Analysing Stream Authentication Protocols in Autonomous Agent-Based Systems

Mehmet A. Orgun, Ji Ma
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
Email: {mehmet,jma}@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

Guido Governatori
School of ITEE
The University of Queensland
Brisbane, QLD 4072, Australia
Email: guido@itee.uq.edu.au

Abstract

In stream authentication protocols used for large-scale data dissemination in autonomuous systems, authentication is based on the timing of the publication of keys, and 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 a protocol. Many existing logics and approaches have successfully been applied to specify other types of authentication protocols, but most of them are not appropriate for analysing stream authentication protocols. We therefore consider a fibred modal logic that combines a belief logic with a linear-time temporal logic which can be used to analyse time-varying aspects of certain problems. With this logical system one is able to build theories of trust for analysing stream authentication protocols, which can deal with not only agent beliefs but also the timing properties of an autonomous agent-based system.

1. Introduction

Multi-agent systems, typically autonomous real word 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 [3], [4], [5], [7].

As an example, we consider the TESLA (Timed Efficient Stream Loss-tolerant Authentication) protocol, a multicast (broadcast) stream authentication protocol for large-scale data dissemination in autonomous systems, developed by Perrig *et al.* [17]. 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. In fact, any stream authentication protocol involves not only a key management scheme but also its critical use of timing. In an autonomous system whose authentication is based on TESLA, the process for verifying data packets received to be authentic also 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.

The work presented in this article has been supported in part by an Australian Research Council (ARC) Discovery Project DP0452628 and a Macquarie University Research Development Grant (MURDG).

Many different logics have successfully been applied for specifying and verifying aspects of some other types of authentication protocols [3], [5], [15]. However, most of these logics are not appropriate for stream authentication protocols because they lack the ability for modelling and reasoning about the evolution of the system in which they are applied (that is, the notion of dynamic change). It has also been understood that any logical system used for modeling active agents should be a combined system of logics of knowledge, belief, time and norms [6] since these are the essential concepts to be reasoned about. In order to analyse stream authentication protocols it is necessary to have a logic that can satisfactorily deal with all aspects of those concepts. Towards achieving such a goal, we consider a fibred logic, which combines a belief logic with a linear-time temporal logic. The fibred logic developed in this paper has more expressive power than a temporalised belief logic such as TML$^+$ [11] does. It allows us to express the behavior and beliefs of an agent as temporal propositions, which are often needed for analysing stream authentication protocols.

In analysing a communication protocol, we accept the fact that honest agents would follow the prescribed steps of the protocol correctly, but also that an intruder may have the ability to interfere with it. With the logical framework proposed in this paper, we encapsulate such assumptions in notions of trust and represent them by axioms (rules). These axioms, together with the logic, form a theory, which we call a *theory of trust* [9], [13]. Such theories can deal with not only agent beliefs but also the timing properties of autonomous agent-based systems, and they provide a basis for analysing stream authentication protocols.

With the aim of providing solutions based on logic for analysing communication protocols for autonomous agent-based systems, we are motivated to investigate methods for formalizing an authentication protocol with theories of trust, and to develop techniques for reasoning about security properties that such a protocol may satisfy.

In the following, we first introduce TESLA and discuss the fibred logic. We then consider how to establish theories of trust to specify the behaviour of the TESLA protocol in the fibred logic. A theory of trust is modular, with each module providing a collection of rules that specifically describe some specific aspect of an authentication protocol. We consider the mechanization of a theory for TESLA with a general send-receive mode, and discuss the correctness of this protocol. Finally, we conclude the paper with a brief discussion.

2. The TESLA Protocol

Adopting the notation in the TESLA protocol [17], when we talk about the protocol, S, R and I are always used to denote the sender, the receiver and the intruder, respectively. We also use the tuple $\langle X, Y\rangle$ to denote the concatenation of X and Y. A stream α is divided into chunks M_i (called messages), so we may have $\alpha = \langle M_1, M_2, \ldots, M_l\rangle$. Each message M_i is sent in a packet P_i, along with additional authentication information. A message authentication code (MAC) is derived by applying an authentication scheme, together with a secret key, to a message.

Perrig *et al.* [17] propose five schemes for the TESLA Protocol. We consider scheme I, and simply call it the PCTS (Perrig-Canetti-Tygar-Song) scheme. Within PCTS, the sender issues a signed commitment to a key that is only known to itself. To send message M_i, the sender uses that key to compute a MAC on a packet P_i, and later discloses the key in packet P_{i+1}, which enables the receiver to verify the commitment and the MAC of packet P_i. A successful verification will imply that packet P_i is authenticated and trusted.

In the notation proposed by Broadfoot and Lowe [2], we formulate the sequence of message packets as follows:

$$\begin{array}{lll}
P_{0r}. & R \to S: & \langle n_R\rangle \\
P_{0s}. & S \to R: & \langle \{f(K_1), n_R\}_{SK(S)}\rangle \\
P_1. & S \to R: & \langle\langle M_1, f(K_2)\rangle, MAC(K'_1, \langle M_1, f(K_2)\rangle)\rangle \\
 & & \vdots \\
P_i. & S \to R: & \langle D_i, MAC(K'_i, D_i)\rangle \text{ (for all } i \geq 2) \\
 & & \vdots
\end{array}$$

where n_R is a nonce (that is, number used once) generated by the receiver R, and $D_i = \langle M_i, f(K_{i+1}), K_{i-1}\rangle$ for all $i \geq 2$, $K'_j = f'(K_j)$ for all $j \geq 1$, and f and f' are two different pseudo-random functions.

Apart from the initial messages P_{0r}, P_{0s} and P_1, any packet P_i has the standard form $\langle D_i, MAC(K'_i, D_i)\rangle$ for all $i \geq 2$ [2]. To send message M_i, the sender first picks a fresh random key K_{i+1} to construct packet $P_i = \langle D_i, MAC(K'_i, D_i)\rangle$, then sends the packet to the receiver. When the packet P_i is received, the receiver cannot verify the MAC immediately, since it cannot reconstruct K'_i without knowing K_i, which is contained in packet P_{i+1}. Therefore, only once P_{i+1} is received, the receiver is able to verify the MAC. Packet $P_{i+1} = \langle D_{i+1}, MAC(K'_{i+1}, D_{i+1})\rangle$ discloses K_i and allows the receiver first to verify that K_i is correct ($f(K_i)$ equals the commitment which was sent in P_{i-1}); and second to compute $K'_i = f'(K_i)$ and check the authenticity of packet P_i by verifying the MAC of P_i.

In analysing the TESLA protocol, we make the following assumptions:

- The sender is honest and works correctly, following all requirements, including timing requirements, of the protocol.
- The receiver accepts packet P_1 as authentic only when it believes the key commitment and the MAC of the packet have been successfully verified. A stream $\alpha = \langle M_1, M_2, \ldots, M_l\rangle$ is considered valid iff the receiver accepts all messages $M_1, M_2, \ldots, M_l$ as authentic, i.e., all packets $P_1, P_2, \ldots, P_l$ are successfully authenticated.
- The intruder is assumed to have 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 pseudorandom function applications and MACs with negligible delay. The intruder may be able to launch two kinds of attacks: *weak attacks* and *strong attacks*. A *weak attack* allows the intruder to inject new packets, but not delete the packets sent by the sender. A *strong attack* allows the intruder the full control of the channel, and so he can add, replace, or delete any packets. Nonetheless, the intruder cannot invert a psedorandom function with non-negligible probability. Also the intruder can only create a fake packet P'_i against P_i when it has received the packet that discloses key K_i.

The security property for the 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. Perrig *et al.* therefore give a security condition, which we re-state as follows:

Statement 1 (security condition [17]): A data packet P_i arrived safely, if the receiver can unambiguously decide, based on the synchronized time, that the sender did not yet send out the corresponding key disclosure packet P_j, i.e., the timing condition $ArrT_i < SenT_j$ holds, where $ArrT_i$ stands for the arrival time of packet P_i, and $SenT_j$ stands for the time when packet P_j is sent out.

Statement 2 indicates that the security of a TESLA scheme does not rely on any assumption on network latency, but only on the security condition.

Statement 2 (secure schemes): A TESLA scheme is secure if the security condition holds for all runs of the protocol scheme, given the assumptions above.

To analyse the TESLA protocol and show a particular scheme is secure, we first need a formal framework for formalizing the protocol. As we have mentioned earlier, we consider fibring of a belief logic with a temporal logic.

3. The Fibred Logic

We combine, using the fibring technique, the Typed Modal Logic (TML), a variant of the modal logic KD of beliefs [11], with the Simple Linear-time Temporal logic (SLTL) which is suitable for specifying events that may run on different clocks (time-lines) of varying rates of progress [10]. 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. With this logical system one is also able to build theories of trust for the description of, and reasoning about, authentication protocols for/in multi-agent systems. The details of the fibring construction for TML and SLTL is given elsewhere [12]. This section gives a more informal and intuitive introduction to the fibred logic and its constituents for completeness.

The fibred logic (FL for short) has two classes of modal operators: (1) belief operators; (2) temporal operators. The belief operator, $\mathbf{B}_a$, is intended to denote "agent a believes that". The belief operators are those of TML whereas the temporal operators are those of SLTL. SLTL is a linear-time logic where the collection of time points is the set of natural numbers with its usual ordering relation <. It has two temporal operators, **first** and **next**, which refer to the initial moment and the next moment in time respectively [10]. The meaning of SLTL 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., $(0, 1, 2, \ldots)$ and a local clock (or simply, a clock) is defined as an infinite subsequence of the global clock.

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

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

Table 1. Interpretation of temporal operators (A is a formula; T represents value **true** and F value **false**)

In the table, suppose that A is the formula $has(bob, key)$ and $ck = (t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7 \ldots)$ is a given local clock where each t_i is a moment on the clock. Suppose that the meaning of formula A over the clock ck is as given in the first line of Table I (where A is true at times t_0 and t_1 and false at times t_2 and t_3 and so on). Since the initial time of ck is t_0, the meaning of a formula of the form $\mathbf{first}\ A$ at any given moment in time is defined by the meaning of A at time t_0 (e.g., true in the above example). The meaning of a formula of the form $\mathbf{next}\ A$ at any given moment in time t_i is defined by the meaning of A at time t_{i+1}; for example, at time t_3 A is false, but $\mathbf{next}\ A$ is true because A is true at time t_4.

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

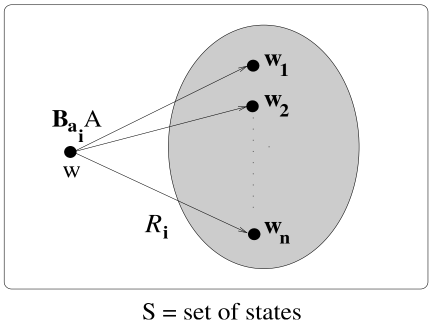
Definition 1 (time models): A time model for the logic SLTL has the form $\mathbf{c} = \langle C, <, \pi^{(\mathbf{c})} \rangle$*, where* $C = (t_0, t_1, t_2, \ldots)$ *is a clock,* $<$ *is the usual ordering relation over C, and* $\pi^{(\mathbf{c})}$ *is an assignment function that gives a value* $\pi^{(\mathbf{c})}(t,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

Then the semantics of the temporal operators of SLTL are given as follows:

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

Let us assume that there are n agents $a_1, \ldots, a_n$ and there are n corresponding modal operators $\mathbf{B}_{a_1}, \ldots, \mathbf{B}_{a_n}$ in the logic, where $\mathbf{B}_{a_i}$ $(1 \le i \le n)$ stands for "agent a_i believes that". A *classical Kripke model* [8] 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 or possible worlds; 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 non-empty set 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 (or accessible); and π is the *assignment function*, which gives a value $\pi(s, q) \in \{true, false\}$ for any $s \in S$ and atomic formula q. Formula φ is satisfiable in the model $\mathbf{m}$ if there exists $s \in S$ such that $\mathbf{m}, s \models \varphi$.

Figure 1 gives an intuitive explanation of the interpretation of belief operators of TML. Recall that R_i is the possibility relation for agent a_i. In other words, given state w, agent a_i considers all the states $w_1, w_2, \ldots, w_n$ possible. Then a formula of the form $\mathbf{B}_{a_i} A$ is true at a given state w if and only if A is true at all states accessible from w with respect to the relation R_i (state w may or may not be accessible from w itself).



S = set of states

Fig. 1. Interpretation of belief operators (R_i is the possibility relation for agent a_i; $w_1, \ldots, w_n$ are the states in S that are accessible from w)

In earlier work, we proposed a temporalised belief logic called TML$^+$ [11], It also has these two classes of modal operators. However, there are certain restrictions on the use of temporal and belief operators, because of the hierarchical combination of belief and temporal logics used. There the temporal logics SLTL is plussed onto the belief logic TML in such a way that temporal operators can never be within the scope of a belief operator in TML$^+$. Hence in TML$^+$, we cannot express a statement asserting that some agent believes an event to happen at some time, e.g., we can have the formula $\mathbf{first}\ \mathbf{B}_{john}\ holds(bob, k)$, but can not have the formula $\mathbf{B}_{john}\ \mathbf{first}\ holds(bob, k)$. The latter formula could be used to express the assertion that *john* believes that at the initial time *bob* holds the key k. Such assertions are often needed in analysing stream authentication protocols, so we therefore consider a more powerful combination technique called *fibring* [6], which treats temporal operators and belief operators equally.

In FL, both the assertions $\mathbf{first}\ \mathbf{B}_{john}\ holds(bob, k)$ and $\mathbf{B}_{john}\ \mathbf{first}\ holds(bob, k)$ are legal formulas. The first formula means that at the initial moment time John believes that Bob has the key k; the second formula means that John believes that at the initial moment time Bob has the key k. FL has stronger expressive power than TML+ , since these operators can occur in any order in formulas, e.g., we may have $\mathbf{next}\ \mathbf{B}_{alice}\ \mathbf{first}\ \mathbf{next}\ \mathbf{B}_{john}\ holds(bob, k)$, which means that “at the next moment Alice believes that at the next moment after initial time John believes that Bob has the key k.”

The formulas of FL may contain any number of applications of the temporal operators and/or the belief operators without any restrictions. 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. This will require us to move between time references and state references freely. The fibring method [6] is used to interweave the semantics of the constituent logics using fibring functions (that move context between time references and state references) in such a way that any formula of FL is interpreted in its proper context.

The discussion of the fibred semantics in the case of the Kripke 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 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 the two logics TML and SLTL have been fibred [12].

In the following, we discuss the axioms and rules of inference of TML that may be used in reasoning about trust theories for authentication protocols in agent-based systems.

The axiom set of FL consists of the following axiom schemata. Let $\mathcal{O} = \{\mathbf{B}_1, \ldots, \mathbf{B}_n, \mathbf{first}, \mathbf{next}\}$ be the set of modal (temporal or belief) operators of BL.

A0. all axioms of classical first-order logic.
A1. $\triangledown(\varphi \rightarrow \psi) \wedge \triangledown\varphi \rightarrow \triangledown\psi$, for all $\triangledown \in \mathcal{O}$.
A2. $\triangledown(\varphi \wedge \psi) \leftrightarrow\ (\triangledown\varphi) \wedge (\triangledown\ \psi)$, for all $\triangledown \in \mathcal{O}$.
A3. $\forall X(\triangledown\varphi(X)) \rightarrow \triangledown(\forall X\varphi(X))$, for all $\triangledown \in \mathcal{O}$.
A4. $\mathbf{B}_i(\neg\varphi) \rightarrow \neg(\mathbf{B}_i\ \varphi)$ for all i $(1 \leq i \leq n)$.
A5. $\mathbf{first}(\neg\varphi) \leftrightarrow \neg(\mathbf{first}\ \varphi)$.
A6. $\mathbf{next}(\neg\varphi) \leftrightarrow \neg(\mathbf{next}\ \varphi)$.
A7. $\mathbf{first}(\mathbf{first}\ \varphi) \leftrightarrow \mathbf{first}\ \varphi$.
A8. $\mathbf{next}(\mathbf{first}\ \varphi) \leftrightarrow \mathbf{first}\ \varphi$.

The rules of inference in the logic FL include:

R1. From φ and $\varphi \rightarrow \psi$ infer ψ. (Modus Ponens)
R2. From $\forall X\varphi(X)$ infer $\varphi(Y)$. (Instantiation)
R3. From $\varphi(X)$ infer $\forall X\varphi(X)$. (Generalisation)
R4. From φ infer $\triangledown\varphi$ for all $\triangledown \in \mathcal{O}$. (Necessitation)

The soundness for the logic FL depends on the soundness theorems for belief logic and SLTL, and is not difficult to prove. The completeness theory for FL can be proved by the techniques used in [6]; we omit the details.

4. Formalizing Authentication Protocols

For analysing a scheme of the TESLA protocol, we first need to establish a theory that describes its behavior or functions of the protocol within the scheme. To gain such a theory, we first analyse the behavior of the protocol (within, for example, the PCTS scheme), identify various aspects of the behavior of the protocol and set them into an appropriate module, and finally transfer informal descriptions of all these aspects into formulae.

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 of time, and it matches the special timing property that the TESLA scheme satisfies: *the sender sends packets at regular time intervals*. The assumption makes our discussion simple without harming its correctness.

We set a theory for specifying (the PCTS scheme of) TESLA 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 axioms (axiom schemata). We assume the basic types include

A, B, S, R, I:	$Agents$
X, Y, D:	$Messages$
K, K_1, K_2:	$Keys$

In the following, we consider three modes, denoted PCTS-0, PCTS-1, and PCTS-2 respectively, as examples, and present the theories that specify these particular schemes of TESLA.

4.1. A Theory for the Scheme PCTS-0

Send-receive mode specification depends on what kind of mode is adopted. We first consider the scheme PCTS-0, for which the send-receive mode is called *the zero-delay mode*. It is based on two assumptions: (1) Zero time (based on the global clock) is spent between sending a message and receiving this message, i.e., the sending time of a packet P_i on the sender's clock 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 scheme PCTS-0, module M_{sr} consists of the following axiom schemata:

Z1. $\mathit{send}(A,B,X) \to \mathit{receive}(B,X)$.

Z2. $\mathbf{first}\ \mathit{send}(S,R,\langle\{f(K_1),n_R\}_{SK(S)}\rangle)$.

Z3. $\mathbf{first\ next}\ \mathit{send}(S,R,\langle\langle M_1,f(K_2)\rangle, \mathit{MAC}(f'(K_1), \langle M_1,f(K_2)\rangle)\rangle$.

Z4. $\mathit{send}(S,R,\langle D,\mathit{MAC}(f'(K),D)\rangle) \leftrightarrow \mathbf{next}\ \mathit{send}(S,R,X) \wedge K \in X$.

Considering this module specific to PCTS-0, we write $\mathrm{M}_{sr}^{(0)} = \{Z1, Z2, Z3, Z4\}$. Other modules are given below.

Module M_{mk} (message receiving and knowledge gained)

G5. $\mathit{receive}(A,\langle X,Y\rangle) \to (\mathit{receive}(A,X) \wedge \mathit{receive}(A,Y))$.

G6. $\mathit{receive}(A,X) \to \mathit{know}(A,X)$.

G7. $\mathit{know}(A,K) \to \mathit{know}(A,f(K)) \wedge \mathit{know}(A,f'(K))$.

G8. $\mathit{know}(A,\{X\}_{SK(B)}) \to \mathit{know}(A,X)$.

G9. $(\mathit{know}(A,K) \wedge \mathit{know}(A,X)) \to \mathit{know}(A,\mathit{MAC}(K,X))$.

G10. $\mathit{know}(A,X) \to \mathbf{next}\ \mathit{know}(A,X)$.

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

Module M_{ms} (Message sending)

G11. $\mathit{send}(A,B,\langle X,Y\rangle) \to (\mathit{send}(A,B,X) \wedge \mathit{send}(A,B,Y))$.

G12. $\mathit{send}(A,B,X) \to \mathit{has\_sent}(A,B,X)$.

G13. $\mathit{has\_sent}(A,B,X) \to \mathbf{next}\ \mathit{has\_sent}(A,B,X)$.

Module M_{ar} (Authentication rules)

G14. $\mathit{is\_auth}(\langle X,\mathit{MAC}(f'(k),D)\rangle) \leftrightarrow \mathit{verify\_success}(f(K)) \wedge \mathit{verify\_success}(\mathit{MAC}(f'(K),D))$.

G15. $\mathit{is\_auth}(X) \to \mathit{has\_been\_auth}(X)$.

G16. $\mathbf{B}_R\ \mathit{has\_been\_auth}(X) \to \mathbf{next}\ \mathbf{B}_R\ \mathit{has\_been\_auth}(X)$.

G17. $\mathit{receive}(R,\langle X,\mathit{MAC}(f'(K),D)\rangle) \wedge \mathbf{B}_R\ \neg \mathit{has\_sent}(S,R,K) \to \mathbf{B}_R\ \mathit{arrive\_safe}(\langle X,\mathit{MAC}(f'(K),D)\rangle)$.

G18. $\mathit{arrive\_safe}(X) \to \mathit{has\_arrive\_safe}(X)$.

G19. $\mathbf{B}_R\ \mathit{has\_arrive\_safe}(X) \to \mathbf{next}\ \mathbf{B}_R\ \mathit{has\_arrive\_safe}(X)$.

G20. $\mathbf{B}_R\ \mathit{verify\_success}(f(K)) \leftrightarrow \mathbf{B}_R\ \mathit{has\_arrive\_safe}(\langle X,\mathit{MAC}(f'(K),D)\rangle) \wedge \mathit{know}(R,K) \wedge \mathbf{B}_R\ \mathit{has\_been\_auth}(\langle D',\mathit{MAC}(f'(K),D')\rangle) \wedge f(K) \in D'$.

G21. $\mathbf{B}_R\ \mathit{verify\_success}(\mathit{MAC}(f'(K),D)) \leftrightarrow \mathbf{B}_R\ \mathit{has\_arrive\_safe}(\langle X,\mathit{MAC}(f'(K),D)\rangle) \wedge \mathit{know}(R,K) \wedge \mathit{MAC}(f'(K),X) = \mathit{MAC}(f'(K),D)$.

We now have the following:

$\mathrm{M}_{mk} = \{\mathrm{G5}, \mathrm{G6}, \mathrm{G7}, \mathrm{G8}, \mathrm{G9}, \mathrm{G10}\}$,

$\mathrm{M}_{ms} = \{\mathrm{G11}, \mathrm{G12}, \mathrm{G13}\}$, and

$\mathrm{M}_{ar} = \{\mathrm{G14}, \mathrm{G15}, \mathrm{G16}, \mathrm{G17}, \mathrm{G18}, \mathrm{G19}, \mathrm{G20}, \mathrm{G21}\}$.

Thus, for the scheme PCTS-0, we have the theory

$$\mathbf{T}_0 = \mathrm{M}_{sr}^{(0)} \cup \mathrm{M}_{mk} \cup \mathrm{M}_{ms} \cup \mathrm{M}_{ar}.$$

In this scheme, what the intruder, I, is able to do is that, after receiving the packet P_i, I waits for the next packet P_{i+1} and, once received P_{i+1}, creates a fake packet P'_i using K_i contained in P_{i+1} and then masquerades as the sender to send P'_i to the receiver. This is a weak attack, with both packets P_i and P'_i reaching the receiver. We express this intruder process by the following formulas:

I1. $\mathit{send}(S,R,P_i) \wedge \mathbf{next}\ \mathit{send}(S,R,P_{i+1}) \to (\mathit{receive}(I,P_i) \wedge \mathbf{next}\ \mathit{receive}(I,P_{i+1}) \to \mathbf{next}\ \mathit{create}(I,P'_i))$.

I2. $\mathit{create}(I,P'_i) \to (\mathit{send}(I(S),R,P'_i) \to \mathit{receive}(R,P'_i))$

Such an attack can be detected by the receiver easily. In fact, the receiver knows that key K_i, which the intruder uses to create P'_i, has been sent out and the receiver may have received it (within packet P_{i+1}) at the time when P'_i arrives. Therefore, it is impossible that the receiver accepts P'_i as authentic.

4.2. Scheme PCTS-1

PCTS-0 is an idealized mode with zero time spent between sending and receiving messages. We now discuss the scheme PCTS-1, for which a different send-receive mode with a smaller time granularity is adopted. In this mode, we can conveniently deal with time intervals when the network delay from the sender to the receiver must be considered. We assume that this mode satisfies the following two assumptions: (1) The arrival time of a packet sent at the current moment in time can be the '*next*' moment or the '*next next*' moment or the '*next next next*' moment in time; and (2) The packet rate is 1/4 (i.e., 1 packet per 4 units time). With this mode, $\mathrm{M}_{sr}^{(0)}$ should be replaced by $\mathrm{M}_{sr}^{(1)}$, which consists of axioms S1 – S4 as follows:

S1. $send(A,B,X) \rightarrow$
$(\mathbf{next}\ receive(B,X) \vee \mathbf{next}^{(2)} receive(B,X)$
$\vee \mathbf{next}^{(3)} receive(B,X))$.
S2. $\mathbf{first}\ send(S,R,\langle\{f(K_1),n_R\}_{SK(S)}\rangle)$.
S3. $\mathbf{first}\ \mathbf{next}^{(4)}\ send(S,R,\langle\langle M_1,f(K_2)\rangle,$
$MAC(f'(K_1),\langle M_1,f(K_2)\rangle)\rangle)$.
S4. $send(S,R,\langle D,MAC(f'(k),D)\rangle) \leftrightarrow$
$\mathbf{next}^{(4)}\ send(S,R,X) \wedge K \in X$.

where $\mathbf{next}^{(i)}$ denotes i applications of $\mathbf{next}$. Here S1 corresponds to Z1, and it specifies the initial action of the scheme. The other three axioms capture the change of the receiving interval and the packet rate, are different from those in the mode PCTS-0. Thus, for the scheme PCTS-1, we have the theory $\mathbf{T}_1 = \mathrm{M}_{sr}^{(1)} \cup \mathrm{M}_{mk} \cup \mathrm{M}_{ms} \cup \mathrm{M}_{ar}$, where $\mathrm{M}_{sr}^{(1)} = \{\mathrm{S1},\mathrm{S2},\mathrm{S3},\mathrm{S4}\}$.

The intruder may do weak attacks in the PCTS-1 scheme as in PCTS-0. This mode also allows the receiver to detect such attacks easily. It is not difficult to show that PCTS-1 satisfies the security condition, but the receiver may not definitely believe that a packet is authenticated at times (see next section).

4.3. Scheme PCTS-2

Now let us consider a mode regarded as an example of a failure mode. There are two assumptions for the scheme PCTS-2: (1) The arrival time of a packet sent at the current moment in time can be in a time interval between the next moment and the next fourth moment in time; and (2) the packet rate is 1/2 (i.e., 1 packet per 2 units time). With this mode, we have the following axioms that correspond to Z1 – Z4, respectively:

F1. $send(A,B,X) \rightarrow$
$(\mathbf{next}\ receive(B,X)$
$\vee \mathbf{next}^{(2)} receive(B,X)$
$\vee \mathbf{next}^{(3)} receive(B,X) \vee \mathbf{next}^{(4)} receive(B,X)$.
F2. $\mathbf{first}\ send(S,R,\langle\{f(K_1),n_R\}_{SK(S)}\rangle)$.
F3. $\mathbf{first}\ \mathbf{next}^{(2)}\ send(S,R,\langle\langle M_1,f(K_2)\rangle,$
$MAC(f'(K_1),\langle M_1,f(K_2)\rangle)\rangle)$.
F4. $send(S,R,\langle D,MAC(f'(K),D)\rangle) \leftrightarrow$
$\mathbf{next}^{(2)}\ send(S,R,X) \wedge K \in X$.

Thus, for the scheme PCTS-2, we have the theory $\mathbf{T}_2 = \mathrm{M}_{sr}^{(2)} \cup \mathrm{M}_{mk} \cup \mathrm{M}_{ms} \cup \mathrm{M}_{ar}$, where $\mathrm{M}_{sr}^{(2)} = \{\mathrm{F1},\mathrm{F2},\mathrm{F3},\mathrm{F4}\}$.

In the scheme PCTS-2, the intruder is able to make strong attacks.

Consider the case as follows: 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 attach will be successful if P_i' reaches the receiver at $t+4$.

The intruder process is formulated as follows:

I1′. $send(S,R,P_i) \wedge \mathbf{next}^{(2)}\ send(S,R,P_{i+1}) \rightarrow$
$(\mathbf{next}^{(2)}\ receive(I,P_i)$
$\wedge \mathbf{next}^{(3)}\ receive(I,P_{i+1}) \rightarrow$
$\mathbf{next}^{(3)}\ create(I,P_i'))$.
I2′. $create(I,P_i') \rightarrow (send(I(S),R,P_i') \rightarrow$
$\mathbf{next}\ receive(R,P_i'))$

According to I1′ and I2′, P_i' may reach the receiver at $t+4$ (the next fourth moment in time). The arrival time of packet P_i' still belongs to the time interval at which packet P_i may arrive. Therefore, the receiver may fail to detect the attack.

5. Correctness Analysis

The correctness for the PCTS scheme (or any other scheme) of TESLA should guarantee that if the receiver can verify that a packet is authentic, then the packet was indeed sent by the sender. For automatically analysing the correctness of a scheme of the protocol, we need to mechanize the theory that describes the behaviour of the protocol in an appropriate proof system. The theories developed for specifying a particular protocol scheme do not depend on a specific implementation. Therefore, in our approach the user is allowed to freely choose the tools for mechanizing these theories. Modular structure of theories further allows the user to translate a theory to an executable code (program) in a certain proof system, such as Isabelle [16], the SMV model checker [14], etc.

In this section, we further discuss a common theory for the PCTS scheme with the general send-receive mode, and, based on this theory, analyse the TESLA protocol.

Based on our logical framework, the security condition in Statement 1 can further be formalized as the following correctness condition:

Statement 3 (correctness condition):
Correctness for a TESLA scheme means that, if receiver R has verified that a packet is authentic, then the packet was indeed sent by sender S. That is,

$$\forall X(\mathbf{B}_R\ has\_been\_auth(X) \wedge has\_sent(A,R,X) \rightarrow A = S).$$

To prove that a scheme of the TESLA protocol is secure, we need to show that the correctness condition holds within the scheme. Furthermore, we extend the definition of send-receive modes by introducing a more generic form.

Definition 2 (time intervals):
For a send-receive mode, there is a time interval *with packet arrival, denoted as $[min, max]$, such that, for all packet P_i,*

$$send(S,R,P_i) \rightarrow \mathbf{next}^{(t)}\ receive(R,P_i), min \le t \le max.$$

We call min the minimum moment *and max the* maximum moment *in time related to the packet arrival in this mode.*

Definition 6 indicates that any packet sent by the sender must arrive at a moment between the min and max moments, defining a time interval during which packets should arrive.

Definition 3 (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 all send-receive modes discussed in the previous section are 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 4 (send-receive modes):
$m([u, v], d)$ is a send-receive mode of the PCTS scheme *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 regarded as the time interval of this mode, and d is the time distance between sending two successive messages. Furthermore, we say that $m([u, v], d)$ is a* safe mode *if $v < d$.*

Thus, the three modes in the previous section can respectively be represented as PCTS-0 $= m([0, 0], 1)$, PCTS-1 $= m([1, 3], 4)$ and PCTS-2 $= m([1, 4], 2)$. We claim that PCTS-0 and PCTS-1 are safe modes, while we demonstrated that PCTS-2 is not.

With the PCTS scheme, axioms (G5-G10) on message receiving and knowledge gained, axioms (G11-G13) on message sending, and axioms (G14-G21) regarding authentication (i.e., modules M_{mk}, M_{ms}, and M_{ar}) are fixed and suitable for all modes. However, M_{sr} specifying the send-receive mode depends on the mode itself. For any send-receive mode $m([u, v], d)$, we have the following generic rules used for specifying the mode:

G1. $send(A, B, X) \rightarrow$
$\quad \mathbf{next}^{(u)} receive(B, X) \vee \ldots \vee \mathbf{next}^{(v)} receive(B, X)$.
G2. $\mathbf{first}\ send(S, R, \langle\{f(K_1), n_R\}_{SK(S)}\rangle)$.
G3. $\mathbf{first}\ \mathbf{next}^{(d)}\ send(S, R, \langle\langle M_1, f(K_2)\rangle,$
$\quad MAC(f'(K_1), \langle M_1, f(K_2)\rangle)\rangle)$.
G4. $send(S, R, \langle D, MAC(f'(K), D)\rangle) \leftrightarrow$
$\quad \mathbf{next}^{(d)}\ send(S, R, X) \wedge K \in X$.

Let $\mathrm{M}_{sr} = \{\mathrm{G1}, \mathrm{G2}, \mathrm{G3}, \mathrm{G4}\}$. Thus, we have the theory that specifies the PCTS scheme with the mode $\mathbf{m}([u, v], d)$:

$$\mathbf{T} = \mathrm{M}_{sr} \cup \mathrm{M}_{mk} \cup \mathrm{M}_{ms} \cup \mathrm{M}_{ar} = \{\mathrm{G1}, \ldots, \mathrm{G21}\}.$$

The theory could be mechanized in a proof systsem. For example, with the SMV, the theory itself can be "MODULE main", submodules should include "MODULE sr", "MODULE mk", "MODULE ms" and "MODULE ar", which are mapped from M_{sr}, M_{mk}, M_{ms} and M_{ar}, respectively.

The theory provides a foundation for analysing the TESLA protocol. In fact, by the theory, we can show that the following lemma holds.

Lemma 1: Given a mode $m([u, v], d)$. Then the PCTS scheme with this mode is secure if $m([u, v], d)$ is a safe mode.

We outline the proof as follows: Within the PCTS scheme, packet P_1 is authenticated with the regular digital signature scheme and can therefore be conducted using a standard verification method. Therefore, proving the correctness for the PCTS scheme of TESLA may be recursively done based on the assumption that the receiver has the authenticated packet P_1 and it was indeed sent by the sender. That is, we have that

$$\mathbf{B}_R\ has\_been\_auth(P_1) \wedge has\_sent(A, R, P_1) \rightarrow A = S.$$

Then, assuming that for all $i (1 \leq i \leq n - 1)$, the formula $\mathbf{B}_R\ has\_been\_auth(P_i) \wedge has\_sent(A, R, P_i) \rightarrow A = S$ holds, we need to show that

$$\mathbf{B}_R\ has\_been\_auth(P_n) \wedge has\_sent(A, R, P_n) \rightarrow A = S.$$

The above assertion holds true based on axioms G10 and G13 if $v < d$, i.e., $m([u, v], d)$ is a safe model.

The theory also gives a basis for the receiver to verify stream messages received through the PCTS scheme of the TESLA protocol if the scheme with its send-receive mode satisfies the correctness condition. From Figure 2, we see that the scheme with the mode PCTS-0 or PCTS-1 is secure, while with the mode PCTS-2 it is not a secure scheme, as a successful attack can be carried out.

Trusting the protocol with a safe send-receive mode, the receiver can unambiguously be sure that, when a packet P_i arrived, the sender must not yet send out the corresponding key disclosure packet P_j (i.e., P_{i+1} in the PCTS scheme). Therefore, the receiver could verify any packet received based on the theory $\mathbf{T}$ that describes the behaviour of the protocol.

However, in practice, safe modes, except for the mode $m([0, 0], 1)$, do not guarantee that the receiver is sure that a packet P_i is authenticated when it arrives, as the sender has not yet sent out the corresponding key to verify P_i. For example, let us consider a case with the mode $m([1, 3], 4)$: the sender S sends packet P_i to the receiver R at time t, R receives it at $t + 3$; then S sends packet P_{i+1} to R at $t + 4$ and R receives it at $t + 5$. It is possible that R may think that P_{i+1} was sent three time units before receiving it, i.e., may think that P_{i+1} was sent at $t + 2$. Therefore the receiver may not accept the fact that, when a packet P_i arrived, S had not yet send out P_{i+1} that contains the key to verify P_i. To exclude such a problem, we recommend a mode $m([u, v], d)$ that satisfies the condition $2v - u < d$.

6. Concluding Remarks

We have presented a logical framework for analysing stream authentication protocols. With the logic, 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 by considering other time models. Such extensions would be needed when one wants to deal with different local clocks (different subsets of the global clock) for multiple receivers involved in a protocol.

In analysing the TESLA protocol, Archer [1] uses the theorem prover TAME, and Broadfood *et al* [2] use model checking techniques. One advantage of those methods is

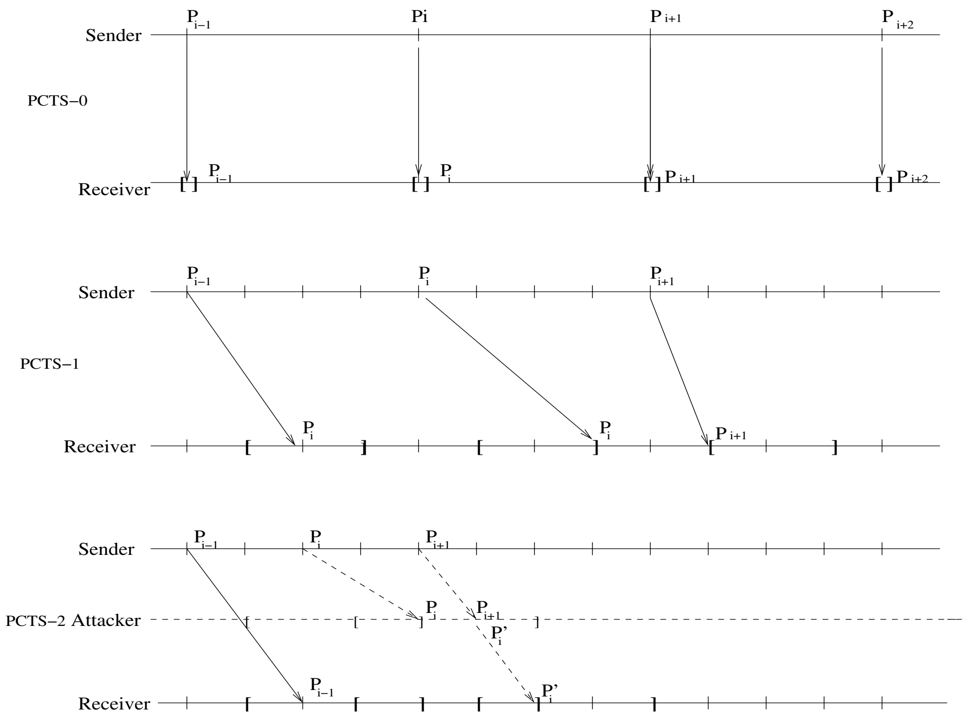


Fig. 2. Modes PCTS-0= $m([0, 0], 1)$, PCTS-1= $m([1, 3], 4)$ and PCTS-2= $m([1, 4], 2)$

that some properties of the protocol can easily be captured through proving systems, but a drawback is that the formal representations involved in such proofs are often not easily understood or validated by the user.

Our approach based on the fibred logic is flexible since the structure of the theory is well-defined, and separating the theory from its implementation helps a protocol designer to capture the meanings of the theory as a whole. Moreover the modular structure makes it easy for the user to modify a theory. Our analysis has shown that the PCTS scheme of TESLA with a safe send-receive mode is secure given that the correctness condition is satisfied.

We believe that our approach can be easily extended such that it is also suitable for other schemes of the TESLA protocol, and for other stream authentication protocols. We have been developing a tableaux-based theorem prover for the fibred logic and we will consider its applications to the verification of stream authentication protocols.

7. References

[1] M. Archer. Proving correctness of the basic TESLA multicast stream authentication protocol with Tame. In *Workshop on Issues in the Theory of Security*, 2002.

[2] P. Broadfoot and G. Lowe. Analysing a stream authentication protocol using model checking, 2002. In *7th European Symposium on Research in Computer Security (ESORICS 2002)*, Pages 146-161.

[3] M. Burrows, M. Abadi and R. Needham. A logic of authentication. ACM Trans. Comput. Syst. Vol.8, No.1 (1990), pp.18–36.

[4] 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.

[5] N. Durgin, J. Mitchell, and D. Pavlovic. A compositional logic for proving security properties of protocols, *Journal of Computer Security*, 11(2003):677–721.

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

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

[8] S. A. 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 Artifi cial Intelligence, LNAI, Vol.2256*, pages 333–344. Springer-Verlag, 2001.

[10] C. Liu and M. A. Orgun. Dealing with multiple granularity of time in temporal logic programming. *Journal of Symbolic Computation*, 22(5/6):699–720, 1996.

[11] C. Liu, M. Ozols and M. A. Orgun. A temporalised belief logic for specifying the dyamics of trust for multi-agent systems. In *Proceedings of the Ninth Asian Computer Science Conference, LNCS, Vol.3321*. Springer-Verlag, 2004.

[12] C. Liu, M. Ozols and M. A. Orgun. A fi bred belief logic for multi-agent systems. In S. Zhang and R. Jarvis (eds), *Proceedings of The 18th Australian Joint Conference on Artifi cial Intelligence*, 5-9 December 2005, LNAI, Vol.3809, pp.29–38. Springer-Verlag.

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

[14] K. L. McMillan. *Symbolic Model Checking — An Approach to the State Explosion Problem*. PhD thesis, Carnegie Mellon University, 1992.

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

[16] Lawrence C. Paulson. *ML for the Working Programmer*. Cambridge University Press, second edition, June 1996.

[17] A. Perrig, R. Canetti, J. D. Tygar, and D. Xiaodong Song. Effi cient authentication and signing of multicast streams over lossy channels. In *Proceedings of The IEEE Symposium on Security and Privacy*, pages 56–73, 2000.