Distributed Temporal Logic for the Analysis of Security Protocol Models

David Basin
Department of Computer Science
ETH Zurich, Switzerland

Carlos Caleiro Jaime Ramos
SQIG - Instituto de Telecomunicações and Department of Mathematics
IST, TU Lisbon, Portugal

Luca Viganò
Department of Computer Science
University of Verona, Italy

August 25, 2011

Abstract

The distributed temporal logic DTL is an expressive logic, well-suited for formalizing properties of concurrent, communicating agents. We show how DTL can be used as a metalogic to reason about and relate different security-protocol models. This includes reasoning about model simplifications, where models are transformed to have fewer agents or behaviors, and verifying model reductions, where to establish the validity of a property it suffices to consider its satisfaction on only a subset of models.

We illustrate how DTL can be used to formalize security models, protocols, and properties, and then present three concrete examples of metareasoning. First, we prove a general theorem about sufficient conditions for data to remain secret during communication. Second, we prove the equivalence of two models for guaranteeing message-origin authentication. Finally, we relate channel-based and intruder-centric models, showing that it is sufficient to consider models in which the intruder completely controls the network. While some of these results belong to the folklore or have been shown, mutatis mutandis, using other formalisms, DTL provides a uniform means to prove them within the same formalism. It also allows us to clarify subtle aspects of these model transformations that are often neglected or cannot be specified in the first place.

1 Introduction

Security protocols are distributed programs that employ cryptography in order to achieve their objectives in possibly hostile environments. They have been widely studied within the formal methods community as they are difficult to design and notoriously prone to error: it is difficult to predict all the possible ways that distributed computation may proceed and thereby foresee all the ways that an intruder can overcome cryptography by exploiting the willingness of honest agents to communicate. The research in this area is extensive and a number of logics and formalisms exist for specifying and verifying security protocols.

Our contribution in this paper is not another method or tool for protocol analysis. Rather, we present a logical foundation for formalizing and reasoning about models of security protocols, which are at the heart of other methods and tools. The foundation is based on DTL, a distributed temporal logic [34]. DTL is an expressive, general-purpose logic that is well-suited for formalizing

both local, agent-specific properties and global properties of distributed communicating processes. Within DTL, we formalize different theories for security protocol analysis, capturing different ways that agents can interact with the network in the presence of an active intruder. We use these theories to show how DTL can be used in two, quite distinct ways:

- as an *object logic* for formalizing specific protocol models and proving properties of protocols with respect to these models, and
- as a *metalogic* for relating different models and proving metatheorems about the models themselves.

To show how DTL can be used as an object logic, we consider the well-known NSPK protocol [54] and its corrected version NSL [42]. We use these to illustrate DTL's application to both protocol falsification and verification. As mentioned above, there are many existing formalisms for reasoning about protocols. However, unlike existing formalisms for object-level protocol analysis, DTL is additionally suitable as a metalogic, and the possibility of using DTL effectively as an object logic is a necessary prerequisite for more advanced metalogical applications. Conversely, an effective metalogic aids object logic applications in that we can derive general metatheorems useful for protocol verification. We will illustrate this by establishing a general result about sufficient conditions for data to remain secret during communication.

To further show how DTL can be used as a metalogic, we present applications to formalizing and verifying properties of security protocol models and translations between models. Our motivations here are both theoretical and practical. Within the security community, a wide variety of different models have been proposed, often with slightly differing assumptions, concerning communication, the powers of the intruder, and the abstractions used in describing security protocols. Theoretically, we would like to understand, and have a formal foundation for establishing, the relationship between these different models. Practically, when building security protocol analysis tools, we would like to reduce the number of different scenarios that must be searched (*model reduction*) as well as to simplify the scenarios considered (*model simplification*). Such optimizations can substantially improve the efficiency of algorithmic verification tools and should be employed whenever possible, provided they preserve the properties of interest. The challenge, of course, is to formalize and prove this. Our contribution is to show how DTL can be used to give a simple, rigorous, and uniform account of different model reduction and simplification techniques as well as other kinds of metatheoretic properties of security protocol models. We do this by studying concrete models, showing how they can be formalized naturally and reasoned about using DTL.

First, we formalize and prove the equivalence of two models for guaranteeing message-origin authentication. The first model is an abstract model where principals may use a special channel, controlled by a trusted third party, that logs all incoming messages and issues evidence of their origin to the recipients. The second model is more concrete and uses digital signatures to implement an authentic channel. We present two different transformations of the corresponding DTL models and corresponding notions of equivalence based on preserving translations of properties. This example is not aimed at justifying a model simplification, but rather at showing how DTL can be used to relate models at different levels of abstraction. The equivalences we prove can be used to justify that concrete (signature-based) designs achieve the more abstract notion of message-origin authentication defined in terms of a trusted third party.

Afterwards, we use DTL as a metalogic to explore the exact meaning of different modeling assumptions that are common in security protocol analysis. Namely, even when working with an intruder who controls the network, one must make numerous modeling decisions. For example, is the intruder identified with the network? That is, does he coexist with the network and intercept messages sent to the network versus is he identified with the network and all messages are sent directly to him? Alternatively, does one consider all possible interleavings between intruder actions and those of honest agents or only some subset? In particular, under suitable assumptions, we establish the following results:

(1) the intruder can be identified with the network;

(2) the steps (actions) of honest agents can be "compressed" in the sense that the receipt of a message can always be immediately followed by the agent sending a response; and

(3) all distribution in the model can be eliminated by considering all actions from the intruder's point of view.

The results (1) and (3) constitute model simplifications as they involve a translation between models (and also properties), leading to models that are simpler in the sense that they involve fewer agents or collapse behaviors. Result (2) exemplifies a model reduction, where to establish the validity of a property it suffices to consider its satisfaction on only a subset of models, namely just those models where steps are compressed.

While some of the results we prove belong to the folklore or have, mutatis mutandis, already been shown using other formalisms, our logic provides a means to prove them in a general, uniform way within the same formalism. It also allows us to clarify aspects of these properties that are often neglected or cannot be specified in the first place. For example, the equivalences proved in the message-origin example depend critically on the capabilities of the intruder using the trusted channel, what exactly is signed, and that different keys are used for different purposes.

While other logics or formalisms could be used for these tasks, we believe that DTL offers a number of advantages. Security protocols are carried out by distributed agents, with individual state, who concurrently execute protocols, synchronizing over shared communication. DTL provides a rich language with a corresponding semantics that naturally captures all of these aspects. It has a distributed dimension that captures the agents, their local state, and communication. Concurrent execution, as well as the formalization of security properties, is captured by adding a temporal dimension based on past and future time linear temporal operators. The logic is quite flexible and avoids commitment to particular models and properties, which are formalized as DTL theories. The semantics of DTL is based on interpretation structures, which are a model of concurrent, distributed systems that is well-suited for carrying out semantic reasoning about, and transformation of, such systems.

A strength of our DTL formalization is that it allows us to spell out all the fine details of the security proofs that we give in this paper. As is well known, this is particularly important in the area of Information Security, where researchers often have well-developed intuitions, but their models and proofs are prone to subtle errors. Although we have developed a tableau system for DTL [7, 8], our proofs in this paper are semantic. We prefer semantic arguments because they are shorter and far more intuitive than tableau proofs. In contrast, proofs with the tableau system can be machine checked. We note, in this regard, that the validity of DTL formulas can be decided by using a trace-consistent translation to LTL [8], which also makes DTL amenable to model checking.

We proceed as follows. In Section 2 we introduce the distributed temporal logic DTL. In Section 3, we define in DTL a protocol-independent distributed communication model, on top of which protocols and security goals can be formalized and analyzed, as shown in Section 3.3. In Section 4 we illustrate how DTL can be used as an object logic to either verify security protocols or construct counter-examples to their claimed properties. However, the results are established as a corollary of a metalevel result about secrecy that we state and prove. The core results of the paper are in Section 5, where we present the metalevel results, mentioned above. We draw conclusions and discuss related and future work in Section 6.

2 Distributed temporal logic

The *distributed temporal logic DTL* is a logic for reasoning about temporal properties of distributed systems from the local point of view of the system's agents, which are assumed to execute sequentially and to interact by synchronous event sharing. In this paper, we use DTL theories given, in *latu sensu*, by classes of models satisfying axioms and other defining conditions. Note that the set of axioms may be infinite, resulting in validity being undecidable in the theory, despite the decidability result mentioned above. This is the case for the theories that we later present.

2.1 Syntax

The logic is defined over a *distributed signature*

$$\Sigma = \langle Id, \{Act_i\}_{i \in Id}, \{Prop_i\}_{i \in Id} \rangle \,,$$

where Id is a finite set of *agent identifiers* and, for each $i \in Id$, Act_i is a set of *local action symbols* and $Prop_i$ is a set of *local state propositions*. In a nutshell, the actions Act_i correspond to true statements about an agent when they have just occurred and the state propositions $Prop_i$ characterize the current local states of the agents. Following standard protocol terminology, we will also refer to the agents participating in a protocol execution as *principals*.

The *global language* $\mathcal{L}$ is defined by the grammar

$$\mathcal{L} ::= @_{i_1}[\mathcal{L}_{i_1}] \mid \cdots \mid @_{i_n}[\mathcal{L}_{i_n}] \mid \bot \mid \mathcal{L} \Rightarrow \mathcal{L} \,,$$

for $Id = \{i_1, \ldots, i_n\}$, where the *local languages* $\mathcal{L}_i$ for each $i \in Id$ are defined by

$$\mathcal{L}_i ::= Act_i \mid Prop_i \mid \bot \mid \mathcal{L}_i \Rightarrow \mathcal{L}_i \mid \mathcal{L}_i \,\mathsf{U}\, \mathcal{L}_i \mid \mathcal{L}_i \,\mathsf{S}\, \mathcal{L}_i \mid ©_j[\mathcal{L}_j] \,,$$

with $j \in Id$. As notation, we will use γ and δ for global formulas, and φ and ψ for local formulas.

A *global formula* $@_i[\varphi]$ means that the local formula φ holds for agent i. *Local formulas*, as the name suggests, hold locally for the different agents. For instance, locally for an agent i, the operators U and S are the usual (strong) *until* and *since* temporal operators, while the *communication formula* $©_j[\psi]$ means that agent i has just communicated (synchronized) with agent j, for whom ψ held.[1]

As notation, we write $\mathcal{L}_i^{@}$ to denote the set of all purely temporal formulas of $\mathcal{L}_i$, that is, excluding communication formulas. We call $\varphi \in \mathcal{L}_i^{@}$ a *private formula*. Furthermore, if φ does not contain the temporal operators U and S, then we call it a *state formula*. Finally, we write $\mathcal{L}^{@}$ to denote the set of all global formulas built from private formulas.

Other logical connectives (conjunction, disjunction, etc.) and temporal operators can be defined as abbreviations, for example:

$\mathsf{X}\,\varphi$	$\equiv$	$\bot \,\mathsf{U}\, \varphi$	tomorrow (next)
$\mathsf{F}\,\varphi$	$\equiv$	$\top \,\mathsf{U}\, \varphi$	sometime in the future
$\mathsf{F}_\circ\,\varphi$	$\equiv$	$\varphi \vee \mathsf{F}\,\varphi$	now or sometime in the future
$\mathsf{G}\,\varphi$	$\equiv$	$\neg\,\mathsf{F}\,\neg\,\varphi$	always in the future
$\mathsf{G}_\circ\,\varphi$	$\equiv$	$\varphi \wedge \mathsf{G}\,\varphi$	now and always in the future
$\varphi \,\mathsf{W}\, \psi$	$\equiv$	$(\mathsf{G}\,\varphi) \vee (\varphi \,\mathsf{U}\, \psi)$	weak until (unless)
$\mathsf{Y}\,\varphi$	$\equiv$	$\bot \,\mathsf{S}\, \varphi$	yesterday (previous)
$\mathsf{P}\,\varphi$	$\equiv$	$\top \,\mathsf{S}\, \varphi$	sometime in the past
$\mathsf{P}_\circ\,\varphi$	$\equiv$	$\varphi \vee \mathsf{P}\,\varphi$	now or sometime in the past
$\mathsf{H}\,\varphi$	$\equiv$	$\neg\,\mathsf{P}\,\neg\,\varphi$	always in the past
$\mathsf{H}_\circ\,\varphi$	$\equiv$	$\varphi \wedge \mathsf{H}\,\varphi$	now and always in the past
$\varphi \,\mathsf{B}\, \psi$	$\equiv$	$(\mathsf{H}\,\varphi) \vee (\varphi \,\mathsf{S}\, \psi)$	weak since (back to)
$*$	$\equiv$	$\mathsf{H}\,\bot$	in the beginning
$\varphi \gg_j \psi$	$\equiv$	$\varphi \Rightarrow ©_j[\psi]$	calling

Here we use the subscript $\circ$ to denote the reflexive versions of operators. Note also that *calling* is specific to DTL as it involves communication: $@_i[\varphi \gg_j \psi]$ means that if φ holds for agent i then he calls (synchronizes with) agent j, for whom ψ must hold.

2.2 Semantics

The protocol models that we consider are based on partially-ordered sets of events with labeling information. We employ *sequences* (of events or labels) to represent protocol executions and we

[1] Note that the DTL syntax here differs slightly from the original presentation in [34]. Previously, the operator $©_i$ was overloaded with $@_i$ and its interpretation was therefore context dependent.

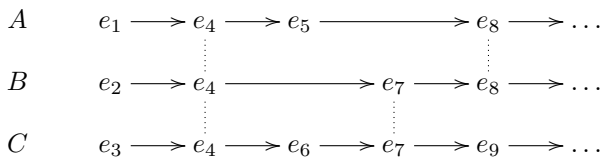


Figure 1: A distributed life-cycle for agents A, B, and C.

$$\pi_A(\emptyset) \xrightarrow{\alpha_A(e_1)} \pi_A(\{e_1\}) \xrightarrow{\alpha_A(e_4)} \pi_A(\{e_1, e_4\}) \xrightarrow{\alpha_A(e_5)} \pi_A(\{e_1, e_4, e_5\}) \xrightarrow{\alpha_A(e_8)} \dots$$

Figure 2: The progress of agent A.

write $w = \langle w_1.w_2.w_3 \dots \rangle$ to denote a possibly infinite sequence whose elements are $w_1, w_2, w_3, \dots$. Furthermore, we write $|w|$ to denote the length of the sequence w, where $|\langle\rangle| = 0$ for the empty sequence $\langle\rangle$ and $|w| = \infty$ whenever w is infinite. Finally, we write $w.w'$ to denote the concatenation of two sequences, provided that the first sequence is finite, and we write $w|_i$ to denote the prefix of w of length i, i.e. $w|_i = \langle w_1 \dots w_i \rangle$, provided that $0 \leq i \leq |w|$. Clearly, $w|_0 = \langle\rangle$.

The interpretation structures of $\mathcal{L}$ are labeled distributed life-cycles, built upon a simplified form of Winskel's *event structures* [66] (see also [67] for the relationship to other concurrency models). A *local life-cycle* of an agent $i \in Id$ is a countable (finite or infinite), discrete, and well-founded total order $\lambda_i = \langle Ev_i, \leq_i \rangle$, where Ev_i is the set of *local events* and $\leq_i$ the *local order of causality*. We define the corresponding *local successor relation* $\rightarrow_i \subseteq Ev_i \times Ev_i$ to be the relation such that $e \rightarrow_i e'$ if $e <_i e'$ and there is no e'' such that $e <_i e'' <_i e'$. As a consequence, $\leq_i = \rightarrow_i^*$, i.e., $\leq_i$ is the reflexive, transitive closure of $\rightarrow_i$.

A *distributed life-cycle* is a family $\lambda = \{\lambda_i\}_{i \in Id}$ of local life-cycles such that $\leq = (\bigcup_{i \in Id} \leq_i)^*$ defines a partial order of *global causality* on the set of all events $Ev = \bigcup_{i \in Id} Ev_i$. Note that communication is modeled by event sharing, and thus for some event e we may have $e \in Ev_i \cap Ev_j$, with $i \neq j$. In that case, requiring $\leq$ to be a partial order amounts to requiring that the local orders are globally compatible, thus excluding the existence of another $e' \in Ev_i \cap Ev_j$ such that $e <_i e'$ but $e' <_j e$.

We can check the progress of an agent by collecting all the local events that have occurred up to a given point. This yields the notion of the *local state* of agent i, which is a finite set $\xi_i \subseteq Ev_i$ down-closed for local causality, i.e., if $e \leq_i e'$ and $e' \in \xi_i$ then also $e \in \xi_i$. The set Ξ_i of all local states of an agent i is totally ordered by inclusion and has $\emptyset$ as the minimal element.

In general, each non-empty local state ξ_i is reached, by the occurrence of an event that we call $last(\xi_i)$, from the local state $\xi_i \setminus \{last(\xi_i)\}$.[2] The local states of each agent are totally ordered, as a consequence of the total order on local events. Since they are discrete and well-founded, we can enumerate them as follows: $\emptyset$ is the 0^{th} state; $\{e\}$, where e is the minimum of $\langle Ev_i, \leq_i \rangle$, is the 1^{st} state; and if ξ_i is the k^{th} state of agent i and $last(\xi_i) \rightarrow_i e$, then $\xi_i \cup \{e\}$ is agent i's $(k+1)^{\text{th}}$ state. We will denote by ξ_i^k the k^{th} state of agent i, so $\xi_i^0 = \emptyset$ is the initial state and ξ_i^k is the state reached from the initial state after the occurrence of the first k events. In fact, ξ_i^k is the only state of agent i that contains k elements, i.e., where $|\xi_i^k| = k$. Given $e \in Ev_i$, $(e{\downarrow}i) = \{e' \in Ev_i \mid e' \leq_i e\}$ is always a local state. Moreover, if ξ_i is non-empty, then $(last(\xi_i) \downarrow i) = \xi_i$. Furthermore, for every local state $\xi_i \neq Ev_i$ there exists a unique next event $next(\xi_i)$, corresponding to the minimum event in $Ev_i \setminus \xi_i$, such that $\xi_i \cup \{next(\xi_i)\}$ is a local state.

We can also define the notion of a *global state*: a finite set $\xi \subseteq Ev$ closed for global causality, i.e. if $e \leq e'$ and $e' \in \xi$, then also $e \in \xi$. The set Ξ of all global states constitutes a lattice under inclusion and has $\emptyset$ as the minimal element. Clearly, every global state ξ includes the

[2]This statement is only sensible with respect to a given distributed life-cycle. Similar comments also hold for other notions considered below. However, to ease readability, we omit explicit reference to these dependencies whenever they are clear from the context.

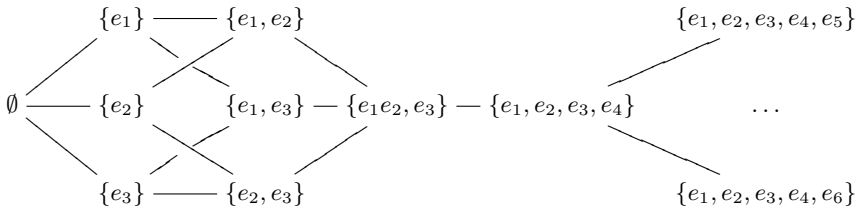


Figure 3: The lattice of global states.

local state $\xi|_i = \xi \cap Ev_i$ of each agent i. Note that we are overloading the notation $\cdot|_i$, which we previously used to denote sequence prefixing, and below we will often write ξ_i for $\xi|_i$. Given $e \in Ev$, $e{\downarrow} = \{e' \in Ev \mid e' \leq e\}$ is always a global state.

An *interpretation structure* $\mu = \langle \lambda, \alpha, \pi \rangle$ consists of a distributed life-cycle λ and a family $\alpha = \{\alpha_i\}_{i \in Id}$ and $\pi = \{\pi_i\}_{i \in Id}$ of local *labeling functions*, where, for each $i \in Id$,

- $\alpha_i : Ev_i \to Act_i$ associates a local action to each local event, and
- $\pi_i : \Xi_i \to \wp(Prop_i)$ associates a set of local state propositions to each local state.

We denote the tuple $\langle \lambda_i, \alpha_i, \pi_i \rangle$ also by μ_i.

Fig. 1 depicts a distributed life-cycle, where each row comprises the local life-cycle of one agent. In particular, $Ev_A = \{e_1, e_4, e_5, e_8, \dots\}$ and $\to_A$ corresponds to the arrows in A's row. We can think of the occurrence of the event e_1 as leading agent A from its initial state $\emptyset$ to the state $\{e_1\}$, and then of the occurrence of the event e_4 as leading to state $\{e_1, e_4\}$, and so on. The state-transition sequence of agent A is displayed in Fig. 2. Shared events at communication points are highlighted by the dotted vertical lines. Note that the numbers annotating the events are there only for convenience since, in general, no global total order on events is imposed. Fig. 3 shows the corresponding lattice of global states.

We can then define the *global satisfaction relation* by

- $\mu \Vdash \gamma$ if $\mu, \xi \Vdash \gamma$ for every $\xi \in \Xi$,

where the global satisfaction relation at a global state is defined by

- $\mu, \xi \nVdash \bot$;
- $\mu, \xi \Vdash \gamma \Rightarrow \delta$ if $\mu, \xi \nVdash \gamma$ or $\mu, \xi \Vdash \delta$;
- $\mu, \xi \Vdash @_i[\varphi]$ if $\mu_i, \xi|_i \Vdash_i \varphi$;

and where the local satisfaction relations at local states are defined by

- $\mu_i, \xi_i \Vdash_i act$ if $\xi_i \neq \emptyset$ and $\alpha_i(last(\xi_i)) = act$;
- $\mu_i, \xi_i \Vdash_i p$ if $p \in \pi_i(\xi_i)$;
- $\mu_i, \xi_i \nVdash_i \bot$;
- $\mu_i, \xi_i \Vdash_i \varphi \Rightarrow \psi$ if $\mu_i, \xi_i \nVdash_i \varphi$ or $\mu_i, \xi_i \Vdash_i \psi$;
- $\mu_i, \xi_i \Vdash_i \varphi \,\mathsf{U}\, \psi$ if $|\xi_i| = k$ and there exists $\xi_i^n \in \Xi_i$ such that $k < n$ with $\mu_i, \xi_i^n \Vdash_i \psi$, and $\mu_i, \xi_i^m \Vdash_i \varphi$ for every $k < m < n$;
- $\mu_i, \xi_i \Vdash_i \varphi \,\mathsf{S}\, \psi$ if $|\xi_i| = k$ and there exists $\xi_i^n \in \Xi_i$ such that $n < k$ with $\mu_i, \xi_i^n \Vdash_i \psi$, and $\mu_i, \xi_i^m \Vdash_i \varphi$ for every $n < m < k$;
- $\mu_i, \xi_i \Vdash_i ©_j[\varphi]$ if $|\xi_i| > 0$, $last(\xi_i) \in Ev_j$, and $\mu_j, (last(\xi_i){\downarrow} j) \Vdash_j \varphi$.

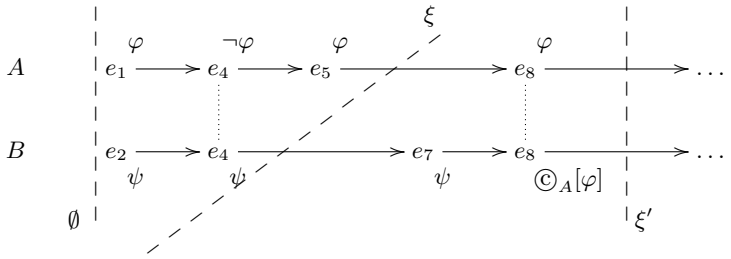


Figure 4: Satisfaction of formulas.

If $\mathcal{M}$ is a set of interpretation structures, then we write $\mathcal{M} \Vdash \varphi$ when $\mu \Vdash \varphi$ for all $\mu \in \mathcal{M}$. We will also use $\mathcal{M}_\Xi$ to denote the set of all pairs $\langle \mu, \xi \rangle$ such that $\mu \in \mathcal{M}$ and ξ is a global state of the distributed life-cycle underlying μ. We say that μ is a *model* of $\Gamma \subseteq \mathcal{L}$ if μ globally satisfies each of the formulas in Γ. We say that Γ *entails* $\gamma \in \mathcal{L}$, written $\Gamma \vDash \gamma$, if every global model of Γ is also a model of γ. Given $\Phi \cup \{\psi\} \subseteq \mathcal{L}_i$, we will write $\Phi \vDash_i \psi$ to denote that every local model of Φ is also a model of ψ, or equivalently, that $\{@_i[\varphi] \mid \varphi \in \Phi\} \vDash @_i[\psi]$.

For instance, the formula $@_i[p \Rightarrow \mathsf{F}\, ©_j[\mathsf{X}\, q]]$ holds in a model if whenever the proposition p holds locally at a state of agent i then there must be a future state of agent i where he has just synchronized with agent j, for whom q will hold in the next state.

Fig. 4 illustrates the satisfaction relation with respect to communication formulas of our running example. Clearly $\mu, \emptyset \Vdash @_B[\psi \,\mathsf{U}\, ©_A[\varphi]]$, because $\mu, \xi' \Vdash @_B[©_A[\varphi]]$. Note however that $\mu, \xi \nVdash @_B[©_A[\varphi]]$, although $\mu, \xi \Vdash @_A[\varphi]$.

We now establish general rules for reasoning about different classes of formulas.

Lemma 2.1 (Local properties) *Let $\varphi \in \mathcal{L}_i$ be a local formula and μ an interpretation structure. Let $\xi, \xi' \in \Xi$ be such that $\xi_i = \xi'_i$. Then $\mu, \xi \Vdash @_i[\varphi]$ if and only if $\mu, \xi' \Vdash @_i[\varphi]$.*

Proof: Straightforward, using the definition of global satisfaction and the assumption $\xi_i = \xi'_i$. Namely, $\mu, \xi \Vdash @_i[\varphi]$ iff $\mu_i, \xi_i \Vdash_i \varphi$ iff $\mu_i, \xi'_i \Vdash_i \varphi$ iff $\mu, \xi' \Vdash @_i[\varphi]$. □

Note that when specifying a distributed system by specifying the local properties of each agent, it makes sense to use local formulas that additionally do not have nested communication. This is because at the specification level, it is not reasonable to require that an agent in a distributed system can be aware of the communication between other agents. As shown in [34], every local DTL formula, even with nested communication, can be expressed by a finite set of local formulas without nested communication.

We now establish a result for local formulas without communication.

Lemma 2.2 (Private properties) *Let $\varphi \in \mathcal{L}_i^{\not©}$ be a private formula and μ and μ' interpretation structures with $\mu_i = \mu'_i$. Let $\xi \in \Xi$ and $\xi' \in \Xi'$ be such that $\xi_i = \xi'_i$. Then $\mu, \xi \Vdash @_i[\varphi]$ if and only if $\mu', \xi' \Vdash @_i[\varphi]$.*

Proof: We prove that $\mu_i, \xi_i \Vdash_i \varphi$ if and only if $\mu'_i, \xi'_i \Vdash_i \varphi$, by induction on φ. If φ is *act* then $\mu_i, \xi_i \Vdash_i act$ iff $\xi_i \neq \emptyset$ and $\alpha_i(last(\xi_i)) = act$ iff $\mu'_i, \xi'_i \Vdash_i act$. The last equivalence follows from $\mu_i = \mu'_i$ and $\xi_i = \xi'_i$, $\xi'_i \neq \emptyset$, and $\alpha'_i(last(\xi'_i)) = act$. The case of $p \in Prop$ is similar and the remaining cases follow by the induction hypothesis. Note that $\varphi \in \mathcal{L}_i^{\not©}$ and thus we do not consider communication formulas. The main result follows then as expected: $\mu, \xi \Vdash @_i[\varphi]$ iff $\mu_i, \xi_i \Vdash_i \varphi$ iff (from what was just proved) $\mu'_i, \xi'_i \Vdash_i \varphi$ iff $\mu', \xi' \Vdash @_i[\varphi]$. □

We also have the following invariance rule for global properties.

Proposition 2.3 (Global invariance rule) *Let $\gamma \in \mathcal{L}$ be a global formula, μ an interpretation structure, and $\xi \in \Xi$ a global state. Suppose that (1) $\mu, \xi \Vdash \gamma$ and (2) $\mu, \xi' \Vdash \gamma$ implies $\mu, \xi' \cup \{e\} \Vdash \gamma$ for every $\xi' \in \Xi$ and $e \in Ev \setminus \xi'$ such that $\xi \subseteq \xi'$ and $\xi' \cup \{e\} \in \Xi$. Then $\mu, \xi' \Vdash \gamma$, for every $\xi' \in \Xi$ such that $\xi \subseteq \xi'$.*

Proof: Let ξ' be in Ξ such that $\xi \subseteq \xi'$. The proof follows by induction on ξ'. If $|\xi'| = |\xi|$ then $\xi' = \xi$ and the result follows from the first assumption. Assume now that $|\xi'| > |\xi|$. Then, $\xi' = \xi'' \cup \{e\}$ for some $e \in Ev \setminus \xi''$. By the induction hypothesis, $\mu, \xi'' \Vdash \gamma$ and so, using the second assumption, it also follows that $\mu, \xi'' \cup \{e\} \Vdash \gamma$. □

For local state properties, the invariance rule can be stated in the following more familiar way. Its proof is similar to the proof of Proposition 2.3, *mutatis mutandis*, and we thus omit it.

Proposition 2.4 (Local invariance rule) *Let $\varphi \in \mathcal{L}_i$ be a local state formula, μ an interpretation structure, and $\xi_i \in \Xi_i$ a local state. Suppose that (1) $\mu_i, \xi_i \Vdash_i \varphi$ and (2) $\mu_i, \xi_i' \Vdash_i \varphi$ implies $\mu_i, \xi_i' \cup \{next(\xi_i')\} \Vdash \varphi$ for every $\xi_i' \in \Xi_i$ such that $\xi_i \subseteq \xi_i' \subsetneq Ev_i$. Then $\mu_i, \xi_i' \Vdash_i \varphi$, for every $\xi_i' \in \Xi_i$ such that $\xi_i \subseteq \xi_i'$, or equivalently, $\mu_i, \xi_i \Vdash_i \mathsf{G}_\circ \varphi$.*

Hence, μ is a model of $@_i[\varphi]$ if and only if μ is a model of both $@_i[* \Rightarrow \varphi]$ and $@_i[(\varphi \wedge \mathsf{X} \top) \Rightarrow \mathsf{X}\varphi]$, or equivalently, $@_i[(\varphi \wedge \mathsf{X}\, act) \Rightarrow \mathsf{X}\varphi]$ for every $act \in Act_i$.

3 Network and protocol modeling

As we remarked above, DTL supports formal specification and reasoning about models of agents communicating in distributed systems. In this paper, we focus on security protocols where principals interact by exchanging messages through an insecure public channel in an open network. We will specify this network, and protocols on top of it, by defining DTL theories over suitable signatures, which correspond to classes of models.

Security protocols describe how principals exchange messages, built using cryptographic primitives, in order to obtain security guarantees. Our presentation is independent of both the specific algebra of messages considered for the different security protocols and the actions that principals can take during protocol execution. We will thus take both the algebra of messages and the actions as parameters of our models, considering standard examples for concreteness.

More generally, protocol specifications are parametric in the sense that they prescribe a general recipe for communication that can be used by different principals playing in the different *protocols roles* (sender, responder, server, etc.). The messages transmitted are bit-strings, but, for our purposes, they can also be taken from any other appropriate set of values and our results are independent of such details. We just assume fixed a network signature.

Definition 3.1 A *network signature* is a pair $\langle Princ, Num \rangle$, where $Princ$ is a finite set of principal identifiers and $Num = Nonces \uplus SymK \uplus PubK$ is a set of "number" symbols used to model atomic data. Num is the union of three disjoint sets: $Nonces$ is a set of nonce symbols, $SymK$ is a set of symmetric key symbols, and $PubK$ is a set of public key symbols. △

We will use upper-case letters like $A, B, C, \ldots$, possibly annotated with subscripts and superscripts, to denote principals, N to denote nonces, and K to denote shared or public keys.

3.1 Messages

The algebra of messages tells us how messages are constructed. Numerous algebras have been considered in the literature on security protocol analysis, e.g. [26, 50], ranging from the free algebra to various formalizations of algebraic properties of the cryptographic operators employed. The following is a standard example of a free algebra of messages. We will use K^{-1} to denote the *private key* that is the *inverse* of a public key $K \in PubK$, and set $PrivK = \{K^{-1} \mid K \in PubK\}$.

Definition 3.2 *Messages*, which we denote by M, possibly with annotations, are built inductively from *atomic messages* (identifiers and number symbols) and private keys, by *pairing*, *encryption*, and *hashing*. For M_1 and M_2 messages, we write

- the pairing of M_1 and M_2 as $M_1; M_2$,
- the *symmetric* encryption of M_1 by M_2 as $\{\!|M_1|\!\}^s_{M_2}$,
- the *asymmetric* encryption of M_1 by $K \in PubK$ (respectively, by $K^{-1} \in PrivK$) as $\{\!|M_1|\!\}^a_K$ (respectively, $\{\!|M_1|\!\}^a_{K^{-1}}$), and
- the application of a hash function H to M_1 as $H(M_1)$.

We write Msg to denote the set of messages. △

Whenever the distinction between symmetric and asymmetric encryption is unimportant, we simply write $\{\!|M_1|\!\}_{M_2}$. As usual, we call M_2 the *key* and say that M_1 is in the *scope of the encryption*; similarly, for $H(M)$, we say that M is in the *scope of the hash function* H. Note that we will often annotate keys with principal names and we will use K both to denote the public part of the private key K^{-1} or a symmetric key. Note also that we assume that from a private key K^{-1} one can compute its public part K (see, e.g., [61]), which is equivalent to the notion of public-private key pairs that is often also considered. For this reason, we do not consider the inverse of a private key, namely $(K^{-1})^{-1}$. Observe also that, as we will enforce below, the only ways for a principal to obtain the inverse of a key are to initially know it, to receive it in a message, or when it is a private key that he freshly generated.

As is often done in symbolic approaches to security protocol analysis, we follow the *perfect cryptography assumption*, which postulates that the cryptographic primitives themselves cannot be attacked and hence the only way to decrypt a message is to possess the appropriate key. We can then define, as is standard, the sets of messages that principals can *analyze* (decompose) and *synthesize* (compose), where we have two analysis rules for asymmetric encryption: one for decrypting with a private key K^{-1} a message M that has been asymmetrically encrypted with the corresponding public key K, and one for decrypting with the public key K a message M that has been asymmetrically encrypted with a private key K^{-1} (as usual, this corresponds to verifying with key K a message that has been signed with key K^{-1}).

Definition 3.3 We write $close(S)$ to denote the closure of a set S of messages under the rules:

$$\frac{M_1; M_2}{M_1}\,\mathcal{A}_{\text{proj}_1} \qquad \frac{M_1; M_2}{M_2}\,\mathcal{A}_{\text{proj}_2} \qquad \frac{\{\!|M|\!\}^s_K \quad K}{M}\,\mathcal{A}_{\text{symm}}$$

$$\frac{\{\!|M|\!\}^a_K \quad K^{-1}}{M}\,\mathcal{A}_{\text{pub}} \qquad \frac{\{\!|M|\!\}^a_{K^{-1}} \quad K}{M}\,\mathcal{A}_{\text{priv}} \qquad \frac{K^{-1}}{K}\,\mathcal{AS}_{\text{privpub}}$$

$$\frac{M_1 \quad M_2}{M_1; M_2}\,\mathcal{S}_{\text{pair}} \qquad \frac{M}{H(M)}\,\mathcal{S}_{\text{hash}} \qquad \frac{M \quad K}{\{\!|M|\!\}^s_K}\,\mathcal{S}_{\text{symm}} \qquad \frac{M \quad K}{\{\!|M|\!\}^a_K}\,\mathcal{S}_{\text{pub}} \qquad \frac{M \quad K^{-1}}{\{\!|M|\!\}^a_{K^{-1}}}\,\mathcal{S}_{\text{priv}}$$

△

With the exception of $\mathcal{AS}_{\text{privpub}}$, all rules are standard and follow the naming convention that rules that decompose messages are labeled with an $\mathcal{A}$ for "analysis" and rules that compose messages with an $\mathcal{S}$ for "synthesis". For instance, $\mathcal{A}_{\text{pub}}$ and $\mathcal{A}_{\text{priv}}$ formalize the decryption of a message encrypted with a public key K and the signature verification of a message signed with a private key K^{-1}. The rule $\mathcal{AS}_{\text{privpub}}$ is both an analysis and a synthesis rule, as it formalizes the notion of a "publication function" in asymmetric cryptographic systems: given a private key K^{-1}, a publication function computes the corresponding public key K.

Based on these closure rules, we give the following definitions of the *content* and of the *immediate parts* of a message, which essentially invert the analysis rules and the synthesis rules, respectively. We then prove some useful properties about sets of secure messages.

Definition 3.4 The *content* of a message M is the set $cont(M)$ of messages defined inductively by

$$cont(M) = \begin{cases} \{M\} & \text{if } M \in Num, \text{ or } M = H(M_1) \text{ for some } M_1, \\ \{K^{-1}\} \cup cont(K) & \text{if } M = K^{-1} \in PrivK, \\ \{M\} \cup cont(M_1) \cup cont(M_2) & \text{if } M = M_1; M_2, \\ \{M\} \cup cont(M_1) & \text{if } M = \{\!|M_1|\!\}_K. \end{cases}$$

When $M' \in cont(M)$, we will often say that M *contains* M' or that M' *is contained in* M. △

Definition 3.5 The *immediate parts* of a message M is the set $parts(M)$ of messages defined by

$$parts(M) = \begin{cases} \emptyset & \text{if } M = N \in Nonces \text{ or } M = K \in SymK \text{ or } M = K^{-1} \in PrivK, \\ \{K^{-1}\} & \text{if } M = K \in PubK, \\ \{M_1, M_2\} & \text{if } M = M_1; M_2, \\ \{M_1, K\} & \text{if } M = \{\!|M_1|\!\}_K, \\ \{M_1\} & \text{if } M = H(M_1). \end{cases}$$

△

Let $S \subseteq Msg$ be a set of *secret* messages, i.e. messages that should not be disclosed. We can then define S-secure messages as follows.

Definition 3.6 For $S \subseteq Msg$, an *S-secure encryption* is either a symmetric encryption $\{\!|M|\!\}^s_K$ with $K \in S$ or an asymmetric encryption $\{\!|M|\!\}^a_K$ with $K^{-1} \in S$. A message M is said to be *S-secure* if each occurrence of an element of S in the content of M appears either under the scope of an S-secure encryption or under the scope of hashing. A message is *S-insecure* if it is not S-secure. We write *S-Sec* to denote the set of all S-secure messages. △

The set *S-Sec* messages contains precisely all the messages that can be safely exchanged over an insecure network without revealing the secrets in S. If a message M is S-insecure, then some element of S must appear in the content of M outside the scope of any S-secure encryption or hashing. In that case, we say that the element occurs *in the clear* (with respect to S). We can then prove the two following properties.

Proposition 3.7 *If $S \subseteq Msg$ then S-Sec $\cap S = \emptyset$.*

This is straightforward: if $M \in S$ then M is not S-secure because M itself occurs in the clear.

For reasonable sets of S of secrets, we expect that *S-Sec* is closed. For instance, it would not make sense to require $M_1; M_2$ to be a secret if one allowed both M_1 and M_2 to be disclosed.

Definition 3.8 We call a set (of secrets) $S \subseteq Msg$ *rational* if whenever S contains a message M and $parts(M) \neq \emptyset$ then $parts(M) \cap S \neq \emptyset$. △

This covers the case when S consists only of atomic data and private keys.

Proposition 3.9 *For every rational set S, we have that close(S-Sec) = S-Sec.*

Proof: Clearly, it suffices to prove that *close*(S-Sec) $\subseteq$ S-Sec. The proof builds on Proposition 3.7 and proceeds by induction on the closure rules. Many of the cases are simple and do not even rely on the rationality of the set S. Below are three of the more interesting cases.

$\mathcal{S}_{\text{pair}}$: If $M_1; M_2 \notin S$-Sec, then some element of S must occur in the clear in the content of $M_1; M_2$. Then either it occurs in the clear in the content of M_1 and therefore $M_1 \notin S$-Sec, or it occurs in the clear in the content of M_2 and therefore $M_2 \notin S$-Sec, or else $M_1; M_2$ is itself in S. Since S is a rational set, M_1 or M_2 must also be in S, and thus not in S-Sec.

$\mathcal{S}_{\text{pub}}$: If $\{\!|M|\!\}^a_K \notin S$-Sec, then some element of S must occur in the clear in the content of $\{\!|M|\!\}^a_K$. Then either it occurs in the clear in the content of M (and the asymmetric encryption is

not S-secure, in which case $K^{-1} \notin S$) and thus $M \notin S\text{-}Sec$, or $\{\!|M|\!\}^a_K$ is itself in S. Since S is a rational set, M or K must also be in S, and thus not in $S\text{-}Sec$.

$\mathcal{S}_{\text{priv}}$: If $\{\!|M|\!\}^a_{K^{-1}} \notin S\text{-}Sec$, then some element of S must occur in the clear in the content of $\{\!|M|\!\}^a_{K^{-1}}$. Then either it occurs in the clear in the content of M and thus $M \notin S\text{-}Sec$, or $\{\!|M|\!\}^a_{K^{-1}}$ is itself in S. Since S is a rational set, M or K^{-1} must also be in S, and thus not in $S\text{-}Sec$. □

3.2 A channel-based model

A *channel-based signature* Σ_{CB} is a distributed signature obtained from a network signature $\langle Princ, Num\rangle$ by taking $Id = Princ \uplus \{Ch\}$, where Ch is the communication channel (used to model asynchronous communication), and defining the action symbols and state propositions of each agent. As an example, consider the following signature of a principal A, whose actions Act_A are

- $send(M, B)$: sending the message M to B,
- $rec(M)$: receiving the message M,
- $spy(M)$: eavesdropping the message M, and
- $fresh(X)$: generating a fresh $X \in Nonces \uplus SymK \uplus PrivK$.

A's state propositions $Prop_A$ are

- $knows(M)$: A knows the message M.

For the channel, Ch, we do not require state propositions, i.e. $Prop_{Ch} = \emptyset$, whereas the actions Act_{Ch} include

- $in(A, M, B)$: the message M, sent by A, arrives on the channel, addressed to B,
- $out(A, M, B)$: the message M, sent by A, is delivered from the channel to B, and
- $leak$: leaking of a message.

These actions reflect that the underlying network may be hostile: sending actions name the intended recipient but receiving actions do not name the message's sender. In fact, we assume, as is standard, that a principal may behave as a *Dolev-Yao intruder* [33] who can compose, send, and intercept messages at will, but, following the perfect cryptography assumption, cannot break cryptography. Our results, however, are independent of the particular intruder capabilities. We use $\mathcal{L}_{CB}$ to denote the DTL language over the channel-based signature Σ_{CB}.

The network model we consider here suffices to abstractly formalize and reason about the properties of communication between principals executing security protocols, as well as about protocol models. This model could, of course, be extended in many ways. For example, we could include additional message constructors, additional actions and state propositions, or variants of the ones given, or we could even include servers and additional channels with distinct accessibility and reliability properties (see Section 5 for some examples).

In the channel-based network model CB that we define, principals can send and receive messages at will, always through the channel. If the principal A sends a message to B, then the message synchronously arrives at the channel, where it is stored for future delivery to B. If delivery ever happens, it must be synchronized with the corresponding receive action of B. However, the principal A can only send M to B if A knows both the name B and how to produce the message M. As usual, the knowledge of principals is not static. In addition to their initial knowledge, principals gain knowledge from the messages they receive and the fresh data they generate (nonces, symmetric keys, and private keys). Principals may also spy on messages leaked by the channel and learn their content. We do not allow principals to explicitly divert messages, but we also do not guarantee that messages delivered to the channel are ever received.

To ensure that principals only learn new information from the messages they receive and the fresh data they generate, we require that the *knows* proposition only holds where necessary. We restrict attention to those interpretation structures μ such that, for every principal A, the following condition holds for all messages M and non-empty local states ξ_A:

(K) $\mu, \xi_A \Vdash_A knows(M)$ iff $M \in close(\{M' \mid \mu, \xi_A \Vdash_A (\mathsf{Y}\, knows(M')) \vee rec(M') \vee spy(M') \vee fresh(M')\})$

(K) implies that, in every model $\mu = \langle \lambda, \alpha, \pi \rangle$ of the specification, π is completely determined by λ and α, given $\pi_A(\emptyset)$ for each $A \in Princ$. This is equivalent to saying that the knowledge of each principal only depends on its initial knowledge and on the actions that have occurred. A number of other useful properties follow from **(K)**, e.g., for each principal $A \in Princ$:

$$@_A[knows(M_1; M_2) \Leftrightarrow (knows(M_1) \wedge knows(M_2))] \quad \text{(K1)}$$

$$@_A[(knows(M) \wedge knows(K)) \Rightarrow knows(\{\!|M|\!\}_K)] \quad \text{(K2)}$$

$$@_A[(knows(\{\!|M|\!\}^s_K) \wedge knows(K)) \Rightarrow knows(M)] \quad \text{(K3.1)}$$

$$@_A[(knows(\{\!|M|\!\}^a_K) \wedge knows(K^{-1})) \Rightarrow knows(M)] \quad \text{(K3.2)}$$

$$@_A[knows(M) \Rightarrow \mathsf{G}_\circ\, knows(M)] \quad \text{(K4)}$$

$$@_A[rec(M) \Rightarrow knows(M)] \quad \text{(K5)}$$

$$@_A[spy(M) \Rightarrow knows(M)] \quad \text{(K6)}$$

$$@_A[fresh(X) \Rightarrow knows(X)] \quad \text{(K7)}$$

To guarantee the freshness and uniqueness of the data generated by each principal, we also require the following axioms, where M ranges over all messages such that $cont(X) \cap cont(M) \neq \emptyset$.

(F1) $@_A[fresh(X) \Rightarrow \mathsf{Y} \neg\, knows(M)]$

(F2) $@_A[fresh(X)] \Rightarrow \bigwedge_{B \in Princ \setminus \{A\}} ©_B[\neg\, knows(M)]$

Together with (K7), **(F1)** and **(F2)** guarantee that every fresh data item is generated at most once, if at all, in each model, and always freshly (also taking into account agents' initial knowledge). The specification of the network model also contains axioms that characterize the behavior of the channel Ch and of each principal $A \in Princ$.

(C1) $@_{Ch}[in(A, M, B) \gg_A send(M, B)]$

(C2) $@_{Ch}[out(A, M, B) \Rightarrow \mathsf{P}\ in(A, M, B)]]$

(C3) $@_{Ch}[out(A, M, B) \gg_B rec(M)]$

(C4) $@_{Ch}[leak \Rightarrow (\bigvee_{B \in Princ} ©_B[\top])]$

(P1) $@_A[send(M, B) \Rightarrow \mathsf{Y}(knows(M) \wedge knows(B))]$

(P2) $@_A[send(M, B) \gg_{Ch} in(A, M, B)]$

(P3) $@_A[rec(M) \gg_{Ch} (\bigvee_{C \in Princ} out(C, M, A))]$

(P4) $@_A[spy(M) \gg_{Ch} (leak \wedge \mathsf{P} \bigvee_{B,C \in Princ} in(B, M, C))]$

(P5) $@_A[\bigwedge_{B \in Princ \setminus \{A\}} \neg ©_B[\top]]$

(P6) $@_A[fresh(X) \Rightarrow \neg\, ©_{Ch}[\top]]$

The channel axioms **(C1)**–**(C3)** are straightforward. They state that a message addressed to A only arrives at the channel if it is sent to A by some principal B; the channel only delivers a message to A if the message for A previously arrived; and if the channel delivers a message to A, then A receives it. **(C4)** states that when the channel is leaking, some principal is listening.

The principal axioms are also simple. **(P1)** states a precondition for sending a message: the sender must know both the message and the recipient beforehand. **(P2)**–**(P3)** are interaction axioms. **(P2)** and **(P3)** state that the sending and receiving of messages must be shared with the corresponding arrival and delivery actions of the channel. **(P4)** guarantees that a spied message must have arrived at the channel, addressed to some recipient. The last two axioms limit the possible interactions: **(P5)** guarantees that principals never communicate directly (only through the channel) and **(P6)** states that actions that generate fresh data are not communication actions.

As our aim is to provide a foundation for modeling security protocols, we will further add, for simplicity, a number of standard restrictions. To start with, we assume there exists a special principal $Z \in \mathit{Princ}$, also known as the *intruder*. As is well-known, it suffices to consider one Dolev-Yao intruder, instead of several ones. This can be formally proved using DTL, which we have done in [16] by showing that *one intruder is enough* (along the lines of the "two (honest) agents are sufficient" result of [24]). We define the set of *honest principals* to be $\mathit{Hon} = \mathit{Princ} \setminus \{Z\}$.[3] To make sense of the terminology, we must of course ensure that honest principals do not act dishonestly, namely by spying messages. Thus, for every $A \in \mathit{Hon}$, we require also that:

(Hon) $@_A[\neg\, \mathit{spy}(M)]$, for every message M.

We further assume that Z does not send a message to principal A if A will not receive it, or if that same message has already been sent to A. Namely, we assume the following axioms:

(EZ1) $@_Z[\mathit{send}(M, A) \Rightarrow \copyright_{Ch}[\mathsf{F}\, \mathit{out}(Z, M, A)]]$ and

(EZ2) $@_Z[\mathit{send}(M, A) \Rightarrow \copyright_{Ch}[\neg\, \mathsf{P} \bigvee_{B \in \mathit{Princ}} \mathit{in}(B, M, A)]]$.

This represents a common simplification, which can be made without loss of generality. As shown, for instance in [16], such assumptions do not compromise the model with respect to its ability to represent security-sensitive behaviors, as we will introduce below.

The CB models μ will be those interpretation structures over Σ_{CB} satisfying all these properties.

3.3 Modeling security protocols

In this section, we show how to model protocols and properties on top of our channel-based network model. In a typical situation, we will assume a network signature where each principal $A \in \mathit{Princ}$ is assigned a private key denoted by K_A^{-1}, whose corresponding public key is the atom K_A. Whereas it is possible (even desirable) that other principals know A's public key, we will assume that, at least initially, K_A^{-1} is known only by A. We formalize this as follows:

(aKey1) $@_A[* \Rightarrow \mathit{knows}(K_A^{-1})]$

(aKey2) $@_B[* \Rightarrow \neg\, \mathit{knows}(M)]$, for every $B \in \mathit{Princ} \setminus \{A\}$ and every M containing K_A^{-1}

Similarly, we may assume that there exist symmetric shared atomic keys K_{AB} for each pair of principals $A, B \in \mathit{Princ}$. As above, we require that K_{AB} is initially known only by A and B:

(sKey1.1) $@_A[* \Rightarrow \mathit{knows}(K_{AB})]$

(sKey1.2) $@_B[* \Rightarrow \mathit{knows}(K_{AB})]$

(sKey2) $@_C[* \Rightarrow \neg\, \mathit{knows}(M)]$, for every $C \in \mathit{Princ} \setminus \{A, B\}$ and every M containing K_{AB}

[3] Given this distinction between the intruder and the honest participants, we could rewrite several of the axioms by distinguishing the nature of the principals involved, but we refrain from doing so for brevity.

We may also assume, for simplicity, that all principals $A, B \in \mathit{Princ}$ know each other's names and public keys from the very beginning.

(N) $@_A[* \Rightarrow \mathit{knows}(B)]$

(PK) $@_A[* \Rightarrow \mathit{knows}(K_B)]$

Such properties may influence the executability of the protocol by the participants, as we have also discussed in [18]. Here, we will simply assume that the initial knowledge of the principals guarantees that the protocols can be executed.

There are several approaches to extracting formal protocol specifications from a protocol description in Alice-and-Bob-style notation, i.e., as a sequence of message exchange steps. Rather than going through the general case, we will illustrate the method by modeling the standard example of the (flawed) simplified Needham-Schroeder Public Key Protocol NSPK [42]. The formalization steps we take are straightforward and they would not be difficult to generate automatically from such an Alice-and-Bob-style protocol description, as explained, for instance, in [18, 39, 44, 50].

The NSPK protocol can be described by the following sequence of message exchanges.

$$\begin{array}{llcll} (msg_1) & a \to b & : & (n_1). & \{\!|n_1; a|\!\}^a_{K_b} \\ (msg_2) & b \to a & : & (n_2). & \{\!|n_1; n_2|\!\}^a_{K_a} \\ (msg_3) & a \to b & : & & \{\!|n_2|\!\}^a_{K_b} \end{array}$$

In this notation, a and b are variables identifying the principals playing in the different protocol roles (initiator and responder), n_1 and n_2 are variables representing the nonces created by these principals, and the arrows represent communication from the sender to the receiver. The parenthesized nonces prefixing the first two messages signify that these nonces are freshly generated before the message is sent. Moreover, it is assumed that the principals' public keys have been distributed before the protocol starts (or else a and b would not be able to construct the messages). This can be straightforwardly expressed by appropriate instances of the axioms **(N)** and **(PK)**.

Formalizing a protocol like the above involves defining the sequences of actions (*send*, *rec*, and *fresh*) taken by agents executing each protocol role. Specifically, given concrete principals A and B and fresh nonces N_1 and N_2, the role instantiations should correspond to the execution, by principal A, of the sequence of actions $\mathit{run}_A^{\mathit{Init}}(A, B, N_1, N_2)$:

$$\langle \mathit{fresh}(N_1).\mathit{send}(\{\!|N_1; A|\!\}^a_{K_B}, B).\mathit{rec}(\{\!|N_1; N_2|\!\}^a_{K_A}).\mathit{send}(\{\!|N_2|\!\}^a_{K_B}, B)\rangle\,,$$

and to the execution, by principal B, of the sequence $\mathit{run}_B^{\mathit{Resp}}(A, B, N_1, N_2)$:

$$\langle \mathit{rec}(\{\!|N_1; A|\!\}^a_{K_B}).\mathit{fresh}(N_2).\mathit{send}(\{\!|N_1; N_2|\!\}^a_{K_A}, A).\mathit{rec}(\{\!|N_2|\!\}^a_{K_B})\rangle\,.$$

In general, an Alice-and-Bob-style protocol description P may involve j principal identifier variables $a_1, \ldots, a_j$, corresponding to j distinct roles, and k fresh data variables $f_1, \ldots, f_k$ (standing for the freshly created nonces, symmetric keys, or private asymmetric keys), and consist of a sequence $\langle msg_1 \ldots msg_m \rangle$ of message exchanges, each of the form

$$(msg_q) \quad a_s \to a_r \;:\; (f_{q_1}, \ldots, f_{q_t}).\; M\,,$$

where M can include any of the principal identifiers and fresh data variables.

A *protocol instantiation* is a variable substitution σ such that each $\sigma(a_i) \in \mathit{Princ}$ and each $\sigma(f_i) \in \mathit{Nonces} \uplus \mathit{SymK} \uplus \mathit{PrivK}$. Moreover, while the intruder can play different roles in a protocol instantiation, we require that honest agents do not play two roles in the same instantiation. Of course, this does not prevent the same honest agent from playing the same or other roles in other protocol instantiations. Hence, if $\sigma(a_{i_1}), \sigma(a_{i_2}) \in \mathit{Hon}$ and $\sigma(a_{i_1}) = \sigma(a_{i_2})$ then $i_1 = i_2$. We extend σ to messages, actions, sequences, formulas, and indices in the natural way. For example, $\sigma(K_{a_i}) = K_{\sigma(a_i)}$. Each instantiation prescribes a concrete sequence of actions to be

executed by each participant in a protocol run: for each role i, we have the corresponding sequence $run^i = msg^i_1 \centerdot \cdots \centerdot msg^i_m$ where

$$msg^i_q = \begin{cases} \langle fresh(f_{q_1}) \ldots fresh(f_{q_t}).send(M, a_r)\rangle & \text{if } i = s, \\ \langle rec(M)\rangle & \text{if } i = r, \\ \langle\rangle & \text{if } i \neq s \text{ and } i \neq r. \end{cases}$$

If $\sigma(a_i) = A$, we write $run^i_A(\sigma) = \sigma(run^i)$. We can easily formalize the complete execution by principal A of the run corresponding to role i of the protocol, under the protocol instantiation σ. If $run^i_A(\sigma) = \langle act_1 \ldots act_n\rangle$ then we can formalize A's execution by the local formula $role^i_A(\sigma)$:

$$act_n \wedge \mathsf{P}(act_{n-1} \wedge \mathsf{P}(\ldots \wedge \mathsf{P}\, act_1) \ldots)\,.$$

In general, if we denote the set of all protocol instantiations by $Inst$, we can define the set $Runs^i_A$ of all possible concrete runs of principal A in role i, and the set $Runs_A$ of all of A's possible concrete runs in any of the j roles:

$$Runs^i_A = \bigcup_{\sigma \in Inst} \{run^i_A(\sigma) \mid \sigma(a_i) = A \in Princ\} \qquad \text{and} \qquad Runs_A = \bigcup_{i=1}^{j} Runs^i_A\,.$$

It should be clear that $\mu, \xi \Vdash @_A[role^i_A(\sigma)]$ if and only if A has just completed the required sequence of actions $run^i_A(\sigma)$ at ξ. Often, in examples, we will use $\overline{a} = \langle a_1 \ldots a_j\rangle$ and $\overline{f} = \langle f_1 \ldots f_k\rangle$, and write $run^i_A(\sigma(\overline{a}), \sigma(\overline{f}))$ instead of $run^i_A(\sigma)$, and $role^i_A(\sigma(\overline{a}), \sigma(\overline{f}))$ instead of $role^i_A(\sigma)$.

In addition to the assumption that no honest agent ever plays two different roles in the same run, we also require that honest principals strictly follow the protocol. Therefore, if the local life-cycle of $A \in Hon$ is $e_1 \rightarrow_A e_2 \rightarrow_A e_3 \rightarrow_A \ldots$, we require that the corresponding (possibly infinite) sequence of actions

$$w(A) = \langle \alpha_A(e_1).\alpha_A(e_2).\alpha_A(e_3) \ldots\rangle$$

must be an interleaving of prefixes of sequences in $Runs_A$, but using distinct fresh data in each of them. Formally, we say that two distinct sequences of actions w and w' are *independent* provided that if $w_i = fresh(X)$ for some $i \leq |w|$ and some X, and $w'_j = fresh(Y)$ for some $j \leq |w'|$ and some Y, then $X \neq Y$. The requirement on protocol models can now be rigorously defined. For each $A \in Hon$, there must exist a set $W \subseteq Runs_A$ of pairwise independent sequences such that for every $i \leq |w(A)|$ it is possible to choose $w \in W$, $j \leq |w|$, and $i_1 < \cdots < i_j = i$ satisfying $w(A)_{i_k} = w_k$ for all k, where $k \leq j$. We will use the protocol name P to denote the resulting set of models.

Note that this is similar to approaches such as [55], where the behavior of an honest agent A is defined inductively so that the ith action of a sequence $w \in Runs_A$ can be executed only if the previous $i - 1$ actions have been executed. It is also similar to strand spaces [17, 62, 63] where essentially the same sequences of $Runs_A$ are used to model honest agents. In all cases, the intruder can act freely, according to the standard Dolev-Yao capabilities.

In the case of NSPK models, the life-cycle of each honest agent must be built by interleaving prefixes of sequences of the form $run^{Init}_A(A, B', N_1, N_2)$ or $run^{Resp}_A(B', A, N_1, N_2)$, where no two such initiator runs can have the same N_1, no two responder runs can have the same N_2, and the N_1 of an initiator run must be different from the N_2 of any responder run.

3.4 Security goals

The aim of security protocol analysis is to prove (or disprove) the correctness of a protocol with respect to the security goals that the protocol should achieve. For instance, the *secrecy* of the critical data exchanged during a protocol's execution is one such goal. In addition, an honest principal running the protocol may wish to *authenticate* the identities of its protocol partners based

on the messages he receives. There are many approaches to specifying secrecy and authentication in the literature, depending in part on the underlying model used. However, the various approaches usually agree on the general picture. Below, we show how to formulate secrecy and authentication goals for protocols in the general case and use the NSPK protocol as an illustration.

As usual, given a protocol and a security goal, we call an *attack* any protocol model μ and state ξ for which the formula expressing the goal does not hold. Let us start with secrecy.

3.4.1 Secrecy

We formalize that the messages in a finite set S will remain a shared secret between the participants $A_1, \ldots, A_j$ after the complete execution of a protocol under the instantiation σ, with each $\sigma(a_i) \in \mathit{Princ}$, by the formula $\mathit{secr}_S(\sigma)$:

$$\bigwedge_{i=1}^{j} @_{A_i}[\mathsf{P}_\circ\ \mathit{role}^i_{A_i}(\sigma)] \Rightarrow \bigwedge_{B \in \mathit{Princ} \setminus \{A_1,\ldots,A_j\}} \bigwedge_{M \in S} @_B[\neg\, \mathit{knows}(M)].$$

Of course, this property can be expected to hold only in particular situations. Assume that all the participants are honest, that is, each $A_i \in \mathit{Hon}$. One might then expect that the "critical" fresh data generated during the run will remain a secret shared only by the participating principals. Indeed, being honest, they will not reuse this fresh data in subsequent protocol runs. Using the logic, we can check the property $\mathit{secr}_{\sigma(F)}(\sigma)$ for the relevant set of fresh data variables $F \subseteq \{f_1, \ldots, f_k\}$. As before, we sometimes write $\mathit{secr}_{\sigma(F)}(\sigma(\overline{a}), \sigma(\overline{f}))$ instead of $\mathit{secr}_{\sigma(F)}(\sigma)$.

In the case of the NSPK protocol, this amounts to requiring that $\mathit{secr}_{\{N_1,N_2\}}(A, B, N_1, N_2)$ holds, with A and B both honest.

3.4.2 Authentication

There are many possible notions of authentication, e.g., the authentication hierarchy of [43]. In most cases, authentication formalizes some kind of correspondence property between the messages an agent receives in a protocol run and the messages that the other participants of the same run are supposed to have sent. The typical (weak) authentication goal states that if an honest principal A completes his part of a run of a protocol in role i, with certain partners and data, then it must be the case that these partners have actually sent to A the messages that A received.

Let σ be a protocol instantiation such that $\sigma(a_i) = A \in \mathit{Hon}$ and $\sigma(a_j) = B \in \mathit{Princ}$. Then the property that A authenticates B in role j at message q of the protocol can be defined in our logic by the formula $\mathit{auth}^{i,j,q}_{A,B}(\sigma)$, which is

$$@_A[\mathit{role}^i_A(\sigma)] \Rightarrow @_B[\mathsf{P}_\circ\ \mathit{send}(\sigma(M), A)]\,,$$

assuming that the protocol message msg_q requires that a_j sends the message M to a_i. We would therefore require $\mathit{auth}^{i,j,q}_{A,B}(\sigma)$ to hold whenever message q is considered essential for authentication. As before, we sometimes write $\mathit{auth}^{i,j,q}_{A,B}(\sigma(\overline{a}), \sigma(\overline{f}))$ instead of $\mathit{auth}^{i,j,q}_{A,B}(\sigma)$.

In the case of the NSPK protocol we can specify, for an honest principal A acting as initiator, the authentication of the responder B at message 2 using $\mathit{auth}^{\mathit{Init},\mathit{Resp},2}_{A,B}(A, B, N_1, N_2)$ as

$$@_A[\mathit{role}^{\mathit{Init}}_A(A, B, N_1, N_2)] \Rightarrow @_B[\mathsf{P}_\circ\, \mathit{send}(\{|N_1; N_2|\}^a_{K_A}, A)]\,.$$

Analogously, for an honest principal B acting as responder, the authentication of the initiator A at message 3 using $\mathit{auth}^{\mathit{Resp},\mathit{Init},3}_{B,A}(A, B, N_1, N_2)$ is

$$@_B[\mathit{role}^{\mathit{Resp}}_B(A, B, N_1, N_2)] \Rightarrow @_A[\mathsf{P}_\circ\, \mathit{send}(\{|N_2|\}^a_{K_B}, B)]\,.$$

This last property fails due to the man-in-the-middle attack on NSPK [42], as we show below.

4 A meta-level secrecy result and object-level applications

In this section, we show how DTL can be used as an object logic for both protocol verification and falsification. In particular, we show that the responder fails to authenticate the initiator in the NSPK protocol, formalized in the previous section. Afterwards, we verify Lowe's corrected version of this protocol. For this proof, however, we leverage DTL as a metalogic and first prove a general metatheorem about sufficient conditions for data to remain secret during communication. Our verification of the corrected protocol follow as a direct application of this general result.

4.1 Secret data

The result that we will prove is a good example of the kind of meta-level property that any suitable network model should enjoy. We use the properties of secure messages proved in Section 3 to reason about secrecy in a protocol-independent way and afterwards prove a proposition about secrecy properties in protocol models.

Let $S \subseteq \mathit{Msg}$ be a set of *secret* messages, that is, messages that should not be disclosed. The following lemma states that, given a group of principals $G \subseteq \mathit{Princ}$, if all the fresh data in S originates from principals in G, then the secrets in S will remain unknown outside of G as long as the principals in G only send S-secure messages.

Lemma 4.1 (Secret Data) *Let S be a rational set of messages, $G \subseteq \mathit{Princ}$ be a group of principals, and μ be a network model such that*

$$\mu \Vdash \bigvee_{A \in G} @_A[* \Rightarrow (\mathit{knows}(X) \vee \mathsf{F}\, \mathit{fresh}(X))] \text{ for each } X \in (\mathit{Nonces} \uplus \mathit{SymK} \uplus \mathit{PrivK}) \cap S. \quad \text{(i)}$$

Given a global state ξ of μ with

$$\mu, \xi \Vdash \bigwedge_{B \in \mathit{Princ} \setminus G} @_B[\neg\, \mathit{knows}(M)] \text{ for every } M \notin S\text{-}\mathit{Sec} \quad \text{(ii)}$$

then, for every global state $\xi' \supseteq \xi$ of μ, either

$$\mu, \xi' \Vdash \bigwedge_{B \in \mathit{Princ} \setminus G} @_B[\neg\, \mathit{knows}(M)] \text{ for every } M \notin S\text{-}\mathit{Sec} \quad \text{(iii)}$$

or else there exists $M \notin S$-Sec such that

$$\mu, \xi' \Vdash \bigvee_{A \in G, C \in \mathit{Princ}} @_A[\mathsf{P}_\circ\, \mathit{send}(M, C)]. \quad \text{(iv)}$$

Proof: We assume (i) and (ii), and prove either (iii) or (iv) for every global state $\xi' \supseteq \xi$ of μ. The proof proceeds by induction, using the global invariance rule of Proposition 2.3. The rule's base case (1), with $\xi' = \xi$, is trivial, as in this case (iii) coincides (ii). We turn to the step case (2), and must show that assuming, by induction hypothesis, that either (iii) or (iv) hold for ξ', then (iii) or (iv) must also hold for any extended global $\xi' \cup \{e\}$.

By definition, if (iv) holds for ξ', then it also holds for $\xi' \cup \{e\}$. Thus, we are left with proving that if (iii) holds for ξ', then (iii) or (iv) hold for $\xi' \cup \{e\}$. Suppose then that (iii) holds for ξ' but not for $\xi' \cup \{e\}$, that is $\mu, \xi' \cup \{e\} \Vdash @_B[\mathit{knows}(M)]$ for some S-insecure message M and some $B \notin G$. Then, by Lemma 2.1, it must be the case that $e \in \mathit{Ev}_B$, and so the local states of all other principals do not change (see axiom **(P5)**). Moreover, $\alpha_B(e)$ cannot be a sending action since this would not change the knowledge of principal B (see condition **(K)**). If $\alpha_B(e)$ was either $\mathit{rec}(M')$ or $\mathit{spy}(M')$ then, using Proposition 3.9 and the assumption that (iii) holds for ξ', it would follow that $M' \notin S\text{-}\mathit{Sec}$. However, since M' must have been previously sent to the channel (by the axioms **(P3)**, **(C2)**, **(P4)**, and **(C1)**), axiom **(P1)** implies that such a message could only have been sent by some $A \in G$. Hence, one would have $\mu, \xi' \cup \{e\} \Vdash \bigvee_{A \in G, C \in \mathit{Princ}} @_A[\mathsf{P}_\circ\, \mathit{send}(M', C)]$, i.e., (iv) would hold for $\xi' \cup \{e\}$. The only remaining possibility is a *fresh* action. However, it cannot be $\mathit{fresh}(X)$ for some $X \in S$, independently of whether X is a nonce, a symmetric key, or a private key, as this, together with the freshness conditions (**(F1-2)**, **(K7)**, (**aKey1**), (**sKey1.1 − 2**)) and the fact that $B \notin G$, would contradict condition (i), and the result follows. □

Note that the set $Msg \setminus S\text{-}Sec$ of S-insecure messages corresponds to what is called an *ideal* in the context of strand spaces [62]. However, here we have defined it in a more general setting that also includes composed keys and hashing. Similarly, the set $S\text{-}Sec$ of S-secure messages corresponds to a *coideal* in the terminology of [27, 47].

Lemma 4.1 above is a general, protocol-independent result about the network data flow. It can be used to reason about secrecy properties in protocol models and is similar to results obtained for PCL [56], which in turn generalize those found in [16, 27, 47], e.g., to include composed keys and hashing. Indeed, under reasonable conditions, the secrecy of freshly generated data can easily be seen to hold. Recall that we assume that each principal A initially has a private key, denoted by K_A^{-1}, and that each pair of principals A and B initially shares a symmetric key, denoted by K_{AB}, subject to the key axioms (**aKey1**), (**aKey2**), (**sKey1.1**), (**sKey1.2**) and (**sKey2**) given in Section 3.3. If the protocol at hand does not require the existence of some of these initially distributed keys, then the result still holds if we simply remove the unused keys from the set S.

Proposition 4.2 (Secrecy) *A protocol guarantees $secr_{\sigma(F)}(\sigma)$ for an instantiation σ of the fresh data $\sigma(F)$ generated in a protocol run by honest participants $\sigma(a_1) = A_1$, ..., $\sigma(a_j) = A_j$ provided that all the messages ever sent by $A_1, \ldots, A_j$ in any protocol run are S-Secure, for $S = (\{K_{A_i}^{-1} \mid 1 \leq i \leq j\} \cup \{K_{A_iA_k} \mid 1 \leq i, k \leq j, i \neq k\} \cup \sigma(F))$.*

Proof: The result follows from Lemma 4.1 using $G = \{A_1, \ldots, A_j\}$. To begin with, S is a rational set. Let μ be a network model and ξ a global state and assume that $\mu, \xi \Vdash \bigwedge_{i=1}^{j} @_{A_i}[\mathsf{P}_\circ\ role_{A_i}^i(\sigma)]$. Condition (i) of the lemma follows as all the corresponding roles of the protocol have been completed and therefore all fresh data in $\sigma(F)$ is generated in μ among the principals in G. Moreover, for the initial state $\emptyset$, clearly, no principal outside G knows S-insecure messages. For fresh nonces, condition (ii) follows directly from axioms (**F1** $-$ **2**). For private and shared keys, condition (ii) follows from the axioms (**aKey2**) and (**sKey2**). Using the lemma, we then have that (iii) or (iv) must hold for any global state of μ. However, the assumption that $A_1, \ldots, A_j$ only send S-secure messages rules out the possibility that condition (iv) of Lemma 4.1 ever holds. Hence, (iii) must be the case also at ξ, i.e., $\mu, \xi \Vdash \bigwedge_{B \in Princ \setminus G} @_B[\neg\, knows(M)]$ for every $M \notin S\text{-}Sec$. It follows that $\mu, \xi \Vdash \bigwedge_{B \in Princ \setminus \{A_1, \ldots, A_j\}} \bigwedge_{M \in \sigma(F)} @_B[\neg\, knows(M)]$, and the secrecy property holds. □

Note that our assumption that all the messages sent by $A_1, \ldots, A_j$ in any protocol run are S-secure corresponds to the notion of *discreetness* of [27, 47].

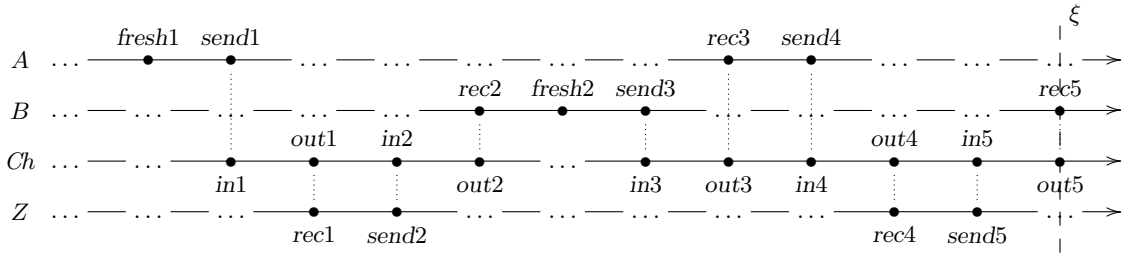
4.2 Object-level analysis of NSPK

We will now illustrate the use of DTL as an object logic using two well-known examples: the NSPK protocol and its corrected version by Lowe.

4.2.1 Protocol falsification

Recall that in the previous section, we formalized the NSPK protocol and the property that the responder authenticates the initiator: $auth_{B,A}^{Resp,Init,3}(A, B, N_1, N_2)$. We now show that this cannot be proved by presenting a model that falsifies it. The model formalizes the well-known man-in-the-middle-attack discovered by Lowe [42], where the intruder Z makes B believe he has run the protocol with A. It illustrates how concrete execution scenarios can be formalized as DTL models.

Figure 5 presents this model. It is straightforward to see that it satisfies all of the axioms of the channel model presented in Section 3. For example, (**C1**) and (**P2**) are globally satisfied because every *in* action is synchronized with a corresponding *send* action by the same principal and vice versa. Another example is (**P1**), which says that only known messages are sent. This holds for each of the 5 *send* actions. For example, for *send*2, the intruder must know $\{\!|N_1; A|\!\}_{K_B}^a$, which is the case because he previously received $\{\!|N_1; A|\!\}_{K_Z}^a$, which he can decrypt, and because he also knows public keys, in particular K_B. Additionally, the antecedent of our authentication



where

$$
\begin{array}{ll}
\mathit{fresh}1 = \mathit{fresh}(N_1) & \mathit{fresh}2 = \mathit{fresh}(N_2) \\
\mathit{send}1 = \mathit{send}(\{\!|N_1, A|\!\}^a_{K_Z}, Z) & \mathit{in}1 = \mathit{in}(A, \{\!|N_1, A|\!\}^a_{K_Z}, Z) \\
\mathit{out}1 = \mathit{out}(A, \{\!|N_1, A|\!\}^a_{K_Z}, Z) & \mathit{rec}1 = \mathit{rec}(\{\!|N_1, A|\!\}^a_{K_Z}) \\
\mathit{send}2 = \mathit{send}(\{\!|N_1, A|\!\}^a_{K_B}, B) & \mathit{in}2 = \mathit{in}(Z, \{\!|N_1, A|\!\}^a_{K_B}, B) \\
\mathit{out}2 = \mathit{out}(Z, \{\!|N_1, A|\!\}^a_{K_B}, B) & \mathit{rec}2 = \mathit{rec}(\{\!|N_1, A|\!\}^a_{K_B}) \\
\mathit{send}3 = \mathit{send}(\{\!|N_1, N_2|\!\}^a_{K_A}, A) & \mathit{in}3 = \mathit{in}(B, \{\!|N_1, N_2|\!\}^a_{K_A}, A) \\
\mathit{out}3 = \mathit{out}(B, \{\!|N_1, N_2|\!\}^a_{K_A}, A) & \mathit{rec}3 = \mathit{rec}(\{\!|N_1, N_2|\!\}^a_{K_A}) \\
\mathit{send}4 = \mathit{send}(\{\!|N_2|\!\}^a_{K_Z}, Z) & \mathit{in}4 = \mathit{in}(A, \{\!|N_2|\!\}^a_{K_Z}, Z) \\
\mathit{out}4 = \mathit{out}(A, \{\!|N_2|\!\}^a_{K_Z}, Z) & \mathit{rec}4 = \mathit{rec}(\{\!|N_2|\!\}^a_{K_Z}) \\
\mathit{send}5 = \mathit{send}(\{\!|N_2|\!\}^a_{K_B}, B) & \mathit{in}5 = \mathit{in}(Z, \{\!|N_2|\!\}^a_{K_B}, B) \\
\mathit{out}5 = \mathit{out}(Z, \{\!|N_2|\!\}^a_{K_B}, B) & \mathit{rec}5 = \mathit{rec}(\{\!|N_2|\!\}^a_{K_B})
\end{array}
$$

Figure 5: Man-in-the-middle attack to NSPK

formula, $@_B[\mathit{role}_B^{\mathit{Resp}}(A, B, N_1, N_2)]$, is satisfied at the global state ξ since the responder B has completed his role. In particular, *rec*2, *fresh*2, *send*3, and *rec*5 are the actions that are required by B to complete his role. However, the consequent $@_A[\mathsf{P}_\circ\, \mathit{send}(\{\!|N_2|\!\}^a_{K_B}, B)]$ is not satisfied at ξ because there is no previous action $\mathit{send}(\{\!|N_2|\!\}^a_{K_B}, B)$. Hence, this model is a counter-example to our authentication property as it shows that NSPK $\not\models \mathit{auth}_{B,A}^{\mathit{Resp},\mathit{Init},3}(A, B, N_1, N_2)$.

4.2.2 Protocol verification

We now show that the authentication property $\mathit{auth}_{B,A}^{\mathit{Resp},\mathit{Init},3}(A, B, N_1, N_2)$ for honest B, which failed for NSPK, actually holds for the NSL protocol [42] given below:

$$
\begin{array}{llcll}
(msg_1) & a \to b & : & (n_1). & \{\!|n_1; a|\!\}^a_{K_b} \\
(msg_2) & b \to a & : & (n_2). & \{\!|n_1; n_2; b|\!\}^a_{K_a} \\
(msg_3) & a \to b & : & & \{\!|n_2|\!\}^a_{K_b}
\end{array}
$$

Again, we have two roles: an initiator role *Init*, represented by a, and a responder role *Resp*, represented by b. Given principals A and B and nonces N_1 and N_2, the role instantiations should correspond to the execution by principal A of the sequence of actions $\mathit{run}_A^{\mathit{Init}}(A, B, N_1, N_2)$:

$$\langle \mathit{fresh}(N_1).\mathit{send}(\{\!|N_1; A|\!\}^a_{K_B}, B).\mathit{rec}(\{\!|N_1; N_2; B|\!\}^a_{K_A}).\mathit{send}(\{\!|N_2|\!\}^a_{K_B}, B)\rangle\,,$$

and to the execution by principal B of the sequence $\mathit{run}_B^{\mathit{Resp}}(A, B, N_1, N_2)$:

$$\langle \mathit{rec}(\{\!|N_1; A|\!\}^a_{K_B}).\mathit{fresh}(N_2).\mathit{send}(\{\!|N_1; N_2; B|\!\}^a_{K_A}, A).\mathit{rec}(\{\!|N_2|\!\}^a_{K_B})\rangle\,.$$

Prior to showing in Proposition 4.5 that this protocol authenticates the initiator, we prove two lemmas. The first lemma allows us to conclude that if A is also honest and does not send the required message $\{\!|N_2|\!\}^a_{K_B}$ to B, then no other agent would ever have the means to do it either.

Lemma 4.3 *Let $A, B \in Hon$, μ be a NSL model, and ξ a global state such that*

$$\mu, \xi \Vdash @_B[\mathit{fresh}(N_2) \wedge \mathsf{F}\, \mathit{send}(\{\!|N_1; N_2; B|\!\}^a_{K_A}, A)].$$

For every global state $\xi' \supseteq \xi$, if

$$\mu, \xi' \nVdash @_A[\mathsf{P}_\circ\, \mathit{send}(\{\!|N_2|\!\}^a_{K_B}, B)]$$

then, for every $C, D, E \in Princ$ with $E \notin \{A, B\}$, and every message M which is not $\{N_2, K_A^{-1}\}$-secure, or M contains N_2 outside a submessage $\{\!|N_1; N_2; B|\!\}^a_{K_A}$, the following holds:

$$\mu, \xi' \nVdash @_{Ch}[\mathsf{P}_\circ\, \mathit{in}(C, M, D)] \text{ and } \mu, \xi' \nVdash @_E[\mathit{knows}(M)].$$

Proof: We prove this with the help of Lemma 4.1. Let $G = \{A, B\}$ and $S = \{N_2, K_A^{-1}\} \cup \{M \mid M$ contains N_2 outside a submessage $\{\!|N_1; N_2; B|\!\}^a_{K_A}\}$. It is not difficult to check that S is a rational set. Indeed, $\mathit{parts}(N_2) = \mathit{parts}(K_A^{-1}) = \emptyset$ and, for each of the remaining messages M, either $N_2 \in \mathit{parts}(M)$ and $\mathit{parts}(M) \cap S \neq \emptyset$ or $M' \in \mathit{parts}(M)$ for some message M' that contains N_2 outside a submessage $\{\!|N_1; N_2; B|\!\}^a_{K_A}$. But, in this case, $M' \in S$ and so $\mathit{parts}(M) \cap S \neq \emptyset$.

Next, we establish condition (i) of Lemma 4.1. The only relevant atoms in S are N_2 and K_A^{-1}. As $\mu, \xi \Vdash @_B[\mathit{fresh}(N_2)]$, it follows that $\mu \Vdash @_B[* \Rightarrow \mathsf{F}\, \mathit{fresh}(N_2)]$. For K_A^{-1}, it follows from axiom **(aKey1)** that $\mu \Vdash @_A[* \Rightarrow \mathit{knows}(K_A^{-1})]$.

Now, let $E \notin G$ and suppose that $\mu, \xi \Vdash @_E[\mathit{knows}(M)]$. From $\mu, \xi \Vdash @_B[\mathit{fresh}(N_2)]$ and the freshness axioms **(F1-2)** it follows not only that M is $\{N_2\}$-secure, but also that M does not even contain N_2. Furthermore, axiom **(aKey2)** and the honesty of A guarantee that all the messages ever in the channel, or known by some principal other than A, are $\{K_A^{-1}\}$-secure. Thus, the message M is S-secure. This implies that condition (ii) of Lemma 4.1 holds, which tells us that for every global state $\xi' \supseteq \xi$ of μ either condition (iii) or condition (iv) of Lemma 4.1 must hold.

We proceed to show that condition (iv) cannot hold. Concretely, we prove by global induction on $\xi' \supseteq \xi$ that if $\mu, \xi' \nVdash @_A[\mathsf{P}_\circ\, \mathit{send}(\{\!|N_2|\!\}^a_{K_B}, B)]$ then condition (iii) of Lemma 4.1 holds for ξ' and condition (iv) of Lemma 4.1 does not. The base case with $\xi' = \xi$ is simple: the same argument that we used to show that condition (ii) of Lemma 4.1 holds, together with axiom **(P1)**, establishes that condition (iv) cannot hold at ξ. For the induction step, let us assume that $\xi' \cup \{e\} \supset \xi' \supseteq \xi$ are global states and that, by the induction hypothesis, if $\mu, \xi' \nVdash @_A[\mathsf{P}_\circ\, \mathit{send}(\{\!|N_2|\!\}^a_{K_B}, B)]$ then (iii) holds for ξ' and (iv) does not. Since, by Lemma 4.1, we know that (iii) or (iv) must hold, it suffices to show that if $\mu, \xi' \cup \{e\} \nVdash @_A[\mathsf{P}_\circ\, \mathit{send}(\{\!|N_2|\!\}^a_{K_B}, B)]$ then (iv) cannot hold for $\xi' \cup \{e\}$. We must then consider, in turn, the two relevant cases for condition (iv) to hold, that is, $e \in Ev_A$ and $e \in Ev_B$.

Case (1): If $e \in Ev_A$, we analyze the possible sending actions $\alpha_A(e)$.

(1.1) If $\alpha_A(e) = \mathit{send}(\{\!|N_1^*; A|\!\}^a_{K_X}, X)$, it must be in a prefix

$$\langle \mathit{fresh}(N_1^*).\mathit{send}(\{\!|N_1^*; A|\!\}^a_{K_X}, X)\rangle$$

of an initiator run of the protocol with some $X \in Princ$. As A freshly generated N_1^*, and B freshly generated N_2, the freshness axioms **(F1-2)** guarantee that $N_1^* \neq N_2$. Therefore, the message $M = \{\!|N_1^*; A|\!\}^a_{K_X} \in S\text{-}Sec$.

(1.2) If $\alpha_A(e) = \mathit{send}(\{\!|N_1^*; N_2^*; A|\!\}^a_{K_X}, X)$, it must be in a prefix

$$\langle \mathit{rec}(\{\!|N_1^*; X|\!\}^a_{K_A}).\mathit{fresh}(N_2^*).\mathit{send}(\{\!|N_1^*; N_2^*; A|\!\}^a_{K_X}, X)\rangle$$

of a responder run of the protocol with some $X \in \mathit{Princ}$. As A freshly generated N_2^*, and B freshly generated N_2, the freshness axioms **(F1-2)** guarantee that $N_2^* \neq N_2$. Moreover, A first received $\{\!|N_1^*; X|\!\}^a_{K_A}$ from the channel, and axioms **(P3)**, **(C2)**, **(C1)**, **(P1)** together with the induction hypothesis guarantee that $N_1^* \neq N_2$. Therefore, $M = \{\!|N_1^*; N_2^*; A|\!\}^a_{K_X} \in S\text{-}\mathit{Sec}$ as in case (1.1) above.

(1.3) If $\alpha_A(e) = \mathit{send}(\{\!|N_2^*|\!\}^a_{K_X}, X)$, it must be in a complete initiator run

$$\langle \mathit{fresh}(N_1^*).\mathit{send}(\{\!|N_1^*; A|\!\}^a_{K_X}, X).\mathit{rec}(\{\!|N_1^*; N_2^*; X|\!\}^a_{K_A}).\mathit{send}(\{\!|N_2^*|\!\}^a_{K_X}, X)\rangle$$

of the protocol with some $X \in \mathit{Princ}$. If, by absurdity, $N_2^* = N_2$, then as A would first receive $\{\!|N_1^*; N_2; X|\!\}^a_{K_A}$ from the channel, it would follow from axioms **(P3)**, **(C2)**, **(C1)**, **(P1)** and the induction hypothesis that $N_1^* = N_1$ and $X = B$. But this would contradict $\mu, \xi' \cup \{e\} \not\Vdash @_A[\mathsf{P}_\circ \mathit{send}(\{\!|N_2|\!\}^a_{K_B}, B)]$. Hence, $N_2^* \neq N_2$ and $M = \{\!|N_2^*|\!\}^a_{K_X} \in S\text{-}\mathit{Sec}$.

Case (2): If $e \in Ev_B$, we also analyze the possible sending actions $\alpha_B(e)$.

(2.1) If $\alpha_B(e) = \mathit{send}(\{\!|N_1^*; B|\!\}^a_{K_X}, X)$, it must be in a prefix

$$\langle \mathit{fresh}(N_1^*).\mathit{send}(\{\!|N_1^*; B|\!\}^a_{K_X}, X)\rangle$$

of an initiator run of the protocol with some $X \in \mathit{Princ}$. As B freshly generated both N_1^* and N_2, but N_2 in a responder run, B's honesty guarantees that $N_1^* \neq N_2$. Hence, $M = \{\!|N_1^*; B|\!\}^a_{K_X} \in S\text{-}\mathit{Sec}$.

(2.2) If $\alpha_B(e) = \mathit{send}(\{\!|N_1^*; N_2^*; B|\!\}^a_{K_X}, X)$, it must be in a prefix

$$\langle \mathit{rec}(\{\!|N_1^*; X|\!\}^a_{K_B}).\mathit{fresh}(N_2^*).\mathit{send}(\{\!|N_1^*; N_2^*; B|\!\}^a_{K_X}, X)\rangle$$

of a responder run of the protocol with some $X \in \mathit{Princ}$. If $N_2^* = N_2$, since $\mu, \xi \Vdash @_B[\mathit{fresh}(N_2) \wedge \mathsf{F}\,\mathit{send}(\{\!|N_1; N_2; B|\!\}^a_{K_A}, A)]$, it follows from B's honesty that also $N_1^* = N_1$ and $X = A$. Clearly, $M = \{\!|N_1; N_2; B|\!\}^a_{K_A} \in S\text{-}\mathit{Sec}$. In contrast, if $N_2^* \neq N_2$ then, as B first received $\{\!|N_1^*; X|\!\}^a_{K_B}$, the axioms **(P3)**, **(C2)**, **(C1)**, **(P1)** together with the induction hypothesis guarantee that $N_1^* \neq N_2$. Thus, again, $M = \{\!|N_1^*; N_2^*; A|\!\}^a_{K_X} \in S\text{-}\mathit{Sec}$.

(2.3) If $\alpha_B(e) = \mathit{send}(\{\!|N_2^*|\!\}^a_{K_X}, X)$, it must be in a complete initiator run

$$\langle \mathit{fresh}(N_1^*).\mathit{send}(\{\!|N_1^*; B|\!\}^a_{K_X}, X).\mathit{rec}(\{\!|N_1^*; N_2^*; X|\!\}^a_{K_B}).\mathit{send}(\{\!|N_2^*|\!\}^a_{K_X}, X)\rangle$$

of the protocol with some $X \in \mathit{Princ}$. Note that, $N_2^* = N_2$ is impossible, as B would first receive $\{\!|N_1^*; N_2; X|\!\}^a_{K_B}$ from the channel, and axioms **(P3)**, **(C2)**, **(C1)**, **(P1)** together with the induction hypothesis would yield a contradiction. Therefore, $N_2^* \neq N_2$, and $M = \{\!|N_2^*|\!\}^a_{K_X} \in S\text{-}\mathit{Sec}$.

Thus, (iv) never holds, (iii) always holds, and the statement follows. □

Note that an analog of this lemma fails for the original NSPK protocol. The proof fails at case (1.3) because the message $\{\!|N_1; N_2|\!\}^a_{K_A}$ can be understood by A as belonging to a run executed with an agent different from B, namely the intruder, as we saw in the previous countermodel.

Our second lemma allows us to conclude that if $A = Z$ is the initiator, then it will never be able to trick an honest principal into sending the message $\{\!|N_2|\!\}^a_{K_B}$ to B in another run of the protocol. Namely, the lemma shows that if B is playing the responder in another protocol run initiated by an honest agent, then it will not mix the relevant data of the two runs.

Lemma 4.4 *Let $B \in \mathit{Hon}$, μ be a NSL model, and ξ a global state such that*

$$\mu, \xi \Vdash @_B[\mathit{fresh}(N_2) \wedge \mathsf{F}\,\mathit{send}(\{\!|N_1; N_2; B|\!\}^a_{K_Z}, Z)],$$

and let also $C \in \mathit{Hon}$*, and* ξ' *a global state such that*

$$\mu, \xi' \Vdash @_C[\mathit{fresh}(N_1^*) \wedge \mathsf{F}\, \mathit{send}(\{\!|N_1^*; C|\!\}^a_{K_B}, B)].$$

For every global state $\xi'' \supseteq \xi'$*, every* $D, E, F \in \mathit{Princ}$ *with* $F \notin \{C, B\}$*, and every message* M' *which is not* $\{N_1^*, K_B^{-1}, K_C^{-1}\}$*-secure, or* M' *contains* $\{|N_1^*; N_2; B|\}^a_{K_C}$*, or* $\{|N_1^*; X|\}^a_{K_Y}$ *with* $X \neq C$ *or* $Y \neq B$*, or* $\{|N; N_1^*; X|\}^a_{K_Y}$ *with any* N, X, Y*, the following holds:*

$$\mu, \xi'' \not\Vdash @_{Ch}[\mathit{in}(D, M', E)] \text{ and } \mu, \xi'' \not\Vdash @_F[\mathit{knows}(M')].$$

Proof: We reuse Lemma 4.1, now with $G = \{B, C\}$ and $S = \{N_1^*, K_B^{-1}, K_C^{-1}\} \cup \{M' \mid M'$ contains $\{|N_1^*; N_2; B|\}^a_{K_C}$, or $\{|N_1^*; X|\}^a_{K_Y}$ with $X \neq C$ or $Y \neq B$, or $\{|N; N_1^*; X|\}^a_{K_Y}$ for any $N, X, Y\}$. The proof that S is a rational set and that conditions (i) and (ii) of Lemma 4.1 hold at ξ' is similar to that of Lemma 4.3. Therefore, Lemma 4.1 tells us that for every global state $\xi'' \supseteq \xi'$ of μ either condition (iii) or condition (iv) of Lemma 4.1 must hold.

We proceed to show that condition (iv) cannot hold. Concretely, we prove by global induction on $\xi'' \supseteq \xi'$ that condition (iii) of Lemma 4.1 holds for ξ'' and condition (iv) of Lemma 4.1 does not hold. The base case with $\xi'' = \xi'$ is immediate. For the induction step, let us assume that $\xi'' \cup \{e\} \supset \xi'' \supseteq \xi'$ are global states and that, by induction hypothesis, (iii) holds for ξ'' and (iv) does not. Since, by Lemma 4.1, we know that (iii) or (iv) must hold, it suffices to show that (iv) cannot hold for $\xi'' \cup \{e\}$. We must then consider, in turn, the two relevant cases for condition (iv) to hold, that is, $e \in \mathit{Ev}_B$ and $e \in \mathit{Ev}_C$.

Case (1): If $e \in \mathit{Ev}_B$, we analyze the possible sending actions $\alpha_B(e)$.

(1.1) If $\alpha_B(e) = \mathit{send}(\{|N_1^\circ; B|\}^a_{K_X}, X)$, it must be in a prefix

$$\langle \mathit{fresh}(N_1^\circ).\mathit{send}(\{|N_1^\circ; B|\}^a_{K_X}, X)\rangle$$

of an initiator run of the protocol with some $X \in \mathit{Princ}$. As B freshly generated N_1°, and C freshly generated N_1^*, the freshness axioms **(F1-2)** guarantee that $N_1^\circ \neq N_1^*$. Thus, the message $M' = \{|N_1^\circ; B|\}^a_{K_X}$ is in S-*Sec*.

(1.2) If $\alpha_B(e) = \mathit{send}(\{|N_1^\circ; N_2^\circ; B|\}^a_{K_X}, X)$, it must be in a prefix

$$\langle \mathit{rec}(\{|N_1^\circ; X|\}^a_{K_B}).\mathit{fresh}(N_2^\circ).\mathit{send}(\{|N_1^\circ; N_2^\circ; B|\}^a_{K_X}, X)\rangle$$

of a responder run of the protocol with some $X \in \mathit{Princ}$. As B freshly generated N_2°, and C freshly generated N_1^*, the freshness axioms **(F1-2)** guarantee that $N_2^\circ \neq N_1^*$. Moreover, as B first receives $\{|N_1^\circ; X|\}^a_{K_B}$, axioms **(P3)**, **(C2)**, **(C1)**, **(P1)** together with the induction hypothesis guarantee that either $N_1^\circ \neq N_1^*$, or $N_1^\circ = N_1^*$ and $X = C$. In the former case, it is immediate that $M' = \{|N_1^\circ; N_2^\circ; B|\}^a_{K_X}$ is S-secure. In the latter case, $N_2^\circ = N_2$ is impossible, because $\mu, \xi \Vdash @_B[\mathit{fresh}(N_2) \wedge \mathsf{F}\, \mathit{send}(\{|N_1; N_2; B|\}^a_{K_Z}, Z)]$ and B's honesty would necessitate $N_1^\circ = N_1$ and $X = Z$, which contradicts C's honesty. Therefore, $N_2^\circ \neq N_2$ and the message $M' = \{|N_1^*; N_2^\circ; B|\}^a_{K_C}$ is S-secure.

(1.3) If $\alpha_B(e) = \mathit{send}(\{|N_2^\circ|\}^a_{K_X}, X)$, it must be in a complete initiator run

$$\langle \mathit{fresh}(N_1^\circ).\mathit{send}(\{|N_1^\circ; B|\}^a_{K_X}, X).\mathit{rec}(\{|N_1^\circ; N_2^\circ; X|\}^a_{K_B}).\mathit{send}(\{|N_2^\circ|\}^a_{K_X}, X)\rangle$$

of the protocol with some $X \in \mathit{Princ}$. As B will first receive $\{|N_1^\circ; N_2^\circ; X|\}^a_{K_B}$, axioms **(P3)**, **(C2)**, **(C1)** and **(P1)** along with the induction hypothesis guarantee that $N_2^\circ \neq N_1^*$. Thus, the message $M' = \{|N_2^\circ|\}^a_{K_X}$ is S-secure.

Case (2): If $e \in \mathit{Ev}_C$, we also analyze the relevant sending actions $\alpha_C(e)$.

(2.1) If $\alpha_C(e) = \mathit{send}(\{|N_1^\circ; C|\}^a_{K_X}, X)$, it must be in a prefix

$$\langle \mathit{fresh}(N_1^\circ).\mathit{send}(\{|N_1^\circ; C|\}^a_{K_X}, X)\rangle\,.$$

If $N_1^\circ \neq N_1^*$ then $M' = \{|N_1^\circ; C|\}^a_{K_X}$ is S-secure. In contrast, if $N_1^\circ = N_1^*$, as $\mu, \xi' \Vdash @_C[\mathit{fresh}(N_1^*) \wedge \mathsf{F}\, \mathit{send}(\{|N_1^*; C|\}^a_{K_B}, B)]$ it follows from C's honesty that also $X = B$, and the message $M' = \{|N_1^*; C|\}^a_{K_B}$ is S-secure.

(2.2) If $\alpha_C(e) = send(\{\!|N_1^\circ; N_2^\circ; C|\!\}^a_{K_X}, X)$, it must be in a prefix

$$\langle rec(\{\!|N_1^\circ, X|\!\}^a_{K_C}).fresh(N_2^\circ).send(\{\!|N_1^\circ; N_2^\circ; C|\!\}^a_{K_X}, X)\rangle$$

of a responder run of the protocol with some $X \in Princ$. As C freshly generates N_1^* and N_2°, but N_1^* in an initiator run, C's honesty ensures that $N_2^\circ \neq N_1^*$. Moreover, since C first received $\{\!|N_1^\circ; X|\!\}^a_{K_C}$ from the channel, the axioms **(P3)**, **(C2)**, **(C1)** and **(P1)** along with the induction hypothesis guarantee that $N_1^\circ \neq N_1^*$. Therefore, the message $M' = \{\!|N_1^\circ; N_2^\circ; C|\!\}^a_{K_X}$ is S-secure.

(2.3) If $\alpha_C(e) = send(\{\!|N_2^\circ|\!\}^a_{K_X}, X)$, it must be in a complete initiator run

$$\langle fresh(N_1^\circ).send(\{\!|N_1^\circ; C|\!\}^a_{K_X}, X).rec(\{\!|N_1^\circ; N_2^\circ; X|\!\}^a_{K_C}).send(\{\!|N_2^\circ|\!\}^a_{K_X}, X)\rangle$$

of the protocol with some $X \in Princ$. The proof is similar to step (1.3) above.

Thus, (iv) never holds, (iii) always holds, and the statement follows. □

We now proceed to the main result: the responder authenticates the initiator. Recall from Section 3.3 that given an honest principal B, $auth^{Resp,Init,3}_{B,A}(A, B, N_1, N_2)$ corresponds to

$$@_B[role^{Resp}_B(A, B, N_1, N_2)] \Rightarrow @_A[\mathsf{P}_\circ\, send(\{\!|N_2|\!\}^a_{K_B}, B)],$$

where $role^{Resp}_B(A, B, N_1, N_2)$ is

$$rec(\{\!|N_2|\!\}^a_{K_B}) \wedge \mathsf{P}(send(\{\!|N_1; N_2; B|\!\}^a_{K_A}, A) \wedge \mathsf{P}(fresh(N_2) \wedge \mathsf{P}\, rec(\{\!|N_1; A|\!\}^a_{K_B}))).$$

Proposition 4.5 *NSL* $\Vdash auth^{Resp,Init,3}_{B,A}(A, B, N_1, N_2)$ *for* $A \in Princ$, $B \in Hon$, *and* N_1 *and* N_2 *arbitrary distinct nonces.*

Proof: Let μ be an NSL model, ξ a global state such that $\mu, \xi \Vdash @_B[role^{Resp}_B(A, B, N_1, N_2)]$. Recall that B is honest. We have two cases: either (1) A is also honest or (2) A is actually Z.

Case (1): Assume, by absurdity, that $\mu, \xi \not\Vdash @_A[\mathsf{P}_\circ\, send(\{\!|N_2|\!\}^a_{K_B}, B)]$. Since both A and B are honest, we can consider ξ', where $\xi' \subseteq \xi$ and $\mu, \xi' \Vdash @_B[fresh(N_2)]$, and use Lemma 4.3 to conclude that $\mu, \xi \not\Vdash @_{Ch}[\mathsf{P}_\circ \bigvee_{C \in Princ} in(C, \{\!|N_2|\!\}^a_{K_B}, B)]$. However, given axioms **(P3)** and **(C2)**, this contradicts the assumption that $\mu, \xi \Vdash @_B[role^{Resp}_B(A, B, N_1, N_2)]$, as in particular it must be the case that $\mu, \xi \not\Vdash @_B[rec(\{\!|N_2|\!\}^a_{K_B})]$. Hence $\mu, \xi \Vdash @_A[\mathsf{P}_\circ\, send(\{\!|N_2|\!\}^a_{K_B}, B)]$.

Case (2): If $A = Z$ then B's run is actually

$$\langle rec(\{\!|N_1; Z|\!\}^a_{K_B}).fresh(N_2).send(\{\!|N_1; N_2; B|\!\}^a_{K_Z}, Z).rec(\{\!|N_2|\!\}^a_{K_B})\rangle.$$

We must show that $\mu, \xi \Vdash @_Z[\mathsf{P}_\circ\, send(\{\!|N_2|\!\}^a_{K_B}, B)]$. From axioms **(P3)** and **(C1-2)**, we know that $\mu, \xi \Vdash \bigvee_{C \in Princ} @_C[\mathsf{P}_\circ\, send(\{\!|N_2|\!\}^a_{K_B}, B)]$. Hence, it suffices to prove that Z cannot trick an honest principal C into sending the message $\{\!|N_2|\!\}^a_{K_B}$ to B. We can exclude the case when $C = B$, since in protocol models no honest principal sends messages to himself. Now, an honest $C \neq B$ will only send such a message in an initiator role of the form

$$\langle fresh(N_1^*).send(\{\!|N_1^*; C|\!\}^a_{K_B}, B).rec(\{\!|N_1^*; N_2; B|\!\}^a_{K_C}).send(\{\!|N_2|\!\}^a_{K_B}, B)\rangle\,.$$

But clearly only B himself, or Z, would send $\{\!|N_1^*; N_2; B|\!\}^a_{K_C}$. Moreover, B can be excluded because his honesty guarantees that he will not use N_2 but rather a fresh value, according to the definition of protocol models or else it must be the case that $N_1^* = N_1$ and $C = Z$, which contradicts the honesty of C. Moreover, if for any $\xi' \subseteq \xi$ we have indeed that $\mu, \xi' \Vdash @_C[fresh(N_1^*) \wedge \mathsf{F}\, send(\{\!|N_1^*; C|\!\}^a_{K_B}, B)]$, then Lemma 4.4 guarantees that $\mu, \xi \not\Vdash @_Z[knows(\{\!|N_1^*; N_2; B|\!\}^a_{K_C})]$, and therefore C would never receive $\{\!|N_1^*; N_2; B|\!\}^a_{K_C}$ and thus could not complete the run. We can therefore conclude that $\mu, \xi \Vdash @_Z[\mathsf{P}_\circ\, send(\{\!|N_2|\!\}^a_{K_B}, B)]$. □

As NSPK and NSL are well studied, our results are not surprising and the proof ideas are similar to those found, e.g., in [55]. However, our development illustrates well how we can use our channel model to give semantic proofs and how we can exploit metatheoretic properties, all within DTL. Other security protocols can be falsified or verified similarly, using DTL as an object logic. We now turn to the main strength of DTL: its use as a metalogic to relate different models for security protocols and to prove further metatheorems about the models themselves.

5 Meta-level model analysis

In this section, we present two concrete examples of metareasoning. First, we prove the equivalence of two models for guaranteeing message-origin authentication. Second, we relate channel-based and intruder-centric models, showing that it is sufficient to consider models with a single intruder who controls the network.

5.1 Message-origin authentication

The following example is of a different nature than the last one. We now show how DTL can be used to relate models at different levels of abstraction. Establishing such formal relationships is a central paradigm in Information Security and is used, for example, when carrying out simulation proofs to show that concrete cryptographic operations implement a given ideal functionality. Our example is centered around *message-origin authentication*: ensuring that a message purported to come from an agent really originated with the agent. We will use DTL to study the relationship between two models designed to guarantee message-origin authentication.

1. An *abstract model* TTP where principals may use a special "logged" channel T controlled by a trusted third party. This channel logs all incoming messages and issues evidence of their origin to the recipients.

2. A *concrete model* DS that is closer to a possible realization of an authentic channel. Communication takes place in DS over a public channel, but principals digitally sign the messages they send so that their signatures can be verified by the recipients.

We investigate several relationships between these two models by exploring transformations of their corresponding DTL models, along with translations of their properties. By abstracting away details of the communication media, we prove that the two models are equivalent under mild assumptions about the nature of message-origin authentication.

5.1.1 *TTP*: trusted third party logging

In this model, we extend the channel-based model CB of Section 3 with an additional communication medium T, representing the logged channel and controlled by a trusted third party. Principals can choose to send or receive messages either through the public channel or T. Messages exchanged through T are logged by the trusted third party, who issues evidence of their origin to the recipients. Hence, all principals are augmented with actions for communicating using T and with state propositions that provide evidence of origin for the messages received from T.

Recall that we consider fixed a network signature $\langle Princ, Num\rangle$. The signature $\Sigma_{TTP} = \langle Princ \uplus \{Ch, T\}, Act, Prop\rangle$ is such that for each $A \in Princ$ we have that

- Act_A is composed of
  - $send(M, B)$, $rec(M)$, $spy(M)$, and $fresh(X)$, as in Σ_{CB},
  - $send_T(M, B)$: sending message M to B via T,
  - $rec_T(B, M)$: receiving from T message M originating from B,
  - $spy_T(B, M)$: eavesdropping in T message M originating from B;

- $Prop_A$ includes the state propositions
  - $knows(M)$, as in Σ_{CB},
  - $evid(B, M)$: evidence was obtained from T that message M originates from B;

and for Ch, as in Σ_{CB}, and also for T, we have

- $Act_{Ch} = Act_T$, where both consist of the actions $in(A, M, B)$, $out(A, M, B)$, and $leak$;
- $Prop_{Ch} = Prop_T = \emptyset$.

We use $\mathcal{L}_{TTP}$ to denote the DTL language over the signature Σ_{TTP}.

The axiomatization of TTP includes the axioms **(F1–F2)**, **(C1–C4)**, and **(P1–P6)** of CB, from Section 3. It also includes corresponding versions of the channel axioms for T, namely

(T1) $@_T[in(A, M, B) \gg_A send_T(M, B)]$

(T2) $@_T[out(A, M, B) \Rightarrow \mathsf{P}\ in(A, M, B)]$

(T3) $@_T[out(A, M, B) \gg_B rec_T(A, M)]$

(T4) $@_T[leak \Rightarrow (\bigvee_{B \in Princ} ©_B[\top])]$

and axioms for the interaction of each principal $A \in Princ$ with T, that is

($\mathbf{P_T1}$) $@_A[send_T(M, B) \Rightarrow \mathsf{Y}(knows(M) \wedge knows(B))]$

($\mathbf{P_T2}$) $@_A[send_T(M, B) \gg_T in(A, M, B)]$

($\mathbf{P_T3}$) $@_A[rec_T(B, M) \gg_T out(B, M, A)]$

($\mathbf{P_T4}$) $@_A[spy_T(B, M) \gg_T (leak \wedge \mathsf{P}(\bigvee_{C \in Princ} in(B, M, C)))]$

($\mathbf{P_T6}$) $@_A[fresh(X) \Rightarrow \neg\, ©_T[\top]]$.

An additional axiom is needed to define the state propositions representing evidence for each principal $A \in Princ$, namely

(E) $@_A[evid(B, M) \Leftrightarrow \mathsf{P}_\circ\, rec_T(B, M)]$.

TTP models μ are those interpretation structures over Σ_{TTP} that satisfy the above axioms and the following clause ($\mathbf{K_T}$), which replaces ($\mathbf{K}$) of the CB model. For each $A \in Princ$, $M \in Msg$, and non-empty local state ξ_A,

($\mathbf{K_T}$) $\mu, \xi_A \Vdash_A knows(M)$ iff $M \in close(\{M' \mid \mu, \xi_A \Vdash_A (\mathsf{Y}\, knows(M')) \vee rec(M') \vee spy(M') \vee fresh(M') \vee (\bigvee_{B \in Princ} rec_T(B, M') \vee spy_T(B, M'))\})$.

Clearly, (K1–K7) also follow from $\mathbf{K_T}$, as well as the following properties:

$$@_A[rec_T(B, M) \Rightarrow knows(M)] \qquad (\mathrm{K}_T5)$$

$$@_A[spy_T(B, M) \Rightarrow knows(M)] \qquad (\mathrm{K}_T6)$$

We also strengthen the honesty requirement. Besides axiom **(Hon)**, we require for every $A \in Hon$:

($\mathbf{Hon_T}$) $@_A[\neg\, spy_T(B, M)]$, for every $B \in Princ$ and message M.

In TTP, we can prove that the state propositions that provide each principal evidence of origin of the messages he received via T are actually correct.

Proposition 5.1 *$TTP \Vdash @_A[evid(B, M)] \Rightarrow @_B[\mathsf{P}_\circ\, send_T(M, A)]$ for $A, B \in Princ$ and $M \in Msg$.*

Proof: Follows easily from axioms **(E)**, **($\mathbf{P_T3}$)**, **(T2)**, and **(T1)**. □

5.1.2 Digital signatures

The *DS* model specializes the channel-based model *CB* of Section 3. We require that every principal A possesses a secret, special-purpose asymmetric key K_A^{-1}, subject to the key axioms **(aKey1–aKey2)**. We assume that these keys are new, i.e., $\{K_A \mid A \in \mathit{Princ}\} \cap \mathit{Num} = \emptyset$, and that the network signature $\langle \mathit{Princ}, \mathit{Num}\rangle$ is augmented to $\langle \mathit{Princ}, \mathit{Num}^+\rangle$, with $\mathit{Num}^+ = \mathit{Nonces} \uplus \mathit{SymK} \uplus \mathit{PubK}^+$ where $\mathit{PubK}^+ = \mathit{PubK} \uplus \{K_A \mid A \in \mathit{Princ}\}$. We write Msg^+ to denote the set of messages in the augmented signature, in contrast to *Msg*, the messages in the original signature. For the purpose of message-origin authentication, we specify that a principal A should indicate the origin of a message M by sending it along with A's name and a signature, that is, A sends $M; A; \{\!|M|\!\}^a_{K_A^{-1}}$. By using the associated public key K_A, the message's receiver can then verify the signature to determine whether M originates from A.

We define Σ_{DS} to be identical to Σ_{CB}, except defined over the augmented network signature. We let $\mathcal{L}_{DS}$ denote the DTL language over the signature Σ_{DS}. To guarantee the desired behavior, we require that each honest principal A only uses his private key for signing messages where signatures or their associated public/private keys do not occur. Similarly, we require that A never receives messages that use the special-purpose public/private keys, unless they are properly signed. Namely, we require

(NS) $@_A[\neg\, \mathit{send}(M', B)]$ if $M' \in \mathit{Msg}^+ \setminus \mathit{Msg}$ and $M' \neq M; A; \{\!|M|\!\}^a_{K_A^{-1}}$ for some $M \in \mathit{Msg}$,

(NR) $@_A[\neg\, \mathit{rec}(M')]$ if $M' \in \mathit{Msg}^+ \setminus \mathit{Msg}$ and $M' \neq M; B; \{\!|M|\!\}^a_{K_B^{-1}}$ for some $M \in \mathit{Msg}$ and $B \in \mathit{Princ}$.

Note that $A \in \mathit{Hon}$ will never disclose his special-purpose private key K_A^{-1} or forward messages signed by other principals using the special-purpose keys. As above, we assume for simplicity that principals know each other's names **(N)** and public keys **(PK)**, and that honest principals do not spy **(Hon)**. The *DS* models are the interpretation structures that satisfy these axioms along with the *CB* requirements.

5.1.3 Comparing the models

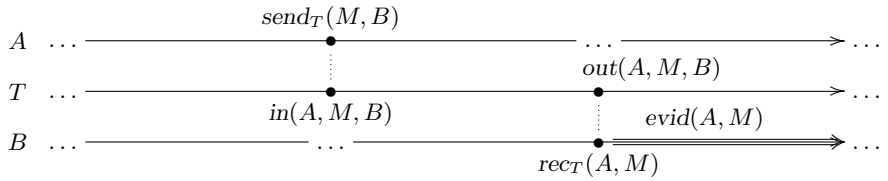
The *TTP* model is more abstract than the *DS* model in that it provides a higher-level message-origin authentication mechanism without using cryptography. Still, we would like that the two models provide message-origin authentication mechanisms with comparable behavior, as depicted in Figure 6, where the triple arrow indicates that $\mathit{evid}(A, M)$ holds from that point on.

It is clear that we must use the more abstract language of the *TTP* and provide a translation to the language of the *DS* model. Given a formula $\gamma \in \mathcal{L}_{TTP}$, let $\overline{\gamma} \in \mathcal{L}_{DS}$ be the formula obtained from γ by uniformly replacing each occurrence of

- T with Ch,
- $\mathit{send}_T(M, B)$ local to principal A with $\mathit{send}(M; A; \{\!|M|\!\}^a_{K_A^{-1}}, B)$,
- $\mathit{rec}_T(B, M)$ with $\mathit{rec}(M; B; \{\!|M|\!\}^a_{K_B^{-1}})$,
- $\mathit{spy}_T(B, M)$ with $\mathit{spy}(M; B; \{\!|M|\!\}^a_{K_B^{-1}})$,
- $\mathit{evid}(B, M)$ with $\mathsf{P}_\circ\, \mathit{rec}(M; B; \{\!|M|\!\}^a_{K_B^{-1}})$,
- $\mathit{in}(A, M, B)$ local to T with $\mathit{in}(A, M; A; \{\!|M|\!\}^a_{K_A^{-1}}, B)$,
- $\mathit{out}(A, M, B)$ local to T with $\mathit{out}(A, M; A; \{\!|M|\!\}^a_{K_A^{-1}}, B)$.

We would like to prove that *TTP* and *DS* are equivalent in the sense that $\mathit{TTP} \Vdash \gamma$ iff $\mathit{DS} \Vdash \overline{\gamma}$. One way to establish such an equivalence is to define model transformations $\beta : \mathit{TTP}_\Xi \to \mathit{DS}_\Xi$ and $\theta : \mathit{DS}_\Xi \to \mathit{TTP}_\Xi$ such that, for every $\gamma \in \mathcal{L}_{TTP}$, $\langle \mu_{TTP}, \xi_{TTP}\rangle \in \mathit{TTP}_\Xi$, and $\langle \mu_{DS}, \xi_{DS}\rangle \in \mathit{DS}_\Xi$:

TTP model:



DS model:

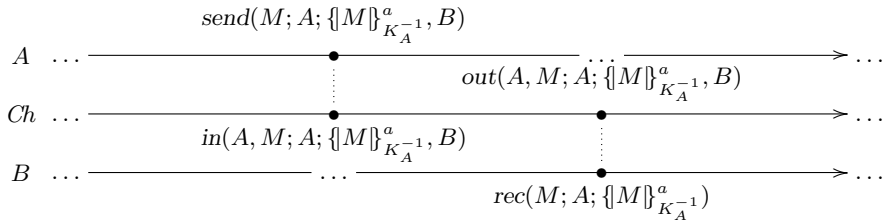


Figure 6: Comparing a *TTP* model and a *DS* model.

(i) $\mu_{TTP}, \xi_{TTP} \Vdash \gamma$ if and only if $\beta(\mu_{TTP}, \xi_{TTP}) \Vdash \overline{\gamma}$, and

(ii) $\theta(\mu_{DS}, \xi_{DS}) \Vdash \gamma$ if and only if $\mu_{DS}, \xi_{DS} \Vdash \overline{\gamma}$.

Observe that the composition of the two model transformations β and θ need not be the identity, but only preserve logical equivalence for $\mathcal{L}_{TTP}$ modulo translation of formulas to the language of the *DS* model.

There are two subtle issues here. First, the model transformations, associated to β and guided by the syntactic translation defined above, must merge *Ch* and T together, since there is only one communication medium in *DS*.[4] Hence, one cannot expect property (i) to hold in general when γ involves temporal formulas local to either the public or the logged channel. This is, however, a minor restriction, as we should still be able to prove (i) for all relevant properties concerning the behavior of the principals.

The second issue concerns property (ii): the transformation θ must be able to represent in *TTP* all behaviors allowed in *DS* models. The problem is that there is a minor incompatibility between the models related to the use of the special-purpose keys in *DS*. The axioms **(NS)** and **(NR)** require that honest agents use signatures appropriately in their realization of a trusted channel. However, a priori, it does not seem reasonable to similarly restrict the intruder. Still, it is clear that Z gains nothing from sending messages that no other principal is willing to receive. Hence, as a consequence of axiom **(EZ1)**, we have the following property:

($\mathbf{NS_Z}$) $@_Z[\neg\, send(M', A)]$ if $M' \in Msg^+ \setminus Msg$ and $M' \neq M; B; \{\!|M|\!\}^a_{K_B^{-1}}$ for some $M \in Msg$ and $B \in Princ$.

($\mathbf{NS_Z}$) is weaker than the axiom **(NS)** for the honest principals. As a consequence, Z can still prevent *message-origin authentication* in the *DS* model, which we established for the *TTP* model in Proposition 5.1. The reason is that in the *TTP* model, T not only issues evidence of message origin but it also provides a stronger form of correspondence, in that the intended recipient indeed receives the signed message. However, this cannot be guaranteed in the *DS* model since nothing prevents Z from spying a message $M; A; \{\!|M|\!\}^a_{K_A^{-1}}$ sent by a principal A to a principal B, and then forwarding it to some other principal C. Clearly, when receiving it, C will have evidence of the message's origin but cannot be sure that A really sent the message to him. This situation, which

[4]We will refrain here from working out the details of this model transformation since we will do this in detail shortly in a slightly different, but compatible, context.

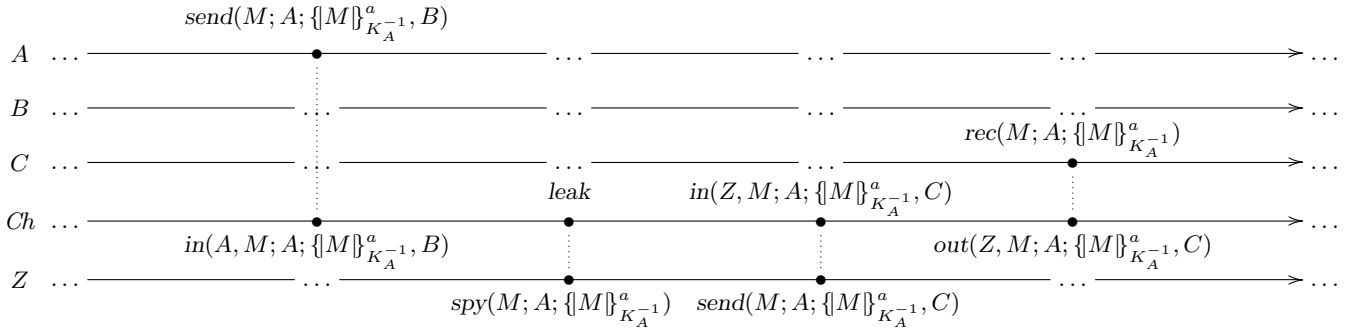


Figure 7: Z interferes with *message-origin authentication* in the DS model

is illustrated in Figure 7, cannot be mimicked in the TTP model. Thus, there are two options for establishing the desired equivalence:

1. Relax the TTP model to allow the intruder to divert messages from T and forward them to different destinations.
2. Change the way messages are signed in the DS model to guarantee the required form of correspondence.

Although the second option is more appropriate (as discussed in [51]), both options have advantages and disadvantages, which we discuss below.

Option 1. We can solve the above mismatch by changing the TTP model to allow the situation described in Figure 7. Namely, we consider a slightly different model TTP' where the intruder can divert each message sent to the logged channel T to a different recipient. To formalize this, we extend the TTP signature with an additional action dvt_T for each principal and we change or add new axioms to specify its properties. The signature $\Sigma_{TTP'} = \langle Princ \uplus \{Ch, T\}, Act, Prop\rangle$ is such that for each $A \in Princ$, we have that

- Act_A is composed of
  - $send(M, B)$, $rec(M)$, $spy(M)$, $fresh(X)$, $send_T(M, B)$, $rec_T(B, M)$, and $spy_T(B, M)$, as in Σ_{TTP}, and
  - $dvt_T(B, M, C)$: divert to principal C the message M originating from B;
- $Prop_A$ includes, as in Σ_{TTP}, the state propositions $knows(M)$ and $evid(B, M)$;

and for Ch and T, as in Σ_{TTP}, we have that

- $Act_{Ch} = Act_T$ includes the actions $in(A, M, B)$, $out(A, M, B)$, and $leak$;
- $Prop_{Ch} = Prop_T = \emptyset$;

We use $\mathcal{L}_{TTP'}$ to denote the DTL language over the extended signature $\Sigma_{TTP'}$. We specify the new divert actions by replacing axiom **(T1)** with

(T1') $@_T[in(B, M, C) \Rightarrow (©_B[send_T(M, C)] \vee (\bigvee_{A \in Princ} ©_A[dvt_T(B, M, C)]))]$

and adding additional axioms for each principal A, namely:

$(\mathbf{P_T 7})$ $@_A[dvt_T(B, M, C) \Rightarrow \mathsf{P}\, spy_T(B, M)]$

$(\mathbf{P_T 8})$ $@_A[dvt_T(B, M, C) \gg_T in(B, M, C)]$.

Note that as a consequence of axioms **(Hon$_\mathbf{T}$)** and **(P$_\mathbf{T}$7)**, honest principals are also not allowed to divert messages. The TTP' models are then the interpretation structures over $\Sigma_{TTP'}$ satisfying (**F1** − **F2**), (**C1** − **C4**), (**P1** − **P6**), **(T1')**, (**T2** − **T4**), (**P$_\mathbf{T}$1** − **P$_\mathbf{T}$4**), (**P$_\mathbf{T}$6** − **P$_\mathbf{T}$8**), **(E)**, **(N)**, and (**K$_\mathbf{T}$**). In this setting, we can no longer interpret message-origin authentication as we did in TTP and Proposition 5.1 is weakened as follows.

Proposition 5.2 *$TTP' \Vdash @_A[evid(B, M)] \Rightarrow @_B[\mathsf{P}_\circ(\bigvee_{C\in Princ} send_T(M, C))]$ for $A, B \in Princ$ and $M \in Msg$.*

Proof: This follows from the definition of satisfaction, using the axioms **(E)**, (**P$_\mathbf{T}$3**), **(T2)**, **(T1')**, (**P$_\mathbf{T}$7**), and (**P$_\mathbf{T}$4**) and the fact that interpretation structures are bounded to the past. □

The models TTP' and DS can now be shown to be equivalent, in the precise sense described above, where we extend the translation of formulas $\gamma \in \mathcal{L}_{TTP'}$ to $\overline{\gamma} \in \mathcal{L}_{DS}$ by additionally replacing each occurrence of

- $dvt_T(B, M, C)$ with $send(M; B; \{\!|M|\!\}^a_{K_B^{-1}}, C)$.

Proposition 5.3 *For every $\gamma \in \mathcal{L}^{@}_{TTP'}$ built from formulas private to principals, $TTP' \Vdash \gamma$ if and only if $DS \Vdash \overline{\gamma}$.*

We omit the proof as it is technically very similar to the upcoming proof of the corresponding property that we give for the second option. As a corollary, we have that the model DS based on digital signatures fulfills a form of message-origin authentication similar to the one stated in Proposition 5.2 for TTP', where evidence that a message M originated in B is provided to a principal A by the past reception of $M; B; \{\!|M|\!\}^a_{K_B^{-1}}$.

Corollary 5.4 *$DS \Vdash @_A[\mathsf{P}_\circ\, rec(M; B; \{\!|M|\!\}^a_{K_B^{-1}})] \Rightarrow @_B[\mathsf{P}_\circ(\bigvee_{C\in Princ} send(M; B; \{\!|M|\!\}^a_{K_B^{-1}}, C))]$ for $A, B \in Princ$ and $M \in Msg$.*

Option 2. In this option, we keep the initial formulation of the TTP model, as well as the original meaning of message-origin authentication, but add more structure to the digitally-signed messages in the concrete model. We let $\Sigma_{DS^\sharp} = \Sigma_{DS}$, but we will still use $\mathcal{L}_{DS^\sharp}$ to denote the corresponding DTL language. We change $DS^\sharp$ by requiring that a principal A must now send a message M whose message-origin authentication is required by the recipient B by including the name B as part of the signed message, that is, A sends $M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}$ (cf. [51]). In addition, axioms **(NS)** and **(NR)** must be rewritten, for every honest A, to:

(NS$^\sharp$) $@_A[\neg\, send(M', B)]$, if $M' \in Msg^+ \setminus Msg$ and $M' \neq M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}$ for some $M \in Msg$ and $B \in Princ$

(NR$^\sharp$) $@_A[\neg\, rec(M')]$, if $M' \in Msg^+ \setminus Msg$ and $M' \neq M; B; \{\!|A; M|\!\}^a_{K_B^{-1}}$ for some $M \in Msg$ and $B \in Princ$

Using axiom **(EZ1)**, we also obtain a variant of the **(NS$_\mathbf{Z}$)** property, namely

(NS$^\sharp_\mathbf{Z}$) $@_Z[\neg\, send(M', A)]$, if $M' \in Msg^+ \setminus Msg$ and $M' \neq M; B; \{\!|A; M|\!\}^a_{K_B^{-1}}$ for some $M \in Msg$ and $B \in Princ$.

Finally, we redefine the translation between the languages of the two models. Given $\gamma \in \mathcal{L}_{TTP}$, let $\overline{\gamma}^\sharp \in \mathcal{L}_{DS^\sharp}$ be the formula obtained from γ by uniformly replacing each occurrence of

- T with Ch,
- $send_T(M, B)$ local to principal A with $send(M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}, B)$,

- $rec_T(B, M)$ local to principal A with $rec(M; B; \{\!|A; M|\!\}^a_{K_B^{-1}})$,
- $spy_T(B, M)$ with $\bigvee_{C \in Princ} spy(M; B; \{\!|C; M|\!\}^a_{K_B^{-1}})$,
- $in(A, M, B)$ local to T with $in(A, M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}, B)$,
- $out(A, M, B)$ local to T with $out(A, M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}, B)$.

The models TTP and $DS^\sharp$ can now be shown to be equivalent, as illustrated in Figure 8.

Proposition 5.5 *For every $\gamma \in \mathcal{L}^{@}_{TTP}$ built from formulas private to principals, $TTP \Vdash \gamma$ if and only if $DS^\sharp \Vdash \overline{\gamma}^\sharp$.*

Proof: Let $\beta : TTP_\Xi \to DS^\sharp_\Xi$ be defined, for each pair $\langle \mu_{TTP}, \xi \rangle \in TTP_\Xi$ with $\mu_{TTP} = \langle \lambda, \alpha, \pi \rangle$, by $\beta(\mu_{TTP}, \xi) = \langle \langle \lambda^\sharp, \alpha^\sharp, \pi^\sharp \rangle, \xi \rangle$ where

- $\lambda^\sharp_A = \lambda_A$ for every $A \in Princ$ and $\lambda^\sharp_{Ch} = \langle Ev^\sharp_{Ch}, \leq^\sharp_{Ch} \rangle$, with $Ev^\sharp_{Ch} = Ev_{Ch} \cup Ev_T$ and $\leq^\sharp_{Ch}$ the restriction to $Ev^\sharp_{Ch}$ of any linearization of $\leq$ compatible with the global state ξ;
- $$\alpha^\sharp_A(e) = \begin{cases} \alpha_A(e) & \text{if } \alpha_A(e) \text{ is an } A\text{-action in } \Sigma_{DS^\sharp} \\ send(M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}, B) & \text{if } \alpha_A(e) = send_T(M, B) \\ rec(M; B; \{\!|A; M|\!\}^a_{K_B^{-1}}) & \text{if } \alpha_A(e) = rec_T(B, M) \\ spy(M; B; \{\!|C; M|\!\}^a_{K_B^{-1}}) & \text{if } \alpha_A(e) = spy_T(B, M) \text{ and} \\ & \alpha_T(e') = in(B, M, C), \text{ for some } e' <_T e \end{cases}$$
  $$\alpha^\sharp_{Ch}(e) = \begin{cases} \alpha_{Ch}(e) & \text{if } e \in Ev_{Ch} \\ in(A, M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}, B) & \text{if } e \in Ev_T \text{ and } \alpha_T(e) = in(A, M, B) \\ out(A, M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}, B) & \text{if } e \in Ev_T \text{ and } \alpha_T(e) = out(A, M, B) \\ leak & \text{if } e \in Ev_T \text{ and } \alpha_T(e) = leak \end{cases}$$
- $\pi^\sharp_A(\emptyset) = \{knows(M') \mid M' \in close(\{M \mid knows(M) \in \pi_A(\emptyset)\} \cup \{K_B \mid B \in Princ\} \cup \{K_A^{-1}\})\}$, whereas $\pi^\sharp_{Ch}$ is always empty.

Note that when $\alpha_A(e) = spy_T(B, M)$, we assume fixed a consistent choice of principal C such that $\alpha^\sharp_A(e) = spy(M; B; \{\!|C; M|\!\}^a_{K_B^{-1}})$, as guaranteed by axiom **($\mathbf{P_T}$4)**. In addition, as in all other cases, when seen as a formula, $\alpha^\sharp_A(e)$ corresponds to $\overline{\alpha_A(e)}^\sharp$. Note also that $\alpha^\sharp_{Ch}$ is well-defined since $Ev_{Ch} \cap Ev_T = \emptyset$. Moreover, $knows(M) \in \pi^\sharp_A(\emptyset)$ iff $knows(M) \in \pi_A(\emptyset)$ for every $M \in Msg$. Whenever one defines a model transformation such as this one, one must guarantee that indeed $\langle \lambda^\sharp, \alpha^\sharp, \pi^\sharp \rangle \in DS^\sharp$. It is straightforward, though tedious, to check that all the necessary conditions are satisfied by the interpretation structure. For instance, note that axiom **(NS')** holds by construction as the only sending actions of an honest principal A are either normal channel-sending actions like $send(M, B)$ with $M \in Msg$, or else they come from actions like $send_T(M, B)$ where again $M \in Msg$ and they are transformed into $M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}$. Another axiom, such as **(C1)**, holds in $DS^\sharp$ as a consequence of **(C1)** and **(T1)** holding in any TTP model. Observe, moreover that, by construction, ξ is still a global state of $\langle \lambda^\sharp, \alpha^\sharp, \pi^\sharp \rangle$.

We must prove that $\mu_{TTP}, \xi \Vdash \gamma$ iff $\beta(\mu_{TTP}, \xi) \Vdash \overline{\gamma}^\sharp$, for every $\gamma \in \mathcal{L}^{@}_{TTP}$ built from formulas private to principals. As any such γ must be a Boolean combination of private formulas of different principals, it suffices to show that $\mu_{TTP}, \xi \Vdash @_A[\varphi]$ iff $\beta(\mu_{TTP}, \xi) \Vdash \overline{@_A[\varphi]}^\sharp$, for every $A \in Princ$ and $\varphi \in \mathcal{L}^{@}_A$. The result follows by a trivial adaptation of Lemma 2.2, once we note that the translation of formulas matches the relabeling of the events introduced by β (and θ), and that π and $\pi^\sharp$ agree for messages in Msg.

Conversely, given a $DS^\sharp$ model $\mu_{DS^\sharp} = \langle \lambda^\sharp, \alpha^\sharp, \pi^\sharp \rangle$, we must be able to identify the channel events that correspond, in a TTP model, to the trusted third-party T. Of course, these are precisely the events whose label is somehow related to a message that uses the special-purpose signatures. Below, we will call $e \in Ev^\sharp_{Ch}$ a T-event if one of the following conditions holds:

- $\alpha^\sharp_{Ch}(e) = in(A, M, B)$ or $\alpha^\sharp_{Ch}(e) = out(A, M, B)$, for some $A, B \in Princ$ and some $M \in Msg^+ \setminus Msg$; or
- $\alpha^\sharp_{Ch}(e) = leak$, and $e \in Ev^\sharp_Z$ with $\alpha^\sharp_Z(e) = spy(M)$, for some $M \in Msg^+ \setminus Msg$.

Note that, as a consequence of conditions **(NS$^\sharp$)**, **(NR$^\sharp$)**, and **(NS$^\sharp_{\mathbf{Z}}$)**, all the relevant messages $M \in Msg^+ \setminus Msg$ appearing in the labels of the events above must be of the form $M'; C; \{\!|D; M'|\!\}^a_{K_C^{-1}}$ for some $C, D \in Princ$ and $M' \in Msg$. Furthermore, in the context above, condition **(EZ2)** also ensures that $A = C$ and $B = D$. We now let $\theta : DS^\sharp_\Xi \to TTP_\Xi$ be defined for each pair $\langle \mu_{DS^\sharp}, \xi \rangle \in DS^\sharp_\Xi$ by $\theta(\mu_{DS^\sharp}, \xi) = \langle \langle \lambda, \alpha, \pi \rangle, \xi \rangle$ where

- $\lambda_A = \lambda^\sharp_A$ for every $A \in Princ$, $\lambda_{Ch} = \langle Ev_{Ch}, \leq_{Ch} \rangle$, and $\lambda_T = \langle Ev_T, \leq_T \rangle$, where $Ev_T = \{e \in Ev^\sharp_{Ch} \mid e \text{ is a } T\text{-event}\}$, $Ev_{Ch} = Ev^\sharp_{Ch} \setminus Ev_T$, and $\leq_{Ch}$ and $\leq_T$ are the corresponding restrictions of $\leq^\sharp_{Ch}$;
- $\alpha_A(e) = \begin{cases} \alpha^\sharp_A(e) & \text{if } \alpha^\sharp_A(e) \text{ does not involve a message in } Msg^+ \setminus Msg \\ send_T(B, M) & \text{if } \alpha^\sharp_A(e) = send(M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}, B) \\ rec_T(B, M) & \text{if } \alpha^\sharp_A(e) = rec(M; B; \{\!|A; M|\!\}^a_{K_B^{-1}}) \\ spy_T(B, M) & \text{if } \alpha^\sharp_A(e) = spy(M; B; \{\!|C; M|\!\}^a_{K_B^{-1}}), \end{cases}$

  $\alpha_{Ch}(e) = \alpha^\sharp_{Ch}(e)$ for $e \in Ev_{Ch}$, and

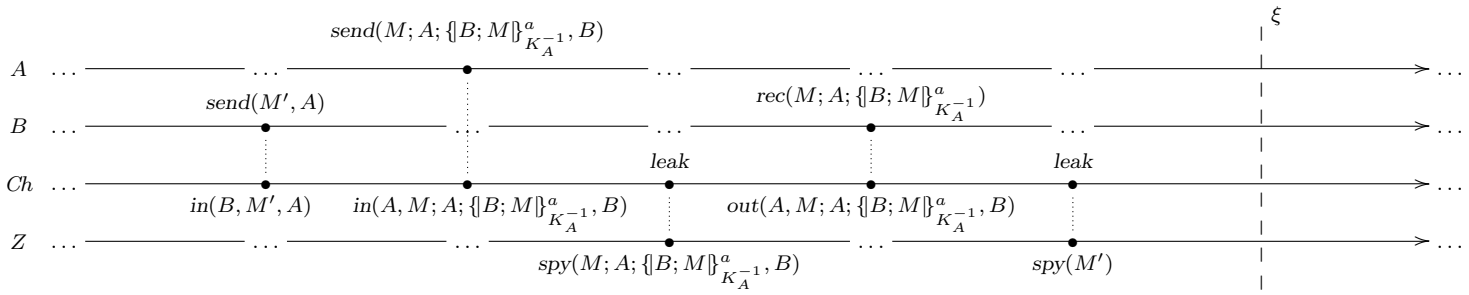
  $\alpha_T(e) = \begin{cases} in(A, M, B) & \text{if } \alpha^\sharp_{Ch}(e) = in(A, M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}, B) \\ out(A, M, B) & \text{if } \alpha^\sharp_A(e) = out(A, M; A; \{\!|B; M|\!\}^a_{K_A^{-1}}, B) \\ leak & \text{if } \alpha^\sharp_A(e) = leak; \end{cases}$
- $\pi_A(\emptyset) = \{knows(M) \mid M \in \pi^\sharp_A(\emptyset) \cap Msg\}$, and π_{Ch} and π_T are both always empty.

It is tedious although not difficult to prove that $\langle \lambda, \alpha, \pi \rangle \in TTP$ and that ξ is still a global state. Moreover, by construction, we have that β and θ are almost inverses of each other, in the sense that $\langle \mu_{DS^\sharp}, \xi \rangle$ and $\beta(\theta(\mu_{DS^\sharp}, \xi))$ may differ only in the choice of a linearization compatible with ξ in the construction of the latter that is different from the linearization implicit in the former. Still, the two models are similar enough to satisfy the preconditions of Lemma 2.2, again, with respect to all formulas $\gamma \in \mathcal{L}^{\not\emptyset}_{TTP}$ built from formulas private to principals. As a consequence, we also obtain that $\mu_{DS^\sharp}, \xi \Vdash \overline{\gamma}^\sharp$ iff $\beta(\theta(\mu_{DS^\sharp}, \xi)) \Vdash \overline{\gamma}^\sharp$ iff $\theta(\mu_{DS^\sharp}, \xi) \Vdash \gamma$.

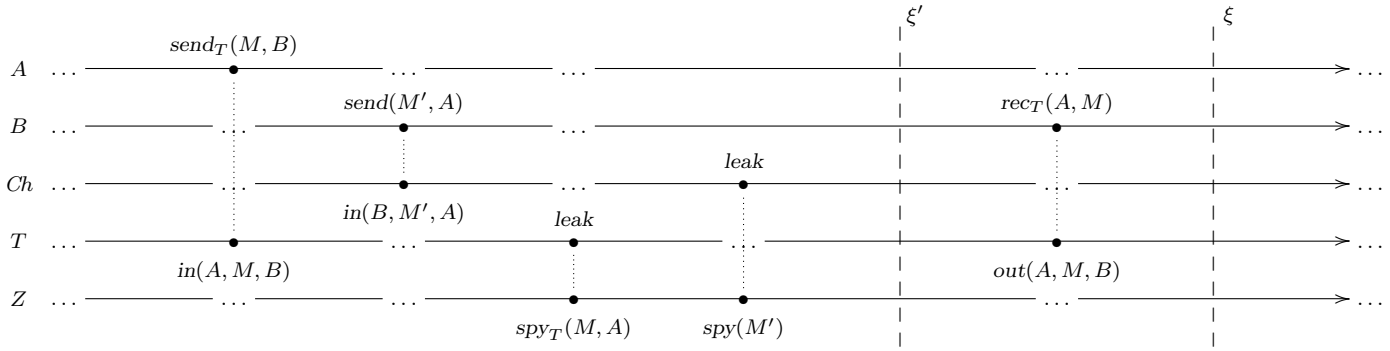
Finally, we can establish the equivalence of the two models. If $TTP \Vdash \gamma$ and $\langle \mu_{DS^\sharp}, \xi \rangle \in DS^\sharp_\Xi$ then $\theta(\mu_{DS^\sharp}, \xi) \in TTP_\Xi$ and thus $\theta(\mu_{DS^\sharp}, \xi) \Vdash \gamma$. Hence, $\mu_{DS^\sharp}, \xi \Vdash \overline{\gamma}^\sharp$ and thus $DS^\sharp \Vdash \overline{\gamma}^\sharp$. Conversely, if $DS^\sharp \Vdash \overline{\gamma}^\sharp$ and $\langle \mu_{TTP}, \xi \rangle \in TTP_\Xi$ then $\beta(\mu_{TTP}, \xi) \in DS^\sharp_\Xi$ and so $\beta(\mu_{TTP}, \xi) \Vdash \overline{\gamma}^\sharp$. Hence, $\mu_{TTP}, \xi \Vdash \gamma$ and thus $TTP \Vdash \gamma$. □

Figure 8 depicts the essential ingredients of the proof of Proposition 5.5 and it closely follows the translation of syntax. Note that the upper and lower diagrams depict two $DS^\sharp$ models that differ only in the linearization of the Ch and T events of the diagram in the middle. Note further that β transforms the middle diagram along with the global state ξ, but the linearization considered would not be compatible with the global state ξ' also depicted.

As a corollary of Proposition 5.5, we have that $DS^\sharp$ fulfills the message-origin authentication requirement originally stated for TTP in Proposition 5.2.



is transformed by θ into



which is transformed back by β into

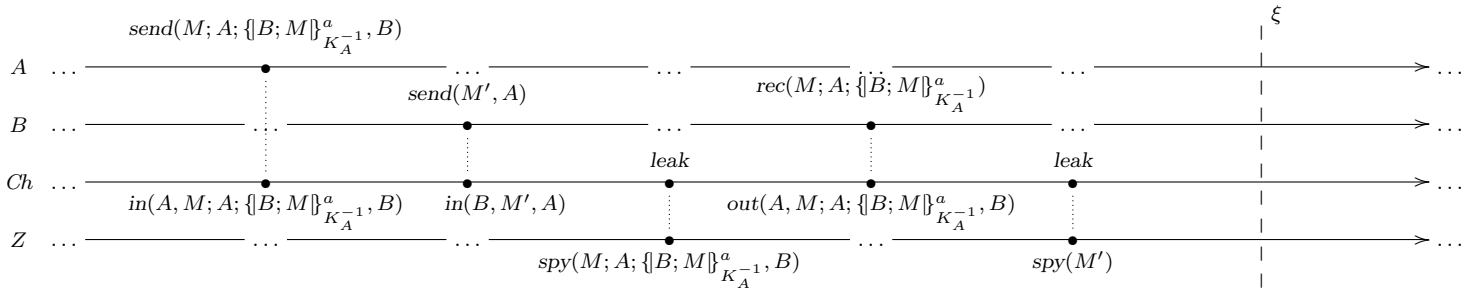


Figure 8: Model transformations between TTP and $DS^\sharp$ models.

Corollary 5.6 $DS^\sharp \Vdash @_A[\mathsf{P}_\circ\, rec(M; B; \{\!|A; M|\!\}^a_{K_B^{-1}})] \Rightarrow @_B[\mathsf{P}_\circ\, send(M; B; \{\!|A; M|\!\}^a_{K_B^{-1}}, A)]$ *for* $A, B \in Princ$ *and* $M \in Msg$.

5.2 Channel-based versus intruder-centric models

We now show that, from the perspective of protocol analysis, the channel-based models we have been using are equivalent to the kinds of intruder-centric models used in many protocol analysis tools [3, 12, 59, 65]. In intruder-centric models, the intruder controls, and is identified with, the network. Moreover, it is even possible to eliminate the honest agents. This avoids the complications of distribution as intruder-centric models correspond to linear models of the network behavior.

Specifically, we will establish a notion called *attack equivalence* between models, which says that attacks are neither lost nor gained by moving between models, where an attack is a countermodel to a security property. In this paper, we restrict our focus to secrecy and authentication properties

as described in Section 3.4.

We will proceed in three phases. In the first phase, we translate our *CB* models to models where the intruder is merged with the channel, which we call *ZB models*. Hence, *ZB* models no longer have a channel and messages sent go directly to the intruder. In the second phase, which we call *step compression*, we reorganize the *ZB* models so that the actions of each principal corresponding to the execution of certain protocol steps appear as consecutive events in local life-cycles. We call these *CZB models*, standing for *compressed ZB* models. In the third and final phase, we drop any explicit reference to the honest principals in *CZB* models, keeping just the intruder. This requires changing the granularity of the actions. These are no longer the atomic actions of sending and receiving messages or generating fresh data, but rather *transactions* corresponding to the execution of complete steps of some fixed protocol by some agent. We call the resulting intruder-centric models *ZC models*.

Below, we introduce all the models and show their attack equivalence by proving properties of the proposed model transformations. Before doing so, we take a closer look at protocols and how we model them. Recall from Section 3 that an Alice-and-Bob-style protocol description corresponds to a sequence of message exchanges

$$\begin{array}{rcccl}
(msg_1) & s_1 \to r_1 & : & (f_{1,1},\ldots,f_{1,t_1}). & M_1 \\
 & \vdots & & & \\
(msg_q) & s_q \to r_q & : & (f_{q,1},\ldots,f_{q,t_q}). & M_q \\
(msg_{q+1}) & s_{q+1} \to r_{q+1} & : & (f_{q+1,1},\ldots,f_{q+1,t_{q+1}}). & M_{q+1} \\
 & \vdots & & & \\
(msg_m) & s_m \to r_m & : & (f_{m,1},\ldots,f_{m,t_m}). & M_m
\end{array}$$

where each $s_i, r_i \in \{a_1,\ldots,a_j\}$ identifies the principal playing one of the j protocol roles. We assume, as is standard and without loss of generality, that $s_{j+1} = r_j$, for $1 \le j < m$.

Whereas we previously considered the protocol run corresponding to each principal given a specific protocol instantiation, we now split the runs in smaller steps. Specifically, consider two consecutive lines of the protocol description, e.g., those labeled msg_q and msg_{q+1}. Then there is a sequence of actions that the principal instantiating $r_q = s_{q+1}$ must execute. Namely, he must receive M_q, freshly generate $f_{q+1,1},\ldots,f_{q+1,t_{q+1}}$, and finally send M_{q+1}. In general, such a protocol description gives rise to $m+1$ protocol steps, where the first protocol step (by the principal instantiating s_1) does not include an initial receiving action and the last protocol step (by the principal instantiating r_m) consists only of receiving the last message.

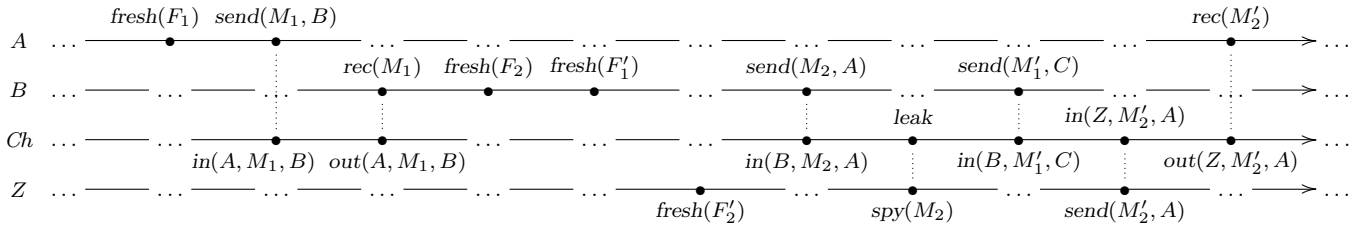
Formally, given a protocol instantiation σ, a protocol role $1 \le i \le j$, and $0 \le k \le m$, we define $step^i_k = msg^i_k \centerdot msg^i_{k+1}$, where we assume that $msg^i_0 = msg^i_{m+1} = \langle\rangle$. Thus, we have:

$$step^i_k = \begin{cases}
\langle fresh(f_{1,1}) \ldots fresh(f_{1,t_1}).send(M_1, r_1)\rangle & \text{if } k = 0 \text{ and } s_1 = a_i \\
\langle rec(M_k).fresh(f_{k+1,1}) \ldots fresh(f_{k+1,t_{k+1}}).send(M_{k+1}, r_{k+1})\rangle & \text{if } 0 < k < m \text{ and} \\
 & \quad r_k = s_{k+1} = a_i \\
\langle rec(M_m)\rangle & \text{if } k = m \text{ and } r_m = a_i \\
\langle\rangle & \text{otherwise}
\end{cases}$$

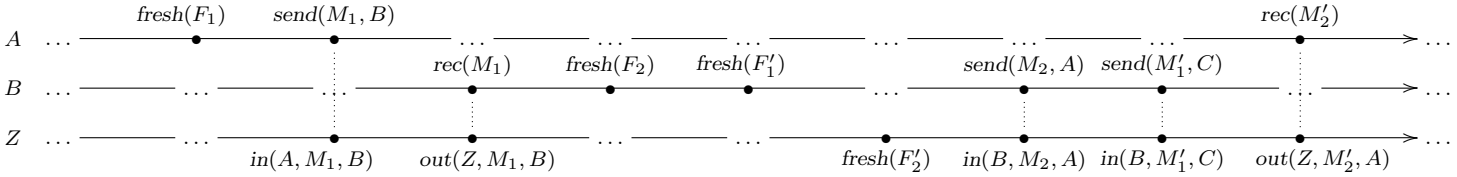
In this way, $run^i = msg^i_1 \centerdot \cdots \centerdot msg^i_m = step^i_0 \centerdot \cdots \centerdot step^i_m$. If $\sigma(a_i) = A$ then we write $step^i_{A,k}(\sigma) = \sigma(step^i_k)$, or simply $step^i_{A,k}(\sigma(\overline{a}), \sigma(\overline{f}))$ where $\overline{a}$ and $\overline{f}$ stand, respectively, for the sequences of identifiers and fresh data symbols used in the protocol description.

Below, in step-compressed models, we will require that the events whose labels match a given protocol step by a principal must be consecutive, that is, protocol steps will not be interleaved with other actions. Then, in intruder-centric models, we will introduce explicit (trans)actions $trans^i_{A,k}(\sigma)$ to represent each protocol step $step^i_{A,k}(\sigma)$. In Figure 9, we illustrate the successive transformation of a *CB* model to a *ZB*, *CZB*, and *ZC* model. For simplicity, we consider there a

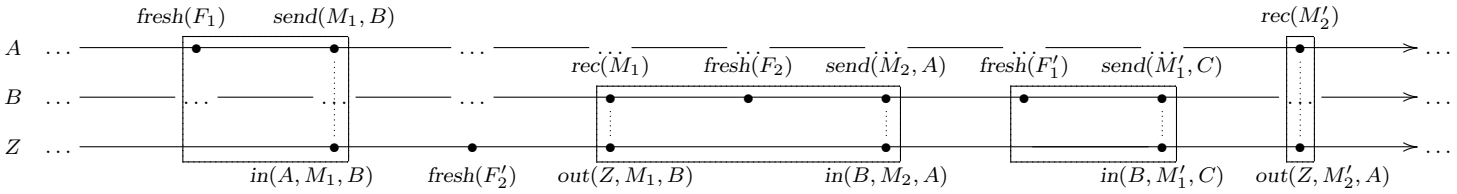
The *CB* model



is attack equivalent (by Corollary 5.13) to the *ZB* model



which is attack equivalent (by Corollary 5.17) to the *CZB* model



which is attack equivalent (by Corollary 5.30) to the *ZC* model

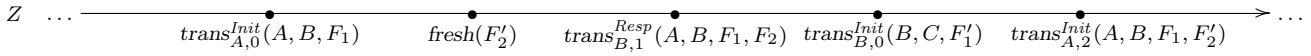


Figure 9: From *CB* to *ZC* models (via *ZB* and *CZB* models)

protocol of the form

$$a \to b \quad : \quad (f_1)\,.\,M_1$$
$$b \to a \quad : \quad (f_2)\,.\,M_2$$

The figure depicts a (possible attack) scenario where honest principals A and B are running the protocol, but B's reply in the second message is intercepted by the intruder, who then sends a fake message to A. While this happens, B is also starting another protocol run with C.

5.2.1 Phase 1: Intruder as the channel

The first phase of our reduction is to *merge* the intruder with the channel, as illustrated by the two topmost models in Figure 9. We call the resulting models *intruder-based models*, or *ZB models* for short. In this step, we keep the same network signature $\langle Princ, Num\rangle$. The *intruder-based signature* Σ_{ZB} is a distributed signature obtained from the network signature, analogously to how we obtained the *channel-based signature* in Section 3.2. However, we now drop the channel, replacing it by the intruder, and the actions *send* and *rec* of the intruder become the actions *in* and *out* of the new intruder. Moreover, when constructing a *ZB* model from a *CB* model, we remove all events representing interaction between the channel and the intruder. Hence, we delete

the actions *spy*, *send*, and *rec* of the intruder and the action *leak* of the channel. The signature $\Sigma_{ZB} = \langle Princ, Act, Prop\rangle$ is such that

- for each $A \in Hon$, Act_A is composed of $send(M, B)$, $rec(M)$, and $fresh(X)$, and $Prop_A$ is composed of the state propositions $knows(M)$, where $B \in Princ$, $M \in Msg$, and $X \in Nonces \uplus SymK \uplus PubK$;
- Act_Z contains $in(A, M, B)$, $out(A, M, B)$, and $fresh(X)$, and $Prop_Z$ contains the state propositions $knows(M)$, where $A, B \in Princ$, $M \in Msg$, and $X \in Nonces \uplus SymK \uplus PubK$.

The *ZB* models μ are those satisfying the axioms **(K)**, for honest principals, **(F1–F2)** of the *CB* model, which guarantee the freshness and uniqueness of nonces, as well as:

($\mathbf{K_Z}$) $\mu_Z, \xi_Z \Vdash_Z knows(M)$ iff $M \in close(\{M' | \mu_Z, \xi_Z \Vdash_Z (\Upsilon\, knows(M')) \vee (\bigvee_{A,B\in Princ} in(A, M', B)) \vee fresh(M')\})$ for every non-empty local state ξ_Z

(Z1) $@_Z[in(A, M, B) \Rightarrow \copyright_A[send(M, B)]]$

(Z2′) $@_Z[out(Z, M, B) \Rightarrow \Upsilon(knows(M) \wedge knows(B))]$

(Z3) $@_Z[out(A, M, B) \Rightarrow \copyright_B[rec(M)]]$

(P1) $@_A[send(M, B) \Rightarrow \Upsilon(knows(M) \wedge knows(B))]$, for every $A \in Hon$

($\mathbf{P_Z2}$) $@_A[send(M, B) \Rightarrow @_Z[in(A, M, B)]]$, for every $A \in Hon$

($\mathbf{P_Z3}$) $@_A[rec(M) \gg_Z out(Z, M, A)]$, for every $A \in Hon$

($\mathbf{P_Z5}$) $@_A[\bigwedge_{B\in Hon\setminus\{A\}} \neg\, \copyright_B[\top]]$

($\mathbf{P_Z6'}$) $@_A[fresh(X) \Rightarrow \neg\, \copyright_Z[\top]]$

($\mathbf{P_Z6''}$) $@_Z[fresh(X) \Rightarrow \bigwedge_{A\neq Z} \neg\, \copyright_A[\top]]$

Axiom **($\mathbf{K_Z}$)** adapts axiom **(K)** for the intruder, which handles the new intruder action *in*, from which Z can learn new messages. The axioms for the channel are adapted to the new model, replacing the channel by the intruder. Axioms **(Z1)** and **(Z3)** (corresponding to **(C1)** and **(C3)**, respectively) are not affected, apart from name replacement. Axiom **(C2)** is replaced by **(Z2')**, as we assume that all outputs are generated by the intruder and therefore do not have corresponding input actions. The only requirement, in this case, is that the intruder knows all the necessary information to produce such messages. Axiom **(C4)** is simply dropped, as spying actions are no longer needed. Some of the axioms for the principals' behavior are also adapted. Axiom **(P1)** is as before but with the restriction to honest agents, which applies also to some of the other axioms. In axioms (**$\mathbf{P_Z2}$**) and (**$\mathbf{P_Z3}$**), the channel is replaced by the intruder. Axiom **(P4)** is dropped as there are no longer *spy* and *leak* actions. Axiom (**$\mathbf{P_Z5}$**) is adapted to allow honest principals to communicate with the intruder but, as before, not with other honest principals. Axiom **(P6)** is split into (**$\mathbf{P_Z6'}$**) and (**$\mathbf{P_Z6''}$**), for honest agents and for the intruder, respectively.

As we changed the signature for the intruder by replacing *send* and *rec* actions by *out* and *in* actions, our formulas must be rewritten accordingly. From now on, given a formula γ over a *CB* signature where *spy* and *leak* do not occur, we denote by $\gamma^\dagger$ the formula over the *ZB* signature obtained from γ by uniformly replacing all occurrences of $send(M, A)$ local to Z by $out(Z, M, A)$, and all occurrences of $rec(M)$ local to Z by $\bigvee_{A\in Hon} in(A, M, Z)$. In particular, $secr_S(\sigma)^\dagger$ is the same as $secr_S(\sigma)$ when all the principals are honest. The situation is similar for authentication, if both principals are honest. Note, however, that Z now controls the channel and outputs all the messages received by honest principals. Hence, it makes no sense for honest principals to authenticate the intruder.

Our goal is to prove that the two models are attack equivalent. We therefore start by defining a model transformation $\beta : CB_\Xi \to ZB_\Xi$. For each $\langle \mu_{CB}, \xi\rangle \in CB_\Xi$, with $\mu_{CB} = \langle \lambda, \alpha, \pi\rangle$, let $\beta(\mu_{CB}, \xi) = \langle\langle \lambda^\dagger, \alpha^\dagger, \pi^\dagger\rangle, \xi^\dagger\rangle$ such that

- $\mu_A^\dagger = \mu_A$ for $A \in \mathit{Hon}$ and $\mu_Z^\dagger$ is such that
  - $\lambda_Z^\dagger = \langle Ev_Z^\dagger, \leq_Z^\dagger \rangle$, where $Ev_Z^\dagger = (Ev_Z \cup Ev_{Ch}) \setminus (Ev_Z \cap Ev_{Ch})$ and $\leq_Z^\dagger$ is some discrete linearization of $\rightarrow_{Ch} \cup \rightarrow_Z$ restricted to $Ev_Z^\dagger$ that has $(\xi|_Z \cup \xi|_{Ch}) \cap Ev_Z^\dagger$ as a local state;
  - $\alpha_Z^\dagger(e) = \begin{cases} \alpha_{Ch}(e) & \text{if } e \in Ev_{Ch} \text{ and } \alpha_{Ch}(e) = \mathit{in}(B,M,C) \text{ for some } B,M,C \\ \mathit{out}(Z,M,C) & \text{if } e \in Ev_{Ch} \text{ and } \alpha_{Ch}(e) = \mathit{out}(B,M,C) \text{ for some } B,M,C \\ \alpha_Z(e) & \text{if } e \in Ev_Z \setminus Ev_{Ch}; \end{cases}$
  - $\pi_Z^\dagger$ is inductively defined as expected, requiring $\pi_Z^\dagger(\emptyset) = \pi_Z(\emptyset)$;
- $\xi_A^\dagger = \xi_A$ and $\xi_Z^\dagger = (\xi_{Ch} \cup \xi_Z) \cap Ev_Z^\dagger$.

Note that $\alpha^\dagger$ is defined so that $\alpha_A^\dagger(e) = \alpha_A(e)^\dagger$, for every $A \in \mathit{Princ}$ and $e \in Ev_A^\dagger$. We claim that $\langle \mu^\dagger, \xi^\dagger \rangle \in ZB_\Xi$. As for the construction of $\mu^\dagger$, observe that the intruder in $\mu^\dagger$ is obtained by *merging* the old intruder with the channel. In doing so, we drop some events, namely all synchronizations between the channel and the (old) intruder, e.g. *leak* and $\mathit{spy}(M_2)$, and $\mathit{in}(Z, M_3, A)$ and $\mathit{send}(M_3, A)$ in the *CB* model in Figure 9. Hence, all that remains from the old intruder are the *fresh* actions. Moreover, the local successor relation $\rightarrow_Z^\dagger$ is defined by a discrete linearization of the union of the successor relations of the channel and the old intruder, e.g., $\mathit{fresh}(F_3)$ is moved before $\mathit{out}(Z, M_3, A)$ in the *ZB* model in Figure 9. Note that it is always possible to linearize these two relations in a way compatible with an existing state ξ. Note also that, as a consequence of the remarks above, the map β is not injective.

We now prove some properties of this model transformation.

Lemma 5.7 *Let $\langle \mu_{CB}, \xi \rangle \in CB_\Xi$, $A \in \mathit{Hon}$ and $\varphi \in \mathcal{L}_A^{\not\mathcal{O}}$ where spy does not occur. Then*

1. *$\mu_{CB}, \xi \Vdash @_A[\varphi]$ if and only if $\beta(\mu_{CB}, \xi) \Vdash @_A[\varphi]$;*
2. *$\mu_{CB}, \xi \Vdash @_A[\mathit{role}_A^i(\sigma)]$ if and only if $\beta(\mu_{CB}, \xi) \Vdash @_A[\mathit{role}_A^i(\sigma)]$.*

Proof: The first condition follows from Lemma 2.2, given that for $\langle \mu^\dagger, \xi^\dagger \rangle = \beta(\mu_{CB}, \xi)$ we have by construction $\mu_A = \mu_A^\dagger$ and $\xi_A = \xi_A^\dagger$. The second condition follows from the first as roles never involve communication formulas. □

In Lemma 5.7, we have that $@_A[\varphi]^\dagger$ equals $@_A[\varphi]$, and $@_A[\mathit{role}_A^i(\sigma)]^\dagger$ equals $@_A[\mathit{role}_A^i(\sigma)]$. Next, we prove a weaker version of this lemma for the intruder.

Lemma 5.8 *Let $\langle \mu_{CB}, \xi \rangle \in CB_\Xi$. If $\mu_{CB}, \xi \Vdash @_Z[\mathit{knows}(M)]$ then $\beta(\mu_{CB}, \xi) \Vdash @_Z[\mathit{knows}(M)]$, for every message M.*

Proof: Let $\langle \mu^\dagger, \xi^\dagger \rangle = \beta(\mu_{CB}, \xi)$. If $\mu_{CB}, \xi \Vdash @_{Ch}[\mathit{in}(A, M, B)]$ then $\mu^\dagger, \xi^\dagger \Vdash @_Z[\mathsf{P}_\circ\, \mathit{in}(A, M, B)]$ so, every message that Z could eventually spy on the *CB* model is received by him in the *ZB* model. Using axiom ($\mathbf{K_Z}$), it follows that $\mu_{CB}, \xi \Vdash @_Z[\mathit{knows}(M)]$ implies that $\mu^\dagger, \xi^\dagger \Vdash @_Z[\mathit{knows}(M)]$, which concludes the proof. □

Note that the converse of Lemma 5.8 fails since in $\beta(\mu_{CB}, \xi)$ the intruder knows all messages sent from one agent to another, which is not the case for $\langle \mu_{CB}, \xi \rangle$. From Lemmas 5.7 and 5.8, it follows that the transformation from *CB* models to *ZB* models is attack preserving, as formalized below.

Proposition 5.9 (Attack preservation) *Let $\langle \mu_{CB}, \xi \rangle \in CB_\Xi$. Then,*

- *an attack on $\mathit{secr}_S(\sigma)$ at $\langle \mu_{CB}, \xi \rangle$ implies an attack on $\mathit{secr}_S(\sigma)^\dagger$ at $\beta(\mu_{CB}, \xi)$;*
- *an attack on $\mathit{auth}_{A,B}^{i,j,q}(\sigma)$ at $\langle \mu_{CB}, \xi \rangle$ implies an attack on $\mathit{auth}_{A,B}^{i,j,q}(\sigma)^\dagger$ at $\beta(\mu_{CB}, \xi)$.*

To prove the converse, we first define a model transformation $\theta : ZB_\Xi \to CB_\Xi$. For each $\langle \mu_{ZB}, \xi^\dagger \rangle \in ZB_\Xi$ with $\mu_{ZB} = \langle \lambda^\dagger, \alpha^\dagger, \pi^\dagger \rangle$ we introduce explicit communication events local to the channel. Hence, let $Fr = \{e \in Ev_Z^\dagger \mid \alpha_Z^\dagger(e) = \mathit{fresh}(X), \text{ for some } X\}$, $Succ = \{e \in Ev_Z^\dagger \mid \alpha_Z^\dagger(e) = \mathit{in}(A, M, B), \text{ for some } A, M, B\}$, $New_S = \{s(e) \mid e \in Succ\}$, $Pred = \{e \in Ev_Z^\dagger \mid \alpha_Z^\dagger(e) = \mathit{out}(Z, M, A), \text{ for some } M, A\}$, $New_P = \{p(e) \mid e \in Pred\}$, and $Sync = Ev_Z^\dagger \setminus Fr$. Note that $s(e)$ and $p(e)$ denote new events representing successor and predecessor events of e in the channel. Then, $\theta(\mu_{ZB}, \xi^\dagger) = \langle \mu, \xi \rangle$ is as follows.

- $\mu_A = \mu_A^\dagger$ for every $A \in Hon$;
- $\mu_{Ch} = \langle \lambda_{Ch}, \alpha_{Ch}, \pi_{Ch} \rangle$ with
  - $\lambda_{Ch} = \langle Ev_{Ch}, \leq_{Ch} \rangle$ where $Ev_{Ch} = (Ev_Z^\dagger \setminus Fr) \cup (New_S \cup New_P)$, and $\to_{Ch}$ is the successor relation obtained from $\leq_Z^\dagger$ by letting $e \to_{Ch} s(e)$ for every $e \in Succ$, and $p(e) \to_{Ch} e$ for every $e \in Pred$;
  - $\alpha_{Ch}(e) = \begin{cases} \alpha_Z^\dagger(e) & \text{if } e \in Ev_Z^\dagger \setminus Fr \\ \mathit{leak} & \text{if } e \in New_S \\ \mathit{in}(Z, M, A) & \text{if } e = p(e') \text{ and } \alpha_Z^\dagger(e') = \mathit{out}(Z, M, A) \end{cases}$
- $\mu_Z = \langle \lambda_Z, \alpha_Z, \pi_Z \rangle$ where
  - $\lambda_Z = \langle Ev_Z, \leq_Z \rangle$ is such that $Ev_Z = Fr \cup New_S \cup New_P$, and $\to_Z$ is obtained from $\to_Z^\dagger$ by replacing every $e \in Succ$ by $s(e)$, and every event in $e \in Pred$ by $p(e)$;
  - $\alpha_Z(e) = \begin{cases} \mathit{spy}(M) & \text{if } e = s(e') \text{ and } \alpha_Z^\dagger(e') = \mathit{in}(A, M, B) \\ \mathit{send}(M, A) & \text{if } e = p(e') \text{ and } \alpha_Z^\dagger(e') = \mathit{out}(Z, M, A) \\ \alpha_Z^\dagger(e) & \text{if } e \in Fr \end{cases}$
  - π_Z is inductively defined as expected, requiring $\pi_Z(\emptyset) = \pi_Z^\dagger(\emptyset)$.
- $\xi_A = \xi_A^\dagger$ for every $A \in Hon$, $\xi_{Ch} = (\xi_Z^\dagger \cap Ev_{Ch}) \cup \{s(e) \in New_S \mid e \in \xi_Z^\dagger\} \cup \{p(e) \in New_P \mid e \in \xi_Z^\dagger\}$, and $\xi_Z = (\xi_Z^\dagger \cap Ev_Z) \cup \{s(e) \in New_S \mid e \in \xi_Z^\dagger\} \cup \{p(e) \in New_P \mid e \in \xi_Z^\dagger\}$.

Note that although θ is not the inverse of β, there is a clear relationship between $\theta(\beta(\mu_{CB}, \xi))$ and $\langle \mu_{CB}, \xi \rangle$. Namely, the former is precisely the *CB* interpretation structure in $\beta^{-1}(\beta(\mu_{CB}, \xi))$ that maximizes the activity of the intruder in terms of spying every possible message from the channel as soon as possible.

We claim that $\theta(\mu_{ZB}, \xi^\dagger) = \langle \mu, \xi \rangle$ is a *CB* model. Recall the *ZB* model (the second) depicted in Figure 9 and consider, for instance, the event $e \in Ev_B^\dagger \cap Ev_Z^\dagger$ such that $\alpha_B^\dagger(e) = \mathit{send}(M_2, A)$ and $\alpha_Z^\dagger(e) = \mathit{in}(B, M_2, A)$. Then, as also shown in the topmost *CB* model, we have two events $e, s(e) \in Ev_{Ch}$ with $e \in Ev_B$ and $s(e) \in Ev_Z$ such that $\alpha_B(e) = \mathit{send}(M_2, A)$, $\alpha_{Ch}(e) = \mathit{in}(B, M_2, A)$, $\alpha_{Ch}(s(e)) = \mathit{leak}$, and $\alpha_Z(s(e)) = \mathit{spy}(M_2)$. Intruder sending actions are introduced similarly. In the translation of the global state, we add to ξ both the events corresponding to incoming messages from other principals that were already in $\xi^\dagger$ and all outgoing messages from the intruder.

The proof of the following lemma is analogous to the proof of Lemma 5.7.

Lemma 5.10 *Let $\langle \mu_{ZB}, \xi^\dagger \rangle \in ZB_\Xi$, $A \in Hon$, and $\varphi \in \mathcal{L}_A^{@}$ where spy does not occur. Then*

1. *$\theta(\mu_{ZB}, \xi^\dagger) \Vdash @_A[\varphi]$ if and only if $\mu_{ZB}, \xi^\dagger \Vdash @_A[\varphi]$;*
2. *$\theta(\mu_{ZB}, \xi^\dagger) \Vdash @_A[\mathit{role}_A^i(\sigma)]$ if and only if $\mu_{ZB}, \xi^\dagger \Vdash @_A[\mathit{role}_A^i(\sigma)]$.*

Observe that in the construction of the translated model, the intruder spies on every message that enters the channel. He therefore has the same knowledge as in the *ZB* model and we can establish the following result.

Lemma 5.11 *Let $\langle \mu_{ZB}, \xi^\dagger \rangle \in ZB_\Xi$. Then*

1. $\theta(\mu_{ZB}, \xi^\dagger) \Vdash @_Z[\mathsf{P}_\circ\, a]$ *if and only if* $\langle \mu_{ZB}, \xi^\dagger \rangle \Vdash @_Z[\mathsf{P}_\circ\, a^\dagger]$, *for non-spying* $a \in Act_Z$ *in* Σ_{CB};

2. $\theta(\mu_{ZB}, \xi^\dagger) \Vdash @_Z[knows(M)]$ *if and only if* $\langle \mu_{ZB}, \xi^\dagger \rangle \Vdash @_Z[knows(M)]$, *for every message* M.

Proof:

1. By the construction of α_Z, the intruder in $\theta(\mu_{ZB}, \xi^\dagger)$ never receives messages. If a is a fresh generation action, then the result is trivial. If $a = send(M, A)$, then it must be the label of an event $p(e')$ such that the label of e' is $a^\dagger = out(Z, M, A)$.

2. The knowledge set computed with $\mathbf{(K_Z)}$ at the given $\langle \mu_{ZB}, \xi^\dagger \rangle$ is the same that can be computed using $(\mathbf{K})$, for principal Z, by just exchanging each $in(A, M, B)$ with $spy(M)$, and recalling that $\pi_Z(\emptyset) = \pi^\dagger_Z(\emptyset)$. □

Lemmas 5.10 and 5.11 yield the following proposition, formalizing *attack reflection*.

Proposition 5.12 (Attack reflection) *Let* $\langle \mu_{ZB}, \xi^\dagger \rangle \in ZB_\Xi$. *Then,*

- *an attack on* $secr_S(\sigma)^\dagger$ *at* $\langle \mu_{ZB}, \xi^\dagger \rangle$ *implies an attack on* $secr_S(\sigma)$ *at* $\theta(\mu_{ZB}, \xi^\dagger)$;
- *an attack on* $auth^{i,j,q}_{A,B}(\sigma)^\dagger$ *at* $\langle \mu_{ZB}, \xi^\dagger \rangle$ *implies an attack on* $auth^{i,j,q}_{A,B}(\sigma)$ *at* $\theta(\mu_{ZB}, \xi^\dagger)$.

As a corollary, we conclude that the two models are attack equivalent.

Corollary 5.13 *The models CB and ZB are attack equivalent.*

5.2.2 Phase 2: Step compression

We now apply *step compression*, a model-reduction technique used to improve the efficiency of various protocol analysis tools, such as [3, 10, 23, 49, 52]. The essence of step compression is that when an honest principal receives a message, he immediately sends a reply. As shown in the third model of Figure 9, this reduction technique groups together entire protocol steps of honest agents into consecutive sequences of actions in the resulting model, as depicted inside the boxes.

Step-compressed models, or *CZB* models, are *ZB* models with an additional restriction. Recall that, for each $A \in Hon$, the sequence of actions labeling A's local life-cycle must interleave prefixes of pairwise independent instantiations of protocol runs for principal A. Let $run^i_A(\sigma) = step^i_{A,0}(\sigma) \centerdot \cdots \centerdot step^i_{A,m}(\sigma) = \langle act_{0,1} \dots act_{0,n_0} \rangle \centerdot \langle act_{1,1} \dots act_{1,n_1} \rangle \centerdot \cdots \centerdot \langle act_{m,1} \dots act_{m,n_m} \rangle = \langle act_{0,1} \dots act_{0,n_0}.act_{1,1} \dots act_{1,n_1} \dots act_{m,1} \dots act_{m,n_m} \rangle$ be one such run and consider the prefix $\langle \dots act_{k,u_k} \rangle$, with $k \leq m$ and $u_k \leq n_k$. Then, for each $0 \leq q \leq k$, we let $u_q = n_q$ and require that there are consecutive events $e_{i_q+1} \rightarrow_A \cdots \rightarrow_A e_{i_q+u_q}$ such that $\alpha_A(e_{i_q+1}) = act_{q,1}, \dots, \alpha_A(e_{i_q+u_q}) = act_{q,u_q}$, i.e., the actions within a step cannot be interleaved with other actions. Below, we refer to the last event $e_{i_q+u_q}$ as the *anchor event* for step q of the run. It should be clear that with the exception of the initial and final steps of a run, every non-empty intermediate step must start with a receiving action and end with a sending action. Thus, we will also require that the corresponding shared events on the intruder side are consecutive. That is, for each $1 \leq q \leq k$, such $q \neq m$ and $u_q = n_q \neq 0$, we require that $e_{i_q+1} \rightarrow_Z e_{i_q+n_q}$.

At the specification level, this corresponds to adding axioms of the form:

$(\mathbf{SC^i_{A,q,u_q}}(\sigma))$ $@_A[act_{q,u_q} \Rightarrow \mathsf{Y}(act_{q,u_q-1} \wedge \mathsf{Y}(\dots \wedge \mathsf{Y}\, act_{q,1} \dots))]$, for each $u_q \leq n_q \neq 0$;

$(\mathbf{ZSC^i_{A,q}}(\sigma))$ $@_Z[©_A[act_{q,n_q}] \Rightarrow \mathsf{Y}\, ©_A[act_{q,1}]]$, for each $0 < q < m$ with $n_q \neq 0$.

We will show that if there is an attack on a security goal in an intruder-based model, then there is also a step-compressed model where the attack happens. The idea is to delay the events corresponding to each protocol step so that they are as close as possible to the step's anchor event, and similarly for the intruder model, to delay the *out* actions so that they are next to the corresponding *in* actions, as described above. We now give the associated construction.

To establish the attack preservation results, we first define a model transformation $\beta : ZB \to CZB$. In this case, we do not consider states in the translation as the attacks on the different security goals may require constructing different global states. For each $\mu_{ZB} = \langle \lambda, \alpha, \pi \rangle \in ZB$, we define the sets of floating events that must be reordered when constructing the step-compressed model as $Flt_A = \{e \in Ev_A \mid e \text{ is not an anchor event}\}$ for each $A \in Hon$, and $Flt_Z = Ev_Z \cap (\bigcup_{A \in Hon} Flt_A)$. Then, $\beta(\mu_{ZB}) = \langle \lambda^\flat, \alpha^\flat, \pi^\flat \rangle$ is as follows:

- for $A \in Hon$, $\mu^\flat_A = \langle \lambda^\flat_A, \alpha^\flat_A, \pi^\flat_A \rangle$ is:
  - $\lambda^\flat_A = \langle Ev^\flat_A, \leq^\flat_A \rangle$ where $Ev^\flat_A = Ev_A$ and $\to^\flat_A$ results from restricting $\leq_A$ to anchor events in $Ev_A \setminus Flt_A$, and requiring that $e_{q,1} \to^\flat_A e_{q,2} \to^\flat_A \cdots \to^\flat_A e_{q,u_q}$ where e_{q,u_q} is the anchor event of the protocol step corresponding to the α_A labels of the events $e_{q,1} \leq_A e_{q,2} \leq_A \cdots \leq_A e_{q,u_q-1} \leq_A e_{q,u_q}$;
  - $\alpha^\flat_A = \alpha_A$ and $\pi^\flat_A(\emptyset) = \pi_A(\emptyset)$;
- for the intruder Z, $\mu^\flat_Z = \langle \lambda^\flat_Z, \alpha^\flat_Z, \pi^\flat_Z \rangle$ where:
  - $\lambda^\flat_Z = \langle Ev^\flat_Z, \leq^\flat_Z \rangle$ where $Ev^\flat_Z = Ev_Z$ and $\to^\flat_Z$ is the restriction of $\leq_Z$ to anchor events in $Ev_Z \setminus Flt_Z$, where we additionally require that $e_{q,1} \to^\flat_Z e_{q,n_q}$ where $e_{q,1} \leq_Z e_{q,n_q}$ are the first and last events of a complete protocol step by some honest principal, and that $\to^\flat_Z$ is compatible with the definition of $\leq^\flat_A$ for all the other floating events;
  - $\alpha^\flat_Z = \alpha_Z$ and $\pi^\flat_Z(\emptyset) = \pi_Z(\emptyset)$.

In $\beta(\mu_{ZB})$, protocol steps are grouped together as illustrated by the boxes in the *CZB* model in Figure 9. This grouping preserves the ordering of actions within runs and therefore also within steps. Only the interleaving of the different steps may change. Hence, $\beta(\mu_{ZB})$ is still a *ZB* model and, since it satisfies the restrictions, it is also a *CZB* model. To show that this transformation is attack preserving, we first prove two auxiliary results.

Lemma 5.14 *Let $\mu_{ZB} \in ZB$ and let ξ be one of its global states. For each global state $\xi^\flat$ of $\beta(\mu_{ZB})$ such that $last(\xi_A) = last(\xi^\flat_A)$, for $A \in Hon$, we have*

$$\mu_{ZB}, \xi \Vdash @_A[role^i_A(\sigma)] \text{ if and only if } \beta(\mu_{ZB}), \xi^\flat \Vdash @_A[role^i_A(\sigma)].$$

Proof: Assume that $\mu_{ZB}, \xi \Vdash @_A[role^i_A(\sigma)]$. Then, $\mu_{ZB_A}, \xi_A \Vdash_A role^i_A(\sigma)$. Assume also that $run^i_A(\sigma) = \langle act_1 \dots act_n \rangle$. Hence, there are $e_1, \dots, e_n \in Ev_A$ such that $e_1 <_A \cdots <_A e_n$, $\alpha_A(e_i) = act_i$, for $i \in \{1, \dots, n\}$ and $last(\xi_A) = e_n$. By construction, $e_1, \dots, e_n \in Ev^\flat_A$ and $\alpha^\flat_A(e_i) = act_i$, for $i \in \{1, \dots, n\}$. Furthermore, since $e_1, \dots, e_n$ are associated with the same protocol run, then $e_1 <^\flat_A \cdots <^\flat_A e_n$ by the construction of $\to^\flat_A$. Hence, as $last(\xi^\flat_A) = last(\xi_A) = e_n$, it follows that $\beta(\mu_{ZB}), \xi^\flat \Vdash @_A[role^i_A(\sigma)]$. The proof of the converse is similar. □

Lemma 5.15 *Let $A \in Hon$, $\mu_{ZB} \in ZB$, and ξ be a global state of μ_{ZB} such that $last(\xi_A) = e_A \in Ev_A \cap Ev_Z$ is an anchor event. Let $\xi^\flat$ be the least global state of $\beta(\mu_{ZB})$ such that $last(\xi^\flat_A) = e_A$. Then $\xi^\flat_Z \subseteq \xi_Z$, and for every $B \in Hon \setminus \{A\}$, if $e \in \xi^\flat_B$ is such that $e \in Ev_B \cap Ev_Z$ is an anchor event, then $e \in \xi_B$.*

Proof: To start with, note that $\xi^\flat = e_A{\downarrow}$ under the global ordering $\leq^\flat$. To show that $\xi^\flat_Z \subseteq \xi_Z$, let $e \in \xi^\flat_Z$. As $e_A \in Ev_Z$, we have that $e \leq^\flat_Z e_A$. Assume, for the sake of contradiction, that $e \notin \xi_Z$. Then $e >_Z e_A$ and from the construction of $\to^\flat_Z$ we conclude that e is not an anchor event. Thus, in the construction, e was moved towards its anchor event, which must also be after e_A. Hence $e >^\flat_Z e_A$, which contradicts $e \leq^\flat_Z e_A$. Finally, let $B \in Hon \setminus \{A\}$ and $e \in \xi^\flat_B$ be such that $e \in Ev_B \cap Ev_Z$ is an anchor event. As both $e, e_A \in Ev_Z$ are anchor events, then $e \leq^\flat_Z e_A$ and the definition of $\to^\flat_Z$ imply that $e \leq_Z e_A$. Therefore, $e \in \xi_Z$ and also $e \in \xi_B$. □

We could also show, under the conditions of the previous lemma, that $\xi^\flat_A \subseteq \xi_A$. However, this fact would not be useful for proving our envisaged result.

Lemma 5.16 (Attack preservation) *Let $\langle \mu_{ZB}, \xi \rangle \in ZB_\Xi$. Then,*

- *an attack on $secr_S(\sigma)^\dagger$ at $\langle \mu_{ZB}, \xi \rangle$ implies an attack on $secr_S(\sigma)$ at some state of $\beta(\mu_{ZB})$;*
- *an attack on $auth^{i,j,q}_{A,B}(\sigma)^\dagger$ at $\langle \mu_{ZB}, \xi \rangle$ implies an attack on $auth^{i,j,q}_{A,B}(\sigma)^\dagger$ at some state of $\beta(\mu_{ZB})$.*

Proof: Let $\mu^\flat = \beta(\mu_{ZB})$. Assume that an attack on $secr_S(\sigma)^\dagger$ happens at the global state ξ of μ_{ZB}. Let $\xi^\flat$ be the least global of $\mu^\flat$ that contains ξ. Then, $\mu_{ZB}, \xi \Vdash \bigwedge_{i=1}^{j} @_{A_i}[\mathsf{P}_\circ\ role^i_{A_i}(\sigma)]$ and $\mu_{ZB}, \xi \not\Vdash \bigwedge_{B \in Princ \setminus \{A_1,\ldots,A_j\}} \bigwedge_{M \in S} @_B[\neg\, knows(M)]$. From Lemma 5.14, since $\xi \subseteq \xi^\flat$, it follows that $\mu^\flat, \xi^\flat \Vdash \bigwedge_{i=1}^{j} @_{A_i}[\mathsf{P}_\circ\ role^i_{A_i}(\sigma)]$. Furthermore, as $\xi^\flat$ extends ξ and the labeling of events does not change, we also have that $\mu^\flat, \xi^\flat \not\Vdash \bigwedge_{B \in Princ \setminus \{A_1,\ldots,A_j\}} \bigwedge_{M \in S} @_B[\neg\, knows(M)]$.

Assume now that an attack on $auth^{i,j,q}_{A,B}(\sigma)^\dagger$ happens at the global state ξ of μ_{ZB}. Let $\xi^\flat$ be the least global state of $\mu^\flat$ such that $last(\xi^\flat_A) = last(\xi_A)$. Then, $\mu_{ZB}, \xi \Vdash @_A[role^i_A(\sigma)]$ and $\mu_{ZB}, \xi \not\Vdash @_B[\mathsf{P}_\circ\, send(\sigma(M), A)]^\dagger$. It follows from Lemma 5.14 that $\mu^\flat, \xi^\flat \Vdash @_A[role^i_A(\sigma)]$. Note also that any event of $B \in Hon \setminus \{A\}$ labeled with a sending action is necessarily an anchor event shared with the channel. Thus, Lemma 5.15 guarantees that $\mu^\flat, \xi^\flat \not\Vdash @_B[\mathsf{P}_\circ\, send(\sigma(M), A)]^\dagger$. □

Hence, the two models are attack equivalent. Note that we do not state a reflection result because any *CZB* model is also a *ZB* model.

Corollary 5.17 *The models ZB and CZB are attack equivalent.*

5.2.3 Phase 3: Intruder-centric models

In our last translation step, we "forget" the honest agents and keep only the intruder, as illustrated in the last two models of Figure 9. An *intruder-centric signature* Σ_{ZC} is a distributed signature obtained from the network signature with just one agent identifier, Z, for the intruder. Hence, $\Sigma_{ZC} = \langle \{Z\}, Act, Prop \rangle$ where, recalling the Alice-and-Bob-style protocol description above:

- Act_Z contains all the actions $trans^i_{A,k}(\sigma)$, for each $A \in Princ$, each protocol role $0 \leq i \leq j$, each $0 \leq k \leq m$, and each protocol instantiation σ, as well as $fresh(X)$ for $X \in Nonces \uplus SymK \uplus PrivK$;
- $Prop_Z$ contains all the state propositions $knows_A(M)$, for every $A \in Princ$ and $M \in Msg$.

Note that we are considering the same network signature $\langle Princ, Num \rangle$ as before.

Before we proceed to establish the preservation results, we need some auxiliary notation:

- $recs(\langle act_1 \ldots act_k \rangle) = \{M \mid act_i = rec(M) \text{ for some } 1 \leq i \leq k\}$;
- $snds(\langle act_1 \ldots act_k \rangle) = \{M \mid act_i = send(M, A) \text{ for some } 1 \leq i \leq k \text{ and } A\}$;
- $fshs(\langle act_1 \ldots act_k \rangle) = \{X \mid act_i = fresh(X) \text{ for some } 1 \leq i \leq k\}$.

We define the knowledge learned by a $B \in Hon$ in a transaction as follows:

- $learn_B(trans^i_{A,k}(\sigma)) = \begin{cases} \emptyset & \text{if } A \neq B \\ recs(step^i_{A,k}(\sigma)) \cup fshs(step^i_{A,k}(\sigma)) & \text{otherwise;} \end{cases}$
- $learn_B(fresh(X)) = \emptyset$.

The knowledge learned by the intruder in each action, assuming $step^i_{A,k}(\sigma) = \langle act_1 \ldots act_{n_k} \rangle$, is:

- $learn_Z(trans^i_{A,k}(\sigma)) = \begin{cases} \emptyset & \text{if } act_{n_k} \neq send(M, A) \text{ for any } A, M \\ snds(step^i_{A,k}(\sigma)) & \text{otherwise;} \end{cases}$

- $learn_Z(fresh(X)) = \{X\}$.

Next, we rewrite all the relevant axioms in this new setting. These include axioms about knowledge, axioms to guarantee freshness and uniqueness of the data generated by each principal, and principal axioms among others.

(K$^\bullet$) $\mu, \xi \Vdash knows_A(M)$ iff $M \in close(\{M' | \mu, \xi \Vdash \mathsf{Y}\, knows_A(M')\} \cup learn_A(\alpha(last(\xi))))$, for every state ξ

(F1$^\bullet_{\mathbf{Z}}$) $@_Z[fresh(X) \Rightarrow \mathsf{Y}(\neg\, knows_Z(M))]$

(F2$^\bullet_{\mathbf{Z}}$) $@_Z[fresh(X) \Rightarrow \bigwedge_{A \in Hon} \neg\, knows_A(M)]$

(F1$^\bullet_{\mathbf{A}}$) $@_Z[trans^i_{A,k}(\sigma) \Rightarrow \mathsf{Y}(\neg\, knows_A(M'))]$

(F2$^\bullet_{\mathbf{A}}$) $@_Z[trans^i_{A,k}(\sigma) \Rightarrow \bigwedge_{B \in Princ \setminus \{A} \neg\, knows_B(M')]$

Here M ranges over all messages such that $cont(X) \cap cont(M) \neq \emptyset$ and M' ranges over all messages such that $cont(X') \cap cont(M') \neq \emptyset$, for every $X' \in fshs(step^i_{A,k}(\sigma))$. Observe that these are exactly the same axioms as $(\mathbf{F1} - \mathbf{2})$, but written in this new language.

Next, we rewrite the axioms about keys, where we assume that $A, B \in Princ$ and $A \neq B$.

(aKey1$^\bullet$) $@_Z[* \Rightarrow knows_A(K_A^{-1})]$

(aKey2$^\bullet$) $@_Z[* \Rightarrow \neg\, knows_B(M)]$, for every M containing K_A^{-1}

(sKey1.1$^\bullet$) $@_Z[* \Rightarrow knows_A(K_{AB})]$

(sKey1.2$^\bullet$) $@_Z[* \Rightarrow knows_B(K_{AB})]$

(sKey2$^\bullet$) $@_Z[* \Rightarrow \neg\, knows_C(M)]$, for every $C \in Princ \setminus \{A, B\}$ and every M containing K_{AB}

We may also assume, for simplicity, that all principals $A, B \in Princ$ know each other's names and public keys from the start.

(N$^\bullet$) $@_Z[* \Rightarrow knows_A(B)]$

(PK$^\bullet$) $@_Z[* \Rightarrow knows_A(K_B)]$

In order for μ to be a model of the protocol, we require that it satisfies the above axioms and also that the projection of the trace on each honest agent A, after expanding the actions, is still a legal run, i.e. an interleaving of prefixes of runs (as before). We delay the details of this expansion until the definition of the model transformation θ from ZC models to CZB models.

Observe that our notion of model requires, for every $trans^i_{A,q}(\sigma)$ and $trans^i_{A,k}(\sigma)$ with $k < q$, that $@_Z[trans^i_{A,q}(\sigma) \Rightarrow \mathsf{P}\, trans^i_{A,k}(\sigma)]$.

The change in the language also affects how our security goals are written. Consider first the notion of *role*. In this setting, if $run^i_A(\sigma) = step^i_{A,k_1} \centerdot \cdots \centerdot step^i_{A,k_q}$ then

$$role^i_A(\sigma)^\bullet \equiv trans^i_{A,k_q}(\sigma) \wedge \mathsf{P}(trans^i_{A,k_q-1}(\sigma) \wedge \mathsf{P}(\ldots \wedge \mathsf{P}\, trans^i_{A,k_1}(\sigma))).$$

To express that the messages in a finite set S will remain a shared secret between participants $A_1, \ldots, A_j$ after the complete execution of a protocol under the instantiation σ, with $\sigma(a_i) = A_i$, we write:

$$@_Z \left[\left(\bigwedge_{i=1}^{j} \mathsf{P}_\circ\ role^i_{A_i}(\sigma)^\bullet \right) \Rightarrow \left(\bigwedge_{M \in S} \bigwedge_{B \in Princ \setminus \{A_1, \ldots, A_j\}} \neg\, knows_B(M) \right) \right]$$

and denote this formula by $secr_S(\sigma)^\bullet$.

As for authentication properties, let σ be a protocol instantiation such that $\sigma(a_i) = A \in \mathit{Hon}$ and $\sigma(a_j) = B \in \mathit{Hon}$. Then the property that A authenticates B in role j at step q of the protocol can be defined by the following formula $\mathit{auth}^{i,j,q}_{A,B}(\sigma)^\bullet$, assuming that msg_q of the protocol requires that B sends the message M to A:

$$@_Z[\mathit{role}^i_A(\sigma)^\bullet \Rightarrow \mathsf{P}_\circ\, \mathit{trans}^j_{B,q}(\sigma)]$$

Note that it must be the case that $\mathit{step}^j_{B,q}(\sigma) = \langle \mathit{act}_1 \ldots \mathit{act}_k \rangle$ and $\mathit{act}_k = \mathit{send}(M, A)$.

We now proceed to prove that the two models are attack equivalent. We start by defining the model transformation $\theta : \mathit{ZC} \to \mathit{CZB}$ from ZC models to SC models, which is considerably easier than the converse direction. The following notation will be useful. Let $\mu_{ZC} = \langle \lambda^\bullet, \alpha^\bullet, \pi^\bullet \rangle \in \mathit{ZC}$ with $\lambda = \langle \mathit{Ev}^\bullet, \to^\bullet \rangle$ and $A \in \mathit{Hon}$. We denote by Ev_A the events of $\mathit{Ev}^\bullet$ that involve actions from A and we will need as many as the number of atomic actions in each of the corresponding steps. Hence, we define

$$\mathit{Ev}_A = \{(e,k) \mid e \in \mathit{Ev}^\bullet_Z,\ \alpha^\bullet(e) = \mathit{trans}^i_{A,q}(\sigma) \text{ and } 1 \le k \le |\mathit{step}^i_{A,q}(\sigma)|\}\,.$$

We also need a similar construction for the intruder:

$$\begin{array}{rcl} \mathit{Ev}_Z & = & \{(e,1) \mid e \in \mathit{Ev}^\bullet_Z,\ \alpha^\bullet_Z(e) = \mathit{trans}^i_{A,q}(\sigma) \text{ and } (\mathit{step}^i_{A,q}(\sigma))_1 = \mathit{rec}(M) \text{ for some } M\} \cup \\ & & \{(e,k) \mid e \in \mathit{Ev}^\bullet_Z,\ \alpha^\bullet_Z(e) = \mathit{trans}^i_{A,q}(\sigma),\ |\mathit{step}^i_{A,q}(\sigma)| = k \text{ and} \\ & & \quad |\mathit{step}^i_{A,q}(\sigma)|_k = \mathit{send}(M,B) \text{ for some } B, M\} \cup \\ & & \{(e,1) \mid e \in \mathit{Ev}^\bullet_Z \text{ and } \alpha^\bullet_Z(e) = \mathit{fresh}(X) \text{ for some } X\}\,. \end{array}$$

Then $\theta(\mu_{ZC}) = \langle \lambda, \alpha, \pi \rangle$ is as follows:

- $\mu_A = \langle \lambda_A, \alpha_A, \pi_A \rangle$, for each $A \in \mathit{Hon}$, is
  - $\lambda_A = \langle \mathit{Ev}_A, \le_A \rangle$ such that $(e_1,k_1) \le_A (e_2,k_2)$ if $e_1 \le^\bullet_Z e_2$, or $e_1 = e_2$ and $k_1 < k_2$;
  - $\alpha_A((e,k)) = (\mathit{step}^i_{A,q}(\sigma))_k$ if $\alpha^\bullet(e) = \mathit{trans}^i_{A,q}$;
  - $\pi_A(\emptyset) = \{\mathit{knows}(M) \mid \mathit{knows}_A(M) \in \pi^\bullet_Z(\emptyset)\}$;
- $\mu_Z = \langle \lambda_Z, \alpha_Z, \pi_Z \rangle$ is
  - $\lambda_Z = \langle \mathit{Ev}_Z, \le_Z \rangle$ such that $(e_1,k_1) \le_Z (e_2,k_2)$ if $e_1 \le^\bullet_Z e_2$, or $e_1 = e_2$ and $k_1 < k_2$;
  - $\alpha_Z((e,k)) = \begin{cases} \mathit{out}(Z,M,A) & \text{if } \alpha_A((e,k)) = \mathit{rec}(M), \text{ for the unique } A \in \mathit{Hon} \\ & \text{such that } (e,k) \in \mathit{Ev}_A \\ \mathit{in}(A,M,B) & \text{if } \alpha_A((e,k)) = \mathit{send}(M,B), \text{ for the unique } A \in \mathit{Hon} \\ & \text{such that } (e,k) \in \mathit{Ev}_A \\ \alpha^\bullet_Z(e) & \text{otherwise;} \end{cases}$
  - $\pi_Z(\emptyset) = \{\mathit{knows}(M) \mid \mathit{knows}_Z(M) \in \pi^\bullet_Z(\emptyset)\}$.

Our construction of CZB models is quite intuitive. We simply expand each action $\mathit{trans}^j_{A,q}(\sigma)$ into its corresponding atomic actions. For instance, in Figure 9, $\mathit{trans}^{\mathit{Init}}_{A,0}$ in the ZC model is expanded to $\mathit{fresh}(F_1) \to_A \mathit{send}(M_1, B)$ for A and to $\mathit{in}(A, M_1, B)$ for Z in the corresponding CZB model.

Given the above translation, it is straightforward to define a translation for states. Given a ZC state $\xi^\bullet$ then ξ is the CZB state such that $\xi_A = \{(e,k) \in \mathit{Ev}_A \mid e \in \xi^\bullet_Z\}$ for each $A \in \mathit{Princ}$.

Next, we prove the reflection of attacks on security goals from ZC models to CZB models. We start with some preliminary results.

Lemma 5.18 *Let $\mu_{ZC} \in \mathit{ZC}$ and let $\xi^\bullet$ be one of its states. Then, for every $A \in \mathit{Princ}$*

$$\mu_{ZC}, \xi^\bullet \Vdash @_Z[\mathit{knows}_A(M)] \text{ if and only if } \theta(\mu_{ZC}), \xi \Vdash @_A[\mathit{knows}(M)], \text{ for every } M.$$

Proof: Assume that $A \in \mathit{Hon}$ and let $\mu = \theta(\mu_{ZC})$. Then, $\mu_{ZC}, \xi^\bullet \Vdash @_Z[\mathit{knows}_A(M)]$ iff $\mathit{knows}_A(M) \in \pi_Z^\bullet(\xi^\bullet)$ iff $M \in \mathit{close}(\{M' \mid \mathit{knows}_A(M') \in \pi_Z^\bullet(\emptyset)\} \cup \bigcup_{e \in \xi^\bullet} \mathit{learn}_A(\alpha_Z^\bullet(e)))$ iff $M \in \mathit{close}(\{M' \mid \mathit{knows}(M') \in \pi_A(\emptyset)\} \cup \bigcup_{e \in \xi_A} \mathit{recs}(\alpha_A(e)) \cup \mathit{fshs}(\alpha_A(e))$ iff $\mathit{knows}(M) \in \pi_A(\xi_A)$ iff $\mu, \xi \Vdash @_A[\mathit{knows}(M)]$.

For the intruder, we have that $\mu_{ZC}, \xi^\bullet \Vdash @_Z[\mathit{knows}_Z(M)]$ iff $\mathit{knows}_Z(M) \in \pi_Z^\bullet(\xi^\bullet)$ iff $M \in \mathit{close}(\{M' \mid \mathit{knows}_Z(M') \in \pi_Z^\bullet(\emptyset)\} \cup \bigcup_{e \in \xi^\bullet} \mathit{learn}_Z(\alpha_Z^\bullet(e)))$ iff $M \in \mathit{close}(\{M' \mid \mathit{knows}(M') \in \pi_Z(\emptyset)\} \cup (\bigcup_{A \in \mathit{Hon}} \bigcup_{e \in (\xi_Z \cap \mathit{Ev}_A)} \mathit{snds}(\alpha_A(e))) \cup (\bigcup_{A \in \mathit{Hon}} \bigcup_{e \in (\xi_Z \setminus \mathit{Ev}_A)} \mathit{fshs}(\alpha_Z(e)))$ iff $\mathit{knows}(M) \in \pi_Z(\xi_Z)$ iff $\mu, \xi \Vdash @_Z[\mathit{knows}(M)]$. □

Lemma 5.19 *Let $\mu_{ZC} \in ZC$ and let $\xi^\bullet$ be one of its states. Then, for every $A \in \mathit{Hon}$*

$$\mu_{ZC}, \xi^\bullet \Vdash @_Z[\mathit{trans}^i_{A,q}(\sigma)] \text{ if and only if } \theta(\mu_{ZC}), \xi \Vdash \mathbf{SC^i_{A,q,u_q}}(\sigma).$$

Proof: Let $\mu = \theta(\mu_{ZC})$. Then, $\mu_{ZC}, \xi^\bullet \Vdash @_Z[\mathit{trans}^i_{A,q}(\sigma)]$ iff $\mathit{last}(\xi_Z^\bullet) = e$ and $\alpha_Z^\bullet(e) = \mathit{trans}^i_{A,q}(\sigma)$ iff, by definition of μ_A, there exist events $(e,1), \ldots, (e,u_q) \in \mathit{Ev}_A$ such that $(e,1) \rightarrow_A \cdots \rightarrow_A (e,u_q)$ and $\alpha_A((e,j)) = \mathit{act}_{q,j}$, for $j \in \{1, \ldots, u_q\}$, iff $\mu, \xi \Vdash \mathbf{SC^i_{A,q,u_q}}(\sigma)$, observing that $\mathit{last}(\xi_A) = (e, u_q)$. □

An immediate consequence of the previous result is the following lemma.

Lemma 5.20 *Let $\mu_{ZC} \in ZC$ and let $\xi^\bullet$ be one of its states. Then, for every $A \in \mathit{Hon}$*

$$\mu_{ZC}, \xi^\bullet \Vdash @_Z[\mathit{role}^i_A(\sigma)^\bullet] \text{ if and only if } \theta(\mu_{ZC}), \xi \Vdash @_A[\mathit{role}^i_A(\sigma)].$$

These results allow us to establish the following proposition.

Proposition 5.21 (Attack reflection) *Let μ_{ZC} and let $\xi^\bullet$ be one of its local states. Then,*

- *an attack on $\mathit{secr}_S(\sigma)^\bullet$ at $\langle \mu_{ZC}, \xi^\bullet \rangle$ implies an attack on $\mathit{secr}_S(\sigma)^\dagger$ at $\langle \theta(\mu_{ZC}), \xi \rangle$;*
- *an attack on $\mathit{auth}^{i,j,q}_{A,B}(\sigma)^\bullet$ at $\langle \mu_{ZC}, \xi^\bullet \rangle$ implies an attack on $\mathit{auth}^{i,j,q}_{A,B}(\sigma)^\dagger$ at $\langle \theta(\mu_{ZC}), \xi \rangle$.*

Proof: Let $\mu = \theta(\mu_{ZC})$. Assume that an attack on $\mathit{secr}_S(\sigma)^\bullet$ happens at the global state $\xi^\bullet$ of μ_{ZC}. Then, it must be the case that $\mu_{ZC}, \xi^\bullet \Vdash @_Z[\bigwedge_{i=1}^j \mathsf{P}_\circ \mathit{role}^i_{A_i}(\sigma)^\bullet]$ and also that $\mu_{ZC}, \xi^\bullet \nVdash @_Z[\bigwedge_{M \in S} \bigwedge_{B \in \mathit{Princ} \setminus \{A_1, \ldots, A_j\}} \neg\, \mathit{knows}_B(M)]$. From Lemma 5.20, it follows that $\mu, \xi \Vdash \bigwedge_{i=1}^j @_{A_i}[\mathsf{P}_\circ\, \mathit{role}^i_{A_i}(\sigma)]$, and from Lemma 5.18, it follows that $\mu, \xi \nVdash \bigwedge_{M \in S} \bigwedge_{B \in \mathit{Princ} \setminus \{A_1, \ldots, A_j\}} @_B[\neg\, \mathit{knows}(M)]$. Hence, an attack on $\mathit{secr}_S(\sigma)^\dagger$ happens at the global state ξ of μ.

Assume now that an attack on $\mathit{auth}^{i,j,q}_{A,B}(\sigma)^\bullet$ happens at the global state $\xi^\bullet$ of μ_{ZC}, for $B \in \mathit{Hon}$. Then, $\mu_{ZC}, \xi^\bullet \Vdash @_Z[\mathit{role}^i_A(\sigma)^\bullet]$ and $\mu_{ZC}, \xi^\bullet \nVdash @_Z[\mathsf{P}_\circ\, \mathit{trans}^j_{B,q}(\sigma)]$. From Lemma 5.20, it follows that $\mu, \xi \Vdash @_A[\mathit{role}^i_A(\sigma)]$, and from Lemma 5.19, it follows that $\mu, \xi \Vdash \mathsf{P}_\circ\, \mathbf{SC^i_{B,q,u_q}}(\sigma)$[5]. Note that, by the construction of μ and ξ, and by the fact that $B \in \mathit{Hon}$ and, thus, is following his role in the protocol, it cannot be the case that $\mu, \xi \Vdash @_B[\mathsf{P}_\circ\, \mathit{act}_{q,u_q}]$ and $\mu, \xi \nVdash @_B[\mathsf{P}_\circ\, \mathsf{Y}(\mathit{act}_{q,u_q-1} \wedge \mathsf{Y}(\ldots \wedge \mathsf{Y}\, \mathit{act}_{q,1}))]$. Furthermore, by the definition of $\mathit{auth}^{i,j,q}_{A,B}(\sigma)^\bullet$, we have that $\mathit{act}_{q,u_q} = \mathit{send}(M, A)$. Hence, $\mu, \xi \nVdash @_B[\mathsf{P}_\circ\, \mathit{act}_{q,u_q}]$. Thus, an attack on $\mathit{auth}^{i,j,q}_{A,B}(\sigma)^\bullet$ happens at the global state ξ of μ. □

Now, we prove the converse: if an attack on a security goal occurs in a *CZB* model it is possible to mimic that attack in a *ZC* model. In this case, we consider two different model transformations depending on the type of attack that we are considering. This has to do with the way we treat incomplete steps. In first transformation, we include the incomplete steps in the resulting model as we will use this translation to show attack preservation on secrecy properties. Then, we consider a second transformation where we forget all the incomplete steps and use this to show attack preservation on authentication properties.

Let $\mu_{CZB} = \langle \lambda, \alpha, \pi \rangle \in \mathit{CZB}$. We consider the following sets of events:

[5]Here, we write $\mathsf{P}_\circ\, \mathbf{SC^i_{B,q,u_q}}(\sigma)$ for $@_B[\mathsf{P}_\circ(\mathit{act}_{q,u_q} \Rightarrow \mathsf{Y}(\mathit{act}_{q,u_q-1} \wedge \mathsf{Y}(\ldots \wedge \mathsf{Y}\, \mathit{act}_{q,1} \ldots)))]$.

- $Ev_f^\bullet = \{e \in Ev_Z \mid \alpha_Z(e) = \mathit{fresh}(X), \text{ for some } X\}$,
- $Ev_c^\bullet = \bigcup_{A \in Hon}\{e \in Ev_Z \mid e \text{ is an anchor event in some run of } A\}$,
- $Ev_a^\bullet = \bigcup_{A \in Hon}\{e \in Ev_A \mid e \text{ is an anchor event in some run of } A\}$.

Note that the set $Ev_c^\bullet$ contains the (anchor) events corresponding to all the steps that were completed. In contrast, $Ev_a^\bullet$ contains all (anchor) events of steps that were started, including the ones corresponding to incomplete steps. Observe that if a step was completed, then the anchor event is shared between the agent and the intruder (the last action of a step is always a *send* or a *rec*). If the step was not completed then the anchor event might not be shared with the intruder, but will always be an event of the agent performing that step. In fact, $Ev_c^\bullet \subseteq Ev_a^\bullet$.

The first model transformation that we consider is $\beta^1 : CZB \to ZC$, which we use to show the preservation of attacks on secrecy properties. In this case, $\beta^1(\mu_{CZB}) = \langle \lambda^\bullet, \alpha^\bullet, \pi^\bullet \rangle$ is as follows:

- $Ev_Z^\bullet = Ev_f^\bullet \cup Ev_a^\bullet$ and $\leq_Z^\bullet$ is any linearization of $Ev_Z^\bullet$ compatible with $\leq_Z$;
- $\alpha_Z^\bullet(e) = \begin{cases} \alpha_Z(e) & \text{if } e \in Ev_f^\bullet \\ \mathit{trans}^i_{A,q}(\sigma) & \text{if } e \in Ev_a^\bullet \text{ and } e \text{ occurs in } \mathit{step}^i_{A,q}(\sigma)\,; \end{cases}$
- $\pi_Z^\bullet(\emptyset) = \bigcup_{A \in Princ}\{\mathit{knows}_A(M) \mid \mathit{knows}(M) \in \pi_A(\emptyset)\}$.

In the following, given a global state ξ of μ_{CZB}, we denote by $\xi^\bullet$ the global state of $\beta^1(\mu_{CZB})$ such that $\xi_Z^\bullet = (\xi_Z \cap Ev_f^\bullet) \cup (\bigcup_{A \in Hon} \xi_A \cap Ev_a^\bullet)$.

Lemma 5.22 *Let $\mu_{CZB} \in CZB$ and let ξ be one of its local states. Then, for every $A \in Hon$,*

$$\mathit{if}\ \mu_{CZB}, \xi \Vdash \mathbf{SC}^{\mathbf{i}}_{\mathbf{A},\mathbf{q},\mathbf{u_q}}(\sigma)\ \mathit{then}\ \beta^1(\mu_{CZB}), \xi^\bullet \Vdash @_Z[\mathsf{P}_\circ\, \mathit{trans}^i_{A,q}(\sigma)].$$

Proof: Let $\mu^\bullet = \beta^1(\mu_{CZB})$. If $\mu_{CZB}, \xi \Vdash \mathbf{SC}^{\mathbf{i}}_{\mathbf{A},\mathbf{q},\mathbf{u_q}}(\sigma)$, then there exist $e_1 \to_A \cdots \to_A e_{u,q}$ such that $\alpha_A(e_j) = (\mathit{step}^i_{A,q}(\sigma))_j$. In particular, e_{u_q} is an anchor and thus $e_{u_q} \in \xi^\bullet$. Furthermore, $\alpha_Z^\bullet(e_{u_q}) = \mathit{trans}^i_{A,q}(\sigma)$ and, so, $\mu^\bullet, \xi^\bullet \Vdash @_Z[\mathsf{P}_\circ\, \mathit{trans}^i_{A,q}(\sigma)]$. □

Lemma 5.23 *Let $\mu_{CZB} \in CZB$ and let ξ be one of its local states. Then, for every $A \in Hon$,*

$$\mathit{if}\ \mu_{CZB}, \xi \Vdash @_A[\mathit{role}^i_A(\sigma)]\ \mathit{then}\ \beta^1(\mu_{CZB}), \xi^\bullet \Vdash @_Z[\mathsf{P}_\circ\, \mathit{role}^i_A(\sigma)^\bullet].$$

Proof: Let $\beta^1(\mu_{CZB}) = \mu^\bullet$ and $\mathit{run}^i_A(\sigma) = \mathit{step}^i_{A,0}(\sigma) \centerdot \cdots \centerdot \mathit{step}^i_{A,m}(\sigma) = \langle \mathit{act}_{0,1} \ldots \mathit{act}_{0,n_0} \cdot \mathit{act}_{1,1} \ldots \mathit{act}_{1,n_1} \ldots \mathit{act}_{m,1} \ldots \mathit{act}_{m,n_m} \rangle$. If $\mu_{CZB}, \xi \Vdash @_A[\mathit{role}^i_A(\sigma)]$ then there are $e_{0,1} \to_A \cdots \to_A e_{0,n_0} \leq_A \cdots \leq_A e_{m,1} \to_A \cdots \to_A e_{m,n_m}$ such that $\alpha_A(e_{j,k}) = \mathit{act}_{j,k}$. In particular, e_{j,n_j}, with $j \in \{0, \ldots, m\}$, are all anchor events and are all shared with Z. Hence, $e_{0,n_0} \leq_Z \cdots \leq_Z e_{m,n_m}$ and, furthermore, $e_{j,n_j} \in Ev_Z^\bullet$, for $j \in \{0, \ldots, m\}$. Thus, $e_{j,n_j} \in \xi_Z^\bullet$, $\alpha_Z^\bullet(e_{j,n_j}) = \mathit{trans}^i_{A,j}(\sigma)$, and $e_{0,n_0} \leq_Z^\bullet \cdots \leq_Z^\bullet e_{m,n_m}$. Hence $\mu^\bullet, \xi^\bullet \Vdash @_Z[\mathsf{P}_\circ\, \mathit{role}^i_A(\sigma)^\bullet]$. □

Lemma 5.24 *Let $\mu_{CZB} \in CZB$ and let ξ be one of its local states. Then, for every $A \in Princ$,*

$$\mathit{if}\ \mu_{CZB}, \xi \Vdash @_A[\mathit{knows}(M)]\ \mathit{then}\ \beta^1(\mu_{CZB}), \xi^\bullet \Vdash @_Z[\mathit{knows}_A(M)].$$

Proof: We start by proving that $\bigcup_{e \in \xi_A} \mathit{recs}(\alpha_A(e)) \cup \mathit{fshs}(\alpha_A(e)) \subseteq \bigcup_{e \in \xi_Z^\bullet} \mathit{learn}_A(\alpha_Z^\bullet(e))$. Let $e \in \xi_A$ such that $\alpha_A(e)$ occurs in some $\mathit{step}^i_{A,q}(\sigma)$. If e is an anchor event, then $e \in \xi_Z^\bullet$ and $\alpha_Z^\bullet(e) = \mathit{trans}^i_{A,q}(\sigma)$. Hence, $\mathit{recs}(\alpha_A(e)) \cup \mathit{fshs}(\alpha_A(e)) \subseteq \mathit{learn}_A(\alpha_Z^\bullet(e))$. If e is not an anchor event, let e' be its anchor event. Again, $e' \in \xi_Z^\bullet$ and $\alpha_Z^\bullet(e') = \mathit{trans}^i_{A,q}(\sigma)$, and, so, $\mathit{recs}(\alpha_A(e)) \cup \mathit{fshs}(\alpha_A(e)) \subseteq \mathit{learn}_A(\alpha_Z^\bullet(e'))$. The rest of the proof follows as in Lemma 5.18 using this result. □

Proposition 5.25 (Attack preservation – secrecy) *Let $\mu_{CZB} \in CZB$ and let ξ be one of its local states. Then an attack on $secr_S(\sigma)^\dagger$ at $\langle \mu_{CZB}, \xi \rangle$ implies an attack on $secr_S(\sigma)^\bullet$ at $\langle \beta^1(\mu_{CZB}), \xi^\bullet \rangle$.*

Proof: The proof of this result is similar to the proof of Proposition 5.21 using Lemma 5.23 and Lemma 5.24. □

We now focus of authentication properties, where the model translation will only consider the anchor events from completed steps. All incomplete steps will be ignored. In this case, $\beta^2(\mu_{CZB}) = \langle \lambda^\bullet, \alpha^\bullet, \pi^\bullet \rangle$ is as follows:

- $Ev_Z^\bullet = Ev_f^\bullet \cup Ev_c^\bullet$ and $\leq_Z^\bullet$ is the restriction of $\leq_Z$ to $Ev_Z^\bullet$;
- $\alpha_Z^\bullet$ and $\pi_Z^\bullet$ are as in β^1.

In the following, given a global state ξ of μ_{CZB} and $A \in Hon$, we denote by $\xi_A^\bullet$ the global state of $\beta^2(\mu_{CZB})$ such that $(\xi_A^\bullet)_Z = last(\xi_A) \downarrow$ (in $\beta^2(\mu_{CZB})$).

Lemma 5.26 *Let $A \in Hon$, $\mu_{CZB} \in CZB$, and ξ be a global state of μ_{CZB} such that $last(\xi_A) \in Ev_A \cap Ev_Z$ is an anchor event. Then, $(\xi_A^\bullet)_Z \subseteq \xi_Z$, and for every $B \in Hon \setminus \{A\}$, if $e \in (\xi_A^\bullet)_Z$ is such that $e \in Ev_B \cap Ev_Z$ is an anchor event, then $e \in \xi_B$.*

Proof: The condition on Z is straightforward by construction of $Ev_Z^\bullet$. Let $e_A = last(\xi_A)$. As both e and e_A are anchor events then $e, a_A \in Ev_Z^\bullet$. If $e \in (\xi_A^\bullet)_Z$, then it must be the case that $e \leq_Z^\bullet e_A$. Hence, $e \leq_Z e_A$ which implies that $e \in \xi_Z$ and, so, $e \in \xi_B$. □

Lemma 5.27 *Let $\mu_{CZB} \in CZB$ and let ξ be one of its local states. Then, for every $A \in Hon$ and $j \in \{1, \ldots, |step_{A,q}^i(\sigma)|\}$,*

$$if\ \beta^2(\mu_{CZB}), \xi_A^\bullet \Vdash @_Z[\mathsf{P}_\circ\ trans_{A,q}^i(\sigma)]\ then\ \mu_{CZB}, \xi \Vdash @_A[\mathsf{P}_\circ(step_{A,q}^i(\sigma))_j].$$

Proof: Let $\mu^\bullet = \beta^2(\mu_{CZB})$ and assume that $step_{A,q}^i(\sigma) = \langle act_1 \ldots act_k \rangle$. If $\mu^\bullet, \xi_A^\bullet \Vdash @_Z[\mathsf{P}_\circ\ trans_{A,q}^i(\sigma)]$ then there is an $e \in (\xi_A^\bullet)_Z$ such that $\alpha_Z^\bullet(e)) = trans_{A,q}^i(\sigma)$. Furthermore, $e \in Ev_A \cap Ev_Z$ and e is an anchor event, and so, $e \in \xi_A$. By the definition of anchor events, there are $e_1 \rightarrow_A \cdots \rightarrow_A e_k = e$ such that $\alpha_A(e_j) = act_j$. Hence, $e_j \in \xi_A$ and the result follows. □

Lemma 5.28 *Let $\mu_{CZB} \in CZB$ and let ξ be one of its local states. Then, for every $A \in Hon$,*

$$\beta^2(\mu_{CZB}), \xi_A^\bullet \Vdash @_Z[role_A^i(\sigma)^\bullet]\ iff\ \mu_{CZB}, \xi \Vdash @_A[role_A^i(\sigma)].$$

Proof: Let $\beta^2(\mu_{CZB}) = \mu^\bullet$ and $run_A^i(\sigma) = step_{A,0}^i(\sigma) \centerdot \cdots \centerdot step_{A,m}^i(\sigma) = \langle act_{0,1} \ldots act_{0,n_0} \cdot act_{1,1} \ldots act_{1,n_1} \ldots act_{m,1} \ldots act_{m,n_m} \rangle$. Assume that $\mu_{CZB}, \xi \Vdash @_A[role_A^i(\sigma)]$. Then, there are $e_{0,1} \rightarrow_A e_{0,n_0} \leq_A \cdots \leq_A e_{m,1} \rightarrow_A e_{m,n_m}$ such that $\alpha_A(e_{j,k}) = act_{j,k}$ with $e_{m,n_m} = last(\xi_A)$ and such that $e_{j,n_j} \in Ev_A \cap Ev_Z$ are anchor events. Then, by construction, $e_{0,n_0} \leq_Z^\bullet \cdots \leq_Z^\bullet e_{m,n_m}$ and $\alpha_Z^\bullet(e_{j,n_j}) = trans_{A,j}^i(\sigma)$. This implies that $\mu^\bullet, \xi_A^\bullet \Vdash @_Z[role_A^i(\sigma)^\bullet]$, since $last((\xi_A^\bullet)_Z) = e_{m,n_m}$. The proof of the converse is similar. □

Proposition 5.29 (Attack preservation – authentication) *Let $\mu_{CZB} \in CZB$ and let ξ be one of its local states. Then an attack on $auth_{A,B}^{i,j,q}(\sigma)^\dagger$ at $\langle \mu_{CZB}, \xi \rangle$ implies an attack on $auth_{A,B}^{i,j,q}(\sigma)^\bullet$ at $\langle \beta^2(\mu_{CZB}), \xi_A^\bullet \rangle$.*

Proof: The proof of this result is similar to the proof of Proposition 5.21 using Lemma 5.28 and Lemma 5.27. □

From Proposition 5.21, Proposition 5.25, and Proposition 5.29, we have the following corollary.

Corollary 5.30 *The models CZB and ZC are attack equivalent.*

6 Related work and conclusions

Communication, distribution, and cryptography are the essential ingredients of security protocols. Many attacks on security protocols arise from problems in communication and distribution, rather than cryptography itself. Hence, we follow the approach often taken in the formal methods community of abstracting away cryptographic details by assuming perfect black-box cryptography. The remaining ingredients — communication and distribution — are precisely the central concepts underlying DTL, which suggests DTL's suitability for this domain. DTL is neutral with respect to the kinds of interpretation structures it formalizes and, as we showed through our case studies, by choosing different signatures and axioms, we can define theories that are well suited for formalizing and reasoning about different application domains and problems in security protocol analysis.

DTL is closely related to the family of temporal logics whose semantics are based on the models of true concurrency, introduced and developed in [40, 41, 57]. DTL was proposed in [34] as a logic for specifying and reasoning about distributed information systems and several versions were given, reflecting different perspectives on how non-local information can be accessed by each agent [8, 16, 17]. In this paper, we use the simplest and most expressive formulation, from [8]. We stick with a propositional language, which suits our purposes, as we describe our schema axioms by taking advantage of explicit meta-level quantifications and of the inductive definition of closed sets of messages.

Of course, there are other formalisms for modeling distributed, communicating systems. A key difference is that DTL provides not just a modeling language (as process algebras also do, for example) but also a logic for reasoning about systems. Reasoning about local temporal properties of distributed agents could also be performed in a linear temporal logic over linearizations of the distributed models. However, this would come at the price of readability and simplicity. In contrast, DTL is simple and robust in the sense that formulas are invariant with respect to different linearizations. We have taken advantage of this in the proofs given in this paper.

With respect to formalizing and reasoning about security protocols and associated models, a large number of logics, formalisms, and tools have been proposed in recent years, e.g. [3, 5, 10, 12, 22, 30, 62, 63, 38, 42, 46, 48, 52, 55, 59, 60, 64, 65]. We will not compare with those logics that have been proposed solely as an object logic for protocol verification, e.g. [13, 15, 21, 38, 56, 58]. Rather, we compare with formalisms and logics that can be used to establish metatheoretic properties of protocol models or even relate protocol models.

The semantics of DTL is based on event structures. The most closely related formalism from the security community is *strand spaces* [36, 60, 63], which is widely used to analyze properties and models of security protocols. In [17], we formally investigated the relationship between the interpretation structures of our DTL network models and strand spaces. This comparison yields, transitively, a comparison with the other approaches that have been related to strand spaces, e.g. [20, 28, 29, 37]. Our results show that DTL network models and strand-space models are compatible, although they offer different views of protocol executions. We defined property-preserving, back-and-forth translations between models in our logic and strand-space models. This is nontrivial as, despite the similarities between the two formalisms (for example, both are based on partially-ordered sets of events with labeling information), there are substantial differences. These differences concern the way the principals and the intruder executing a protocol are represented, the way communication is formalized, and the locality of information. While carrying out semantic reasoning directly in terms of our interpretation structures is not that different than reasoning about bundles in the strand space approach, a fundamental difference is that strand spaces do not provide a logic. Hence, strand spaces lack a means for specifying classes of models axiomatically, a deductive system, a property specification language, and the ability to relate models based on the properties they preserve, possibly under formula translations.

The idea of using DTL to investigate general metatheoretic properties of security protocol models and model simplification techniques was first explored in [16]. In this preliminary work, we used DTL to formalize and establish the correctness of two model-simplification techniques. We first proved that *one intruder is enough*, namely that it is sufficient to consider one Dolev-Yao intruder instead of multiple intruders. Second, we proved the correctness of a *predatory intruder*,

which is an intruder with restricted behavior, e.g. who only sends messages that are immediately received and processed by honest agents. Lemma 4.1 from the current paper generalizes not only a similar result from [16] but also the protocol-independent secrecy results of [27, 47], which capitalize on the notion of honest ideals on strand spaces introduced in [62].

The step-compression technique that we considered in Section 5.2 is used in several other approaches, such as [3, 10, 52]. We have begun applying our logic to validate other techniques that can be used to improve the performance of analysis tools for security protocols, such as the partial-order techniques developed for the OFMC model checker [9, 53].

Numerous formal models have been proposed for reasoning about communication channels in security protocols and services, e.g. [1, 4, 14, 31, 32, 45, 51], For example, [51] gives two different abstract models for channels that are authentic, confidential, or secure: one that represents the ideal functionality of the channel and a second that employs concrete cryptographic messages to realize the channel properties. These two models are then shown equivalent under suitable assumptions. Our metareasoning results are in this spirit, in particular the proof of equivalence of the two models for guaranteeing message-origin authentication. It will be interesting to see how far we can take our approach in this regard. That is, whether and how our approach can be applied not only to obtain other such simulation-based results, but also to reason about protocol compositionality as done, for instance, in [6, 11, 19] at the cryptographic level and in [51] as well as in [1, 2, 14, 25, 31, 32, 35, 56] at a symbolic, black-box cryptography level. The combination of results, techniques, and tools from these and related approaches will play an important role in consolidating research in security protocol analysis.

We close with a final word on our use of DTL in this paper. Admittedly, given that our main example deals with the linearization of the channel model, we do not not require here the full power of the logic. Moreover, the security goals that we have considered, secrecy and authentication, constitute *safety* properties, and therefore we have mostly used formulas with simple past-time operators. Nevertheless, it should be clear that DTL can be used in many other, more complex, scenarios, for example, to model the presence of multiple non-collaborating intruders (possibly with different capabilities) or different channels with different properties (possibly controlled by different intruders). Furthermore, studying other interesting security goals, like forms of fairness in contract-signing protocols, would also require working with *liveness* properties, for which more complex temporal patterns would be necessary. Finally, note that we have avoided writing formulas with nested communications. This does not mean that such formulas are not necessary, as they result easily from our axioms (for instance, from the composition of axioms **(P3)**, **(C2)**, and **(C1)**). It is just a consequence of another nice property of DTL: in specifying each agent, locally, in a distributed system, it is possible to replace any DTL formula with nested communications with an equivalent finite set of formulas without nested communications [34]. The application of DTL to such more complex scenarios, in the context of security protocols and web services, is the subject of future work.

Acknowledgments

This work was partially supported by the FP7-ICT-2007-1 Project no. 216471, "AVANTSSAR: Automated Validation of Trust and Security of Service-oriented Architectures", by the PRIN'07 project "SOFT", and by FCT and EU FEDER via the KLog project PTDC/MAT/68723/2006 of SQIG-IT and the AMDSC UTAustin/MAT/0057/2008 project of IST. We thank Cas Cremers, Bruno Conchinha, Sebastian Mödersheim, and the anonymous referees for their valuable feedback on earlier drafts of this paper.

References

[1] M. Abadi, C. Fournet, and G. Gonthier. Secure Implementation of Channel Abstractions. *Information and Computation*, 174(1):37–83, 2002.

[2] S. Andova, C. Cremers, K. Gjøsteen, S. Mauw, S. Mjølsnes, and S. Radomirović. A framework for compositional verification of security protocols. *Information and Computation*, 206:425–459, 2008.

[3] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P.-C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In *Proceedings of CAV'2005*, LNCS 3576, pages 281–285. Springer-Verlag, 2005.

[4] A. Armando, R. Carbone, L. Compagna, J. Cuellar, and L. Tobarra Abad. Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps. In *Proceedings of FMSE 2008*. ACM Press, 2008.

[5] A. Armando and L. Compagna. SAT-based Model-Checking for Security Protocols Analysis. *International Journal of Information Security*, 6(1):3–32, 2007.

[6] M. Backes, B. Pfitzmann, and M. Waidner. The reactive simulatability framework for asynchronous systems. *Information and Computation*, 2007.

[7] D. Basin, C. Caleiro, J. Ramos, and L. Viganò. A Labeled Tableaux System for the Distributed Temporal Logic DTL. In *Proceedings of TIME 2008*, pages 101–109. IEEE CS Press, 2008.

[8] D. Basin, C. Caleiro, J. Ramos, and L. Viganò. Labeled Tableaux for Distributed Temporal Logic. *Journal of Logic and Computation*, 19(6):1245–1279, 2009.

[9] D. Basin, S. Mödersheim, and L. Viganò. Constraint Differentiation: A New Reduction Technique for Constraint-Based Analysis of Security Protocols. In *Proceedings of CCS'03*, pages 335–344. ACM Press, 2003.

[10] D. Basin, S. Mödersheim, and L. Viganò. OFMC: A symbolic model checker for security protocols. *International Journal of Information Security*, 4(3):181–208, 2005.

[11] M. Bellare, P. Rogaway, and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. In *Proceedings of STOC'98*. ACM Press, 1998.

[12] B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Proceedings of CSFW'01*, pages 82–96. IEEE CS Press, 2001.

[13] C. Boyd and A. Mathuria. *Protocols for Authentication and Key Establishment*. Springer-Verlag, 2003.

[14] M. Bugliesi and R. Focardi. Language based secure communication. In *Proceedings of CSF 21*, pages 3–16. IEEE CS Press, 2008.

[15] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8:18–36, 1990.

[16] C. Caleiro, L. Viganò, and D. Basin. Metareasoning about Security Protocols using Distributed Temporal Logic. In *Proceedings of ARSPA'04*, pages 67–89. ENTCS 125(1), 2005.

[17] C. Caleiro, L. Viganò, and D. Basin. Relating strand spaces and distributed temporal logic for security protocol analysis. *Logic Journal of the IGPL*, 13(6):637–664, 2005.

[18] C. Caleiro, L. Viganò, and D. Basin. On the Semantics of Alice&Bob Specifications of Security Protocols. *Theoretical Computer Science*, 367(1–2):88–122, 2006.

[19] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of FOCS'01*, pages 136–145. IEEE CS Press, 2001.

[20] I. Cervesato, N. A. Durgin, P. D. Lincoln, J. C. Mitchell, and A. Scedrov. A Comparison between Strand Spaces and Multiset Rewriting for Security Protocol Analysis. In *Proceedings of ISSS 2002*, LNCS 2609, pages 356–383. Springer-Verlag, 2003.

[21] I. Cervesato and P. F. Syverson. The logic of authentication protocols. In *Foundations of Security Analysis and Design*, LNCS 2171, pages 63–136. Springer-Verlag, 2001.

[22] Y. Chevalier and L. Vigneron. Automated Unbounded Verification of Security Protocols. In *Proceedings of CAV'02*, LNCS 2404, pages 324–337. Springer-Verlag, 2002.

[23] E. M. Clarke, S. Jha, and W. R. Marrero. Verifying security protocols with brutus. *ACM Trans. Softw. Eng. Methodol.*, 9(4):443–487, 2000.

[24] H. Comon-Lundh and V. Cortier. Security properties: two agents are sufficient. In *Proceedings of ESOP'2003*, LNCS 2618, pages 99–113. Springer-Verlag, 2003.

[25] V. Cortier and S. Delaune. Safely composing security protocols. *Formal Methods in System Design*, 34(1):1–36, 2009.

[26] V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 1:1–43, 2006.

[27] V. Cortier, J. Millen, and H. Rueß. Proving secrecy is easy enough. In *Proceedings of CSFW'01*. IEEE CS Press, 2001.

[28] F. Crazzolara and G. Winskel. Events in security protocols. In *Proceedings of CCS'01*, pages 96–105. ACM Press, 2001.

[29] F. Crazzolara and G. Winskel. Composing strand spaces. In *Proceedings of FST TCS 2002*, LNCS 2556, pages 97–108. Springer-Verlag, 2002.

[30] C. Cremers. The Scyther Tool: Verification, falsification, and analysis of security protocols. In *Proceedings of CAV'08*, LNCS 5123, pages 414–418. Springer-Verlag, 2008.

[31] C. Dilloway. Chaining secure channels. In *Proceedings of FCS-ARSPA-WITS'08*, 2008.

[32] C. Dilloway and G. Lowe. On the specification of secure channels. In *Proceedings of WITS '07*, 2007.

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

[34] H.-D. Ehrich and C. Caleiro. Specifying communication in distributed information systems. *Acta Informatica*, 36:591–616, 2000.

[35] J. D. Guttman. Cryptographic protocol composition via the authentication tests. In *Proceedings of FOSSACS'09*, LNCS 5504, pages 303–317. Springer-Verlag, 2009.

[36] J. D. Guttman and F. J. Thayer Fábrega. Authentication tests and the structure of bundles. *Theoretical Computer Science*, 283(2):333–380, 2002.

[37] J. Y. Halpern and R. Pucella. On the relationship between strand spaces and multi-agent systems. *ACM Transactions on Information and System Security*, 6(1):43–70, 2003.

[38] B. Jacobs and I. Hasuo. Semantics and logic for security protocols. *Journal of Computer Security*, 17(6):909–944, 2009.

[39] F. Jacquemard, M. Rusinowitch, and L. Vigneron. Compiling and Verifying Security Protocols. In *Proceedings of LPAR 2000*, LNCS 1955, pages 131–160. Springer, 2000.

[40] K. Lodaya, R. Ramanujam, and P. Thiagarajan. Temporal logics for communicating sequential agents: I. *Intern. Journal of Foundations of Computer Science*, 3(1):117–159, 1992.

[41] K. Lodaya and P. Thiagarajan. A modal logic for a subclass of event structures. In *Proceedings of ICALP 14*, LNCS 267, pages 290–303. Springer-Verlag, 1987.

[42] G. Lowe. Breaking and Fixing the Needham-Shroeder Public-Key Protocol Using FDR. In *Proceedings of TACAS'96*, LNCS 1055, pages 147–166. Springer-Verlag, 1996.

[43] G. Lowe. A hierarchy of authentication specifications. In *Proceedings of CSFW'97*. IEEE CS Press, 1997.

[44] G. Lowe. Casper: a Compiler for the Analysis of Security Protocols. *Journal of Computer Security*, 6(1):53–84, 1998.

[45] U. M. Maurer and P. E. Schmid. A calculus for security bootstrapping in distributed systems. *Journal of Computer Security*, 4(1):55–80, 1996.

[46] C. Meadows. The NRL Protocol Analyzer: An Overview. *Journal of Logic Programming*, 26(2):113–131, 1996.

[47] J. Millen and H. Rueß. Protocol-independent secrecy. In *2000 IEEE Symposium on Security and Privacy*. IEEE CS Press, May 2000.

[48] J. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *Proceedings of ACM Conference on Computer and Communications Security CCS'01*, pages 166–175, 2001.

[49] J. K. Millen and G. Denker. Capsl and mucapsl. *Journal of Telecommunications and Information Technology*, 4:16–27, 2002.

[50] S. Mödersheim. Algebraic Properties in Alice and Bob Notation. In *Proceedings of Ares'09*, pages 433–440. IEEE Xplore, 2009.

[51] S. Mödersheim and L. Viganò. Secure Pseudonymous Channels. In *Proceedings of Esorics'09*, LNCS 5789, pages 337–354. Springer-Verlag, 2009.

[52] S. Mödersheim and L. Viganò. The Open-Source Fixed-Point Model Checker for Symbolic Analysis of Security Protocols. In *FOSAD 2008/2009*, LNCS 5705, pages 166–194. Springer-Verlag, 2009.

[53] S. Mödersheim, L. Viganò, and D. Basin. Constraint Differentiation: Search-Space Reduction for the Constraint-Based Analysis of Security Protocols. *Journal of Computer Security*, 18(4):575–618, 2010.

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

[55] L. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6:85–128, 1998.

[56] Protocol Composition Logic (PCL). `http://crypto.stanford.edu/protocols/`.

[57] R. Ramanujam. Locally linear time temporal logic. In *Proceedings of LICS 11*, pages 118–127. IEEE CS Press, 1996.

[58] R. Ramanujam and S. Suresh. A (restricted) quantifier elimination for security protocols. *Theoretical Computer Science*, 367:228–256, 2006.

[59] P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, and B. Roscoe. *Modelling and Analysis of Security Protocols*. Addison Wesley, 2000.

[60] D. Song, S. Berezin, and A. Perrig. Athena: a novel approach to efficient automatic security protocol analysis. *Journal of Computer Security*, 9:47–74, 2001.

[61] D. R. Stinson. *Cryptography Theory and Practice (Third Edition)*. CRC Press, Inc., 2005.

[62] F. J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. Honest ideals on strand spaces. In *Proceedings of CSFW 17*, pages 66–78. IEEE CS Press, 1998.

[63] F. J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. Strand spaces: Proving security protocols correct. *Journal of Computer Security*, 7:191–230, 1999.

[64] M. Turuani. The CL-Atse Protocol Analyser. In *Proceedings of RTA'06*, LNCS 4098, pages 277–286, 2006.

[65] L. Viganò. Automated Security Protocol Analysis with the AVISPA Tool. *ENTCS 155*, 155:61–86, 2006.

[66] G. Winskel. Event structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency*, LNCS 255, pages 325–392. Springer-Verlag, 1987.

[67] G. Winskel and M. Nielsen. Models for concurrency. In *Handbook of Logic in Computer Science (Vol. 4): Semantic Modelling*, pages 1–148. Oxford University Press, Oxford, UK, 1995.