of counter-segment it is always possible to split labels into pure sub-labels. Accordingly the definition of the unification for TML is

$$[u; v]\sigma_{TML} = \begin{cases} [u; v]\sigma^{TML_i} & u, v \text{ are } i\text{-pure, or} \\ [c^n(u); c^m(v)]\sigma^{TML_i} & c^n(u), c^m(v) \text{ are } i\text{-pure, and} \\ & w_0 = [s^n(u); s^m(v)]\sigma_{TML}. \end{cases}$$

The logic FL is the fibred combination of TML and SLTL, thus according to [9] we can obtain the unification for it based on the unifications for the component logics. With  $\sigma^{TBL}$  we understood either  $\sigma_{SLTL}$  or  $\sigma_{TML}$ . The unification for FL is:

$$[u; v]\sigma_{FL} = \begin{cases} [u; v]\sigma^{TBL} \\ [c^n(u); c^m(v)]\sigma^{TBL} & c^n(u), c^m(v) \text{ are } i\text{-pure, and} \\ & w_0 = s^n(u)s^m(v) \end{cases}$$

**Theorem 1** *The  $\sigma_{FL}$ -unification of two labels  $u$  and  $v$  can be computed in linear time.*

### 5.3 Inference Rules

For the presentation of the inference rules we assume familiarity with Smullyan-Fitting unifying notation [7].

$$\begin{array}{c} \frac{\alpha : u}{\alpha_1 : u} \quad \frac{A \wedge B : u}{A : u} \quad \frac{\neg(A \vee B) : u}{\neg A : u} \quad \frac{\neg(A \rightarrow B) : u}{A : u} \\ \alpha_2 : u \quad B : u \quad \neg B : u \quad \neg B : u \end{array} \quad (\alpha)$$

The  $\alpha$ -rules are just the familiar linear branch-expansion rules of the tableau method. For the  $\beta$ -rules (formulas behaving disjunctively) we exemplify only the rules for  $\rightarrow$ .

$$\begin{array}{c} \frac{\beta : u}{\beta_i^c : v} \quad (i = 1, 2) \quad \frac{A \rightarrow B : u}{A : v} \quad \frac{A \rightarrow B : u}{\neg B : v} \\ \frac{\beta_{3-i} : [u; v]\sigma_{FL}}{B : [u; v]\sigma_{FL}} \quad \frac{A \rightarrow B : u}{\neg A : [u; v]\sigma_{FL}} \end{array} \quad (\beta)$$

The  $\beta$ -rules are nothing but natural inference patterns such as Modus Ponens, Modus Tollens and Disjunctive syllogism generalised to the modal case. In order to apply such rules it is required that the labels of the premises unify and the label of the conclusion is the result of their unification.

$$\frac{\gamma : u}{\gamma_0(x_n) : u} \quad \frac{\forall xP(x) : u}{P(x_n) : u} \quad \frac{\neg\exists xP(x) : u}{\neg P(x_n) : u} \quad (\gamma)$$

The  $\gamma$  rules are the usual “universal” rules of tableaux method with the usual proviso that  $x_n$  is a variable not previously occurring in the tree [7,2].

$$\frac{\delta : u}{\delta_0(c_n) : u} \quad \frac{\exists xP(x) : u}{P(c_n) : u} \quad \frac{\neg\forall xP(x) : u}{\neg P(c_n) : u} \quad (\delta)$$

The  $\delta$  rules are the usual “existential” rules of the tableau method, where  $c_n$  is a constant that does not occur previously in the tree.

$$\frac{\mathbf{B}_iA : u}{A : (W_n^i, u)} \vee \quad \frac{\neg\mathbf{B}_iA : u}{\neg A : (w^i, u)} \pi \quad (\mu_B)$$

where  $W_n^i$  and  $w_n^i$  are new labels.

The rules for  $\mathbf{B}$  are the normal expansion rule for modal operators of labelled tableaux with free variable. The intuition for the  $\vee$  rule is that if  $\mathbf{B}A$  is true at  $u$ , then  $A$  is true at all worlds accessible via  $R_i$  from  $u$ , and this is the interpretation of the label  $(W_n^i, u)$ ; similarly if  $\mathbf{B}A$  is false at  $u$  (i.e.,  $\neg\mathbf{B}A$  is true), then there must be a world, let us say  $w_n^i$  accessible from  $u$ , where  $\neg A$  is true.

$$\frac{\mathbf{first}A : u}{A : (t_0, u)} \quad \frac{\neg\mathbf{first}A : u}{\neg A : (t_0, u)} \quad \frac{\mathbf{next}A : u}{A : (t_n, u)} \quad \frac{\neg\mathbf{next}A : u}{\neg A : (t_n, u)} \quad (\mu_T)$$

where  $t_n$  is new.

Given the functional interpretation of the temporal accessibility relation and that the initial instant is fixed, we have the same expansion of the labels and there is no need to introduce variables.

$$\frac{}{A : u \quad | \quad \neg A : u} \quad (PB)$$

The “Principle of Bivalence” represents the semantic counterpart of the cut rule of the sequent calculus (intuitive meaning: a formula  $A$  is either true or false in any given world). PB is a zero-premise inference rule, so in its unrestricted version can be applied whenever we like. However, we impose a restriction on its application. PB can be only applied w.r.t. immediate sub-formulas of unanalysed  $\beta$ -formulas, that is  $\beta$  formulas for which we have no immediate sub-formulas with the appropriate labels in the tree.

$$\frac{A(x) : u \quad \neg A(y) : v}{\times} [ \text{if } [u; v] \sigma_{FL} \text{ and } x \text{ and } y \text{ unify} ] \quad (PNC)$$

The rule PNC (*Principle of Non-Contradiction*) states that two labelled formulas are  $\sigma_{FL}$ -complementary when the two formulas are complementary (i.e., the terms in the formula unify according to the standard unification for terms) and their labels  $\sigma_{FL}$ -unify.

## 5.4 Soundness and Completeness

The resulting tableaux system is sound and complete for the logics presented in this paper. As usual with tableaux systems a proof of  $A$  is a closed tableaux for  $\neg A$ , thus a tableaux systems is sound and complete for a particular logic if it is able to generate closed tableaux for all negation of valid formula, and open tableaux (models) for all satisfiable formulas. In proving the results for the logics at hand we will make use of the main result of [9] (Theorem 22) that allows one to obtain a sound and complete labelled tableaux system for a fibred logic based on sound and complete labelled tableaux systems (of the same type of the tableaux system for the fibred logic) for the logics to be combined. The key idea of the Theorem is to conceive the join point of a unification where the labels are split in segments and counter-segments as the counterpart of the fibring function in fibred models.

**Theorem 2** *KEM is sound and complete for SLTL, TML and FL.*

## 6 Analysing Authentication Protocols

In this section, we first build a theory of trust to specify the TESLA protocol, then discuss its correctness. With the purpose of making the logic FL appropriate for specifying the protocol, we restrict the time model of FL to guarantee that the time interval between any moment and its next moment in time has the same length, 1 unit time. This restriction matches the special timing property that the TESLA scheme satisfies: *the sender sends packets at regular time intervals*. The assumption simplifies our discussion without harming its correctness.

### 6.1 The Formalization of TESLA

We now establish a theory that describes the behaviour or functions of the protocol with the scheme PCTS. The basic types of the theory include:  $Agents = \{A, B, S, R, I\}$ ,  $Messages = \{X, Y, D, D'\}$  and  $Keys = \{K, K_1, K_2\}$  where  $S, R, I$  stand for the sender, the receiver, and the intruder, respectively. In case there are multiple receivers, we may have  $R_1, R_2, \dots$  in the type  $Agents$ .

Through an analysis of the TESLA protocol, we set a theory to specify it consisting of four modules,  $M_{sr}$  (*send-receive mode specification*),  $M_{mk}$  (*message receiving and knowledge gained*),  $M_{ms}$  (*message sending*), and  $M_{ar}$  (*authentication rules*). Each module consists of several axiom schemata). Several predicates are used to express these axioms. Given the intuitive reading of the predicates we omit their explanations.

Send-receive mode specification depends on what kind of mode is adopted. We first consider a simple mode called the *zero-delay mode*, which is based on two assumptions: (1) Zero time is spent between sending a message and receiving this message, i.e., the sending time of a packet  $P_i$  is equal to the receiving time of the packet on the synchronized receiver's clock, for any  $P_i$ ; and (2) the packet rate is assumed to be 1 (i.e., 1 packet per unit time). With this mode, module  $M_{sr}$  consists of the following axiom schemata:

- Z1.  $sends(S, R, X) \leftrightarrow receives(R, X)$ .  
 Z2.  $sends(S, R, \langle D, MAC(f'(K), D) \rangle) \leftrightarrow \mathbf{next} sends(S, R, X) \wedge K \in X$ .

The first rule says that, if the sender sends the receiver a message, then the receiver will receive the message at the same time; and the second one says that the sender sends the receiver a message packet with a signed commitment to a key if it will send the receiver a packet containing that key at the next moment in time.

Zero-delay mode is an idealized mode. However, generally the time spent between sending and receiving messages cannot be zero. Considering this point, we give the definition of send-receive modes by introducing a generic form.

**Definition 6 (time intervals)** For a send-receive mode, all packets  $P_i$  must arrive within a certain time interval  $[u, v]$  relative to the current time defined as follows:

$$sends(S, R, P_i) \rightarrow \mathbf{next}^{(m)} receives(R, P_i), u \leq m \leq v.$$

Let the current time be  $t_c$  (time of sending a packet). Definition 6 indicates that any packet sent by the sender must arrive at a moment between  $t_{c+u}$  and  $t_{c+v}$ .

**Definition 7 (time distance of sending)** Let  $d = 1/r$ , where  $r$  is the packet rate (i.e., number of packets sent per unit time). We call  $d$  the time distance of sending between two packets.

Noting that a send-receive mode is in fact determined based on the time interval of packet arrival and the time distance of sending, we have the formal definition of a mode as follows:

**Definition 8 (send-receive modes)** We use the notation  $m([u, v], d)$  to represent a send-receive mode of the PCTS scheme of TESLA or, simply, a mode if  $u, v, d \in \mathcal{N}$ , the set of all natural numbers, and  $u \leq v$ , where  $[u, v]$  is the time interval of this mode, and  $d$  the time distance of sending with it. We say that  $m([u, v], d)$  is a safe mode if  $v < d$ .

The following generic rules specify a given mode  $m([u, v], d)$ :<sup>4</sup>

- G1.  $sends(S, R, X) \leftrightarrow \mathbf{next}^{(u)} receives(R, X) \vee \dots \vee \mathbf{next}^{(v)} receives(R, X)$ .  
 G2.  $sends(S, R, \langle D, MAC(f'(K), D) \rangle) \leftrightarrow \mathbf{next}^{(d)} sends(S, R, X) \wedge K \in X$ .

Mode-specific rules are determined when  $u, v$  and  $d$  are given. For example, within the mode  $m([2, 3], 4)$ , we have

- S1.  $sends(S, R, X) \leftrightarrow \mathbf{next}^{(2)} receives(R, X) \vee \mathbf{next}^{(3)} receives(R, X)$ .  
 S2.  $sends(S, R, \langle D, MAC(f'(K), D) \rangle) \leftrightarrow \mathbf{next}^{(4)} sends(S, R, X) \wedge K \in X$ .

Modules  $M_{mk}$ ,  $M_{ms}$ , and  $M_{ar}$  are fixed for any mode. Due to space limitations, they are listed below without explanations.

**$M_{mk}$  (message receiving and knowledge gained)**

<sup>4</sup> In what follows we will use  $\mathbf{next}^m$  to indicate  $m$  consecutive occurrences of  $\mathbf{next}$ .

- G3.  $receives(A, \langle X, Y \rangle) \rightarrow receives(A, X) \wedge receives(A, Y)$ .
- G4.  $receives(A, X) \rightarrow knows(A, X)$ .
- G5.  $knows(A, K) \rightarrow knows(A, f(K)) \wedge knows(A, f'(K))$ .
- G6.  $knows(A, \{X\}_{SK(B)}) \rightarrow knows(A, X)$ .
- G7.  $knows(A, K) \wedge knows(A, X) \rightarrow knows(A, MAC(K, X))$ .
- G8.  $knows(A, X) \rightarrow \mathbf{next} knows(A, X)$ .

where  $SK(B)$  is the private key of agent  $B$  and its corresponding public key can be known by anybody, so we have G8.

#### $M_{ms}$ (Message sending)

- G9.  $sends(A, B, \langle X, Y \rangle) \rightarrow sends(A, B, X) \wedge sends(A, B, Y)$ .
- G10.  $sends(A, B, X) \rightarrow has\_sent(A, B, X)$ .
- G11.  $has\_sent(A, B, X) \rightarrow \mathbf{next} has\_sent(A, B, X)$ .

#### $M_{ar}$ (Authentication rules)

- G12.  $is\_auth(\langle X, MAC(f'(k), D) \rangle) \leftrightarrow verify\_success(f(K)) \wedge verify\_success(MAC(f'(K), D))$ .
- G13.  $is\_auth(X) \rightarrow has\_been\_auth(X)$ .
- G14.  $\mathbf{B}_R has\_been\_auth(X) \rightarrow \mathbf{next} \mathbf{B}_R has\_been\_auth(X)$ .
- G15.  $receives(R, \langle X, MAC(f'(K), D) \rangle) \wedge$   
 $\mathbf{B}_R \neg has\_sent(S, R, K) \rightarrow \mathbf{B}_R arrive\_safe(\langle X, MAC(f'(K), D) \rangle)$ .
- G16.  $arrive\_safe(X) \rightarrow has\_arrive\_safe(X)$ .
- G17.  $\mathbf{B}_R has\_arrive\_safe(X) \rightarrow \mathbf{next} \mathbf{B}_R has\_arrive\_safe(X)$ .
- G18.  $\mathbf{B}_R verify\_success(f(K)) \leftrightarrow \mathbf{B}_R has\_arrive\_safe(\langle X, MAC(f'(K), D) \rangle) \wedge knows(R, K) \wedge$   
 $\mathbf{B}_R has\_been\_auth(\langle D', MAC(f'(K), D') \rangle) \wedge f(K) \in D'$ .
- G19.  $\mathbf{B}_R verify\_success(MAC(f'(K), D)) \leftrightarrow \mathbf{B}_R has\_arrive\_safe(\langle X, MAC(f'(K), D) \rangle) \wedge$   
 $knows(R, K) \wedge MAC(f'(K), X) = MAC(f'(K), D)$ .

Thus, we have obtained a theory  $\mathbf{T} = M_{sr} \cup M_{mk} \cup M_{ms} \cup M_{ar}$  specifying the PCTS scheme of TESLA given in Section 2, where each module contains the relevant axioms given above.

## 6.2 Correctness Analysis

The correctness condition for a given TESLA scheme should guarantee that if the receiver (receivers) can verify that a packet is authentic, then the packet was indeed sent by the sender.

**Definition 9 (correctness condition)** *The local correctness for a TESLA scheme to the receiver  $R$  who receives messages from the sender  $S$  means that, if  $R$  has verified that a packet is authentic, then the packet was indeed sent by  $S$ . That is,  $\forall X (\mathbf{B}_R has\_been\_auth(X) \wedge has\_sent(A, R, X) \rightarrow A = S)$ . Furthermore, the (global) correctness for the TESLA scheme means that the local correctness for the scheme to all receivers holds.*

The theory discussed above is based on a time model where the clock is regarded as the synchronized receiver's clock (correspondingly to the global clock). It provides a basis for the receiver to verify stream messages received through the PCTS scheme of TESLA if the scheme with its send-receive mode satisfies the correctness condition.

Based on the theory developed above, we can show that the correctness condition of the TESLA protocol holds within the scheme.

**Proposition 1** *The PCTS scheme with the mode  $m([u, v], d)$  mode is secure (i.e., it satisfies the correctness condition) if  $m([u, v], d)$  is a safe mode.*

We can also use the theory to show that the PCTS scheme with an unsafe mode, e.g., the mode  $m([1, 4], 2)$ , provides chances for the intruder to attack the system. Consider the case: assume that packets  $P_i$  and  $P_{i+1}$  are sent out by the sender at time  $t$  (the current moment in time) and at  $t + 2$  (the next next moment), respectively. The intruder,  $I$ , first intercepts  $P_i$  at  $t + 2$  and then, at  $t + 3$ , again intercepts  $P_{i+1}$  when it arrives. By creating a packet  $P'_i$ , instead of  $P_i$ , using key  $K_i$  in packet  $P_{i+1}$ ,  $I$  masquerades as the sender send packet  $P'_i$  to the receiver. The attack will be successful if  $P'_i$  reaches the receiver at  $t + 4$ .

### 6.3 Mechanising Correctness Proofs

In order to automatically analyse the correctness of a scheme of the protocol, we need to mechanize the theory describing the behaviour of the protocol in an appropriate proof system. In our approach, such system-specific trust theories developed for specifying communications protocols do not depend on a specific implementation. The user is therefore allowed to freely choose the tools for mechanizing these theories. Below we will show how modal tableaux can be used to verify the properties of the TESLA protocol. Modular structure offers convenience to the user for translating a theory to an executable code (program) in a mechanised proof system, such as Isabelle [18] or the SMV model checker [17].

With the labelled modal tableaux system KEM, to show a safe mode satisfies the correctness condition, we only need to show that in this mode  $A = S$  is a *KEM-consequence of a set of formulas*  $\Gamma = \{\mathbf{B}_R \text{ has\_been\_auth}(X), \text{has\_sent}(A, R, X)\}$ . Due to space limitations, we only give a simple case to show how the labelled tableaux system works on checking the properties of TESLA. With the send-receive mode  $m([2, 3], 4)$ , we assume that the message has arrived safely and it has been authenticated based on the time the message was received and the contents of the message:

- H1. **first next**<sup>(3)</sup>  $\text{receives}(R, \langle X, Y \rangle)$
- H2. **first next**<sup>(7)</sup>  $\text{receives}(R, X1) \wedge K \in X1$
- H3.  $\text{MAC}(f'(K), X) = Y$
- H4. **first next**<sup>(8)</sup>  $\mathbf{B}_R \text{ is\_auth}(\langle X, Y \rangle)$

Then, we can prove the following property:

$$(A). \quad \mathbf{first\ next}^{(8)} \mathbf{B}_R (\text{is\_auth}(\langle X, Y \rangle) \rightarrow (\mathbf{first\ sends}(S, R, \langle X, Y \rangle) \vee \mathbf{first\ next\ sends}(S, R, \langle X, Y \rangle)))$$

It basically says that if at time  $t_8$ , agent  $R$  believes that if the message is authenticated, then it must have been sent at either time  $t_0$  or time  $t_1$  (agent  $R$  does not really know the exact time when the message was sent, however, it knows about the time interval).

In the following we show the tableaux proof of the property. All the rules of the PCTS scheme of TESLA are at our disposal as well as the assumptions made above; each is labelled with a generic universal label that would unify with any given label. Tableaux rules have been applied exhaustively until all the branches have been completed (details of proof steps are omitted). We also assume a that biconditional (such as S1 used in the proof) is the conjunction of two implications.

1.  $sends(S, R, \langle X, Y \rangle) \leftrightarrow \mathbf{next}^{(2)} receives(R, \langle X, Y \rangle) \vee \mathbf{next}^{(3)} receives(R, \langle X, Y \rangle) : W1$
2.  $\mathbf{first next}^{(3)} receives(R, \langle X, Y \rangle) : W2$
3.  $\mathbf{first next}^{(8)} \mathbf{B}_R is\_auth(\langle X, Y \rangle) : W3$
4.  $\neg \mathbf{first next}^{(8)} \mathbf{B}_R (is\_auth(\langle X, Y \rangle)) \rightarrow (\mathbf{first sends}(S, R, \langle X, Y \rangle) \vee \mathbf{first next sends}(S, R, \langle X, Y \rangle)) : W4$
5.  $\neg \mathbf{next}^{(8)} \mathbf{B}_R (is\_auth(\langle X, Y \rangle)) \rightarrow (\mathbf{first sends}(S, R, \langle X, Y \rangle) \vee \mathbf{first next sends}(S, R, \langle X, Y \rangle)) : (t_0, W4)$
6.  $\neg \mathbf{next}^{(7)} \mathbf{B}_R (is\_auth(\langle X, Y \rangle)) \rightarrow (\mathbf{first sends}(S, R, \langle X, Y \rangle) \vee \mathbf{first next sends}(S, R, \langle X, Y \rangle)) : (t_1, (t_0, W4))$
7. ... (expansion rule for  $\mathbf{next}$  is applied 7 times,  $\mu_T$ )
8.  $\neg \mathbf{B}_R (is\_auth(\langle X, Y \rangle)) \rightarrow (\mathbf{first sends}(S, R, \langle X, Y \rangle) \vee \mathbf{first next sends}(S, R, \langle X, Y \rangle)) : (t_8, (\dots (t_1, (t_0, W4)) \dots))$
9.  $\neg (is\_auth(\langle X, Y \rangle)) \rightarrow (\mathbf{first sends}(S, R, \langle X, Y \rangle) \vee \mathbf{first next sends}(S, R, \langle X, Y \rangle)) : (w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots)))$
10.  $is\_auth(\langle X, Y \rangle) : (w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots)))$
11.  $\neg (\mathbf{first sends}(S, R, \langle X, Y \rangle) \vee \mathbf{first next sends}(S, R, \langle X, Y \rangle)) : (w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots)))$
12.  $\neg \mathbf{first sends}(S, R, \langle X, Y \rangle) : (w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots)))$
13.  $\neg \mathbf{first next sends}(S, R, \langle X, Y \rangle) : (w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots)))$
14.  $\neg sends(S, R, \langle X, Y \rangle) : (t_0, (w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots))))$
15.  $\neg \mathbf{next sends}(S, R, \langle X, Y \rangle) : (t_0, ((w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots))))$
16.  $\neg sends(S, R, \langle X, Y \rangle) : (t_1, (t_0, ((w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots))))$
17.  $sends(S, R, \langle X, Y \rangle) \rightarrow \mathbf{next}^{(2)} receives(R, \langle X, Y \rangle) \vee \mathbf{next}^{(3)} receives(R, \langle X, Y \rangle) : W1$
18.  $\mathbf{next}^{(2)} receives(R, \langle X, Y \rangle) \vee \mathbf{next}^{(3)} receives(R, \langle X, Y \rangle) \rightarrow sends(S, R, \langle X, Y \rangle) : W1$
19.  $\neg (\mathbf{next}^{(2)} receives(R, \langle X, Y \rangle) \vee \mathbf{next}^{(3)} receives(R, \langle X, Y \rangle)) : (t_1, (t_0, ((w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots))))$
20.  $\neg \mathbf{next}^{(2)} receives(R, \langle X, Y \rangle) : (t_1, (t_0, ((w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots))))$
21.  $\neg \mathbf{next}^{(3)} receives(R, \langle X, Y \rangle) : (t_1, (t_0, ((w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots))))$
22.  $\mathbf{next}^{(3)} receives(R, \langle X, Y \rangle) : (t_0, W2)$
23.  $\mathbf{next}^{(2)} receives(R, \langle X, Y \rangle) : (t_1, (t_0, W2))$
24.  $\times [(t_1, (t_0, ((w^i, (t_8, (\dots (t_1, (t_0, W4)) \dots))))$  and  $(t_1, (t_0, W2))$  unify]

This proof has only one branch which is closed. This shows that agent  $R$ 's belief has been justified based on the assumptions.

## 7 Concluding Remarks

With the logic FL, we use a simple case of the fibred semantics arising from Kripke models with a single time model. However, it is not difficult to extend it with other different time models. Such extensions would be needed when one wants to deal with different local clocks for multi-agent systems.

We have discussed an application of the logic FL in analysing the TESLA protocol. Archer [1] uses the theorem prover TAME, and Broadfoot *et al* [3] use model checking techniques, to analyse TESLA. The advantage of these methods is that some properties of the protocol can easily be captured through proof systems, but a drawback is that the formal representations involved in such proofs are often not easily validated by the user. Our approach separates the theory from its implementation and helps a protocol designer to capture the meanings of the theory as a whole. Our analysis has shown that the PCTS scheme of TESLA with a safe send- is secure given that the correctness condition is satisfied. We believe that this approach can be easily extended for the analysis of other schemes of the TESLA protocol, and for other security protocols.

## Acknowledgements

This work is supported in part by Australia Research Council under Discovery Project No. DP0452628 on “Combining modal logic for dynamic and multi-agents systems”.

## References

1. Myla Archer. Proving correctness of the basic TESLA multicast stream authentication protocol with TAME. In *Workshop on Issues in the Theory of Security*, 2002.
2. A. Artosi, P. Benassi, G. Governatori, and A. Rotolo. Shakespearian modal logic: A labelled treatment of modal identity. *Advances in Modal Logic*.1, 1–21. CSLI, 1998.
3. P. Broadfoot and G. Lowe. Analysing a stream authentication protocol using model checking. In *Proc 7th ESORICS*, 2002.
4. M. Burrows, M. Abadi, and R. M. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.
5. E. Clarke, S. Jha, and W. Marrero. A machine checkable logic of knowledge for specifying security properties of electronic commerce protocols. In *Proceedings of the Workshop on Formal Methods and Security Protocols*, 1998.
6. N. Durgin, J. Mitchell, and D. Pavlovic. A compositional logic for proving security properties of protocols. *Journal of Computer Security*, 11(2003):677–721.
7. M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Reidel, 1983.
8. D. M. Gabbay. *Fibring Logics*. OUP, 1999.
9. D.M. Gabbay and G. Governatori. Fibred modal tableaux. *Labelled Deduction*, pages 163–194. Kluwer, 2000.
10. G. Governatori. Labelled tableaux for multi-modal logics. *Proc. Tableaux’95*, LNAI 918, pages 79–94. Springer, 1995.
11. G. Governatori. *Un modello formale per il ragionamento giuridico*. PhD thesis, University of Bologna, 1997.
12. 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.
13. G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic*. Routledge, 1996.
14. S. Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.
15. C. Liu and M. A. Orgun. Dealing with multiple granularity of time in temporal logic programming. *Journal of Symbolic Computation*, 22:699–720, 1996.
16. C. Liu, M. Ozols, and M. A. Orgun. A temporalised belief logic for specifying the dynamics of trust for multi-agent systems. In *Proc. ASIAN 2004*, LNCS, 3321, pages 142–156. Springer, 2004.
17. K. L. McMillan. Symbolic model checking - an approach to the state explosion problem. PhD thesis, SCS, Carnegie Mellon University, 1992.
18. L. C. Paulson. *ML for Working Programmer*. CUP, 1996.
19. A. Perrig, R. Canetti, J. D. Tygar, and D. Song. Efficient authentication and signing of multicast streams over lossy channels. In *IEEE Symposium on Security and Privacy*, pages 56–73, 2000.