# Contract Signing, Optimism, and Advantage *

Rohit Chadha[1,4], John C. Mitchell[2], Andre Scedrov[1], and Vitaly Shmatikov[3]

[1] University of Pennsylvania
[2] Stanford University
[3] SRI International
[4] University of Sussex

**Abstract.** A contract signing protocol lets two parties exchange digital signatures on a pre-agreed text. *Optimistic* contract signing protocols enable the signers to do so without invoking a trusted third party. However, an adjudicating third party remains available should one or both signers seek timely resolution. We analyze optimistic contract signing protocols using a game-theoretic approach and prove a fundamental impossibility result: in any fair, optimistic, timely protocol, an optimistic player yields an advantage to the opponent. The proof relies on a careful characterization of optimistic play that postpones communication to the third party. Since advantage cannot be completely eliminated from optimistic protocols, we argue that the strongest property attainable is the absence of *provable* advantage, *i.e.*, abuse-freeness in the sense of Garay-Jakobsson-MacKenzie.

## 1 Introduction

A variety of contract signing protocols have been proposed in the literature, including gradual-release two-party protocols [5, 7, 12] and fixed-round protocols that rely on an adjudicating "trusted third party" [2, 3, 18, 23, 26]. In this paper, we focus on fixed-round protocols that use a trusted third party optimistically, meaning that when all goes well, the third party is not needed. The reason for designing optimistic protocols is that if a protocol is widely or frequently used by many pairs of signers, the third party may become a performance bottleneck. Depending on the context, seeking resolution through the third party may delay termination, incur financial costs, or raise privacy concerns. Obviously, the value of an optimistic protocol, as opposed to one that requires a third party signature on every transaction, lies in the frequency with which "optimistic" signers can complete the protocol without using the third party.

Some useful properties of contract signing protocols are *fairness*, which means that either both parties get a signed contract, or neither does, and *timeliness*, which generally means that each party has some recourse to avoid unbounded waiting. The reason for using a trusted third party in fixed-round protocols is a basic limitation [14, 24] related to the well-known impossibility of distributed consensus in the presence of faults [17]:

* The authors are partially supported by OSD/ONR CIP/SW URI "Software Quality and Infrastructure Protection for Diffuse Computing" as ONR Grant N00014-01-1-0795. Additional support for Mitchell from NSF Grant CCR-0121403, ITR/SY "Computational Logic Tools for Research and Education," for Scedrov from NSF Grant CCR-0098096, and for Shmatikov from ONR under Grants N00014-02-1-0109 and N00014-01-1-0837.

no fixed-length two-party protocol can be fair. Although there is a trivial protocol with a trusted third party, in which both signers always send their signatures directly to it, protocols that are fair, timely, and usefully minimize demands on the third party have proven subtle to design and verify.

This paper refines previous models, formalizes properties from the literature on fixed-round two-party contract signing protocols, and establishes relationships between them. We use the set-of-traces semantics for protocols, defining each instance of the protocol as the set of all possible execution traces arranged in a tree. Our chosen notation is multiset rewriting [10], but the results would hold for other formalisms with the same basic execution model.

*Model for optimism.* One modeling innovation is an *untimed* nondeterministic setting that provides a set-of-traces semantics for optimism. Intuitively, optimistic behavior in contract signing is easily described as a temporal concept: an optimistic signer is one who waits for some period of time before contacting the trusted third party. If Alice is optimistic, and Bob chooses to continue the protocol by responding, then Alice waits for Bob's message rather than contact the third party. Since the value of an optimistic protocol lies in what it offers to an optimistic player, we evaluate protocols subject to the assumption that one of the players follows an optimistic strategy. As a direct way of mathematically characterizing optimistic play, we allow an optimistic player to give his opponent the chance to signal (out of band) whether to wait for a message. This gives us a relatively easy way to define the set of traces associated with an optimistic signer, while staying within the traditional nondeterministic, untimed setting.

*Impossibility result.* In evaluating protocol performance for optimistic players, we prove that in every fair, timely protocol, an optimistic player suffers a disadvantage against a strategic adversary. The importance of this result is that optimistic protocols are only useful to the extent that signers may complete the protocol optimistically without contacting the third party. In basic terms, our theorem shows that to whatever degree a protocol allows signers to avoid the third party, the protocol proportionally gives one signer unilateral control over the outcome of the protocol.

To illustrate by example, consider an online stock trading protocol with signed contracts for each trade. Suppose the broker starts the protocol, sending her commitment to sell stock to the buyer at a specific price, and the buyer responds with his commitment. To ensure timely termination, the broker also enjoys the ability to abort the exchange by contacting the trusted third party (TTP) if the buyer has not responded. Once the buyer commits to the purchase, he cannot use the committed funds for other purposes. Even if he has the option to contact the TTP immediately, an optimistic buyer will wait for some period of time for the broker to respond, hoping to resolve the transaction amicably and avoid the extra cost or potential delay associated with contacting the TTP. This waiting period may give the broker a useful window of opportunity. Once she has the buyer's commitment, the broker can wait to see if shares are available from a selling customer at a matching or lower price. The longer the buyer is inclined to wait, the greater chance the broker has to pair trades at a profit. If the broker finds the contract unprofitable, she can abort the transaction by falsely claiming to the TTP that the buyer has not responded. This broker strategy succeeds in proportion to the time that the buyer optimistically waits for the broker to continue the protocol; this time interval, if known

exactly or approximately, gives the broker a period where she can decide *unilaterally* whether to abort or complete the exchange.

*Abuse-freeness.* Since advantage against an optimistic player cannot be eliminated, the most a protocol can do is prevent the opponent from *proving* that he has an advantage. For example, even though the broker in our example has control over deciding whether the sale happens, the protocol may still be able to prevent her from showing the buyer's commitment to other parties. Such protocols have been called *abuse-free* in the literature [18]. We use a formal representation of knowledge derived from epistemic logic [19, 16] to formalize the "ability to prove" and analyze abuse-freeness as the lack of *provable advantage*.

The paper is organized as follows. In section 2, we briefly summarize our semantic framework and define the class of two-party contract signing protocols with trusted third party. In section 3, we formalize protocol properties such as fairness, optimism, and timeliness. In section 4, we formalize optimistic behavior of a participant, and show that the optimistic participant is at a disadvantage in any fair, optimistic, timely protocol. In section 5, we formalize provable advantage and abuse-freeness. Related work is discussed in section 6. We summarize our results in section 7.

*Acknowledgments.* We are particularly grateful to D. Malkhi for pointing out the vulnerability of optimistic players in fair exchange. We also thank I. Cervesato, S. Even, D. Gollmann, S. Kremer, J.F. Raskin, C. Meadows, and J. Millen for interesting and helpful discussions.

## 2 Model

### 2.1 Multiset rewriting formalism

Our protocol formalism is multiset rewriting with existential quantification, MSR [10], which can be seen as an extension of some standard models of computation, *e.g.*, multiset transformation [4] and chemical abstract machine [6]. This formalism faithfully expresses the underlying assumptions of the untimed, nondeterministic, asynchronous model. A protocol definition in MSR defines the *set of all possible execution traces* for any instance of the protocol. A number of other formalisms that do so, such as [1, 15] and others, would have suited our purposes as well, and in this sense our main results are independent of the MSR formalism. The synchronous model with a global clock does not seem appropriate for our investigation because fixed-round contract signing protocols in the literature [2, 3, 18, 23, 26] do not rely on a global clock.

MSR syntax involves terms, facts, and rules. To specify a protocol, first choose a vocabulary, or *first-order signature*. We assume that our vocabulary contains some basic sorts such as $public\_key$ for public keys and $mssg$ for protocol messages. As usual, the *terms* over a signature are the well-formed expressions produced by applying functions to arguments of the correct sort. A *fact* is a first-order atomic formula over the chosen signature, without free variables. Therefore, a fact is the result of applying a predicate symbol to ground terms of the correct sort. A *state* is a finite multiset of facts.

A state transition is a *rule* written using two multisets of first-order atomic formulas and existential quantification, in the syntactic form $F_1, \ldots, F_k \longrightarrow \exists x_1 \ldots \exists x_j . G_1, \ldots$

$G_n$. The meaning of this rule is that if some state $S$ contains facts obtained by a ground substitution $\sigma$ from first-order atomic formulas $F_1, \ldots, F_k$, then one possible next state is the state $S^*$ that is similar to $S$, but with facts obtained by $\sigma$ from $F_1, \ldots, F_k$ removed and facts obtained by $\sigma$ from $G_1, \ldots, G_n$ added, where $x_1, \ldots, x_j$ are replaced by new symbols. If there are free variables in the rule $F_1, \ldots, F_k \longrightarrow \exists x_1 \ldots \exists x_j . G_1, \ldots G_n$, these are treated as universally quantified throughout the rule. In an application of a rule, these variables may be replaced by any ground terms.

As an example, consider state $\{P(f(a)), P(b)\}$ and rule $P(x) \longrightarrow \exists z . Q(f(x), z)$. The next state is obtained by instantiating this rule to $P(f(a)) \longrightarrow \exists z . Q(f(f(a)), z)$. Applying the rule, we choose a new value $c$ for $z$ and replace $P(f(a))$ by $Q(f(f(a)), c)$, obtaining the new state $\{Q(f(f(a)), c), P(b)\}$.

*Timer signals.* In our model, timers are interpreted as *local* signals, used by participants to decide when to quit waiting for a message from the other party in the protocol. They do not refer to any global time or imply synchronicity. Timers are formalized by binary *timer predicates*, whose first argument is of the sort $public\_key$ and identifies the participant who receives its signal, while the second argument is one of the following three constants of the sort $timer\_state$: $unset$, $set$, and $timed\_out$.

*Cryptography.* Contract signing protocols usually employ cryptographic primitives. In general, the purpose of cryptography is to provide messages that are meaningful to some parties, but not subject to arbitrary (non-polynomial-time) computation by others. For example, encryption provides messages that are meaningful to any recipient with the decryption key, but not subject to decryption by any agent who does not possess the decryption key. The logic-based formalism of MSR cannot capture subtle distinctions between, for example, functions computable with high probability and functions computable with low or negligible probability. Instead, we must model functions as either feasibly computable, or not feasibly computable.

For each operation used in a protocol, we assume there is some MSR characterization of its computability properties. To give a concrete framework for presenting these rules, let us assume some set of predicates $\mathcal{P} = \{P^\alpha | \alpha \text{ is any sort}\}$. Since the sort $\alpha$ is determined by the sort of the arguments to $P^\alpha$, we will not write the sort when it is either irrelevant, or clear from context. Intuitively, a rule of the form

$$P(s_1), \ldots, P(s_m), F_1, \ldots, F_j \longrightarrow P(t_1), \ldots, P(t_n), F_1, \ldots, F_j$$

means that if an agent possesses data $s_1, \ldots, s_m$, then under conditions specified by facts $F_1, \ldots, F_j$, it is computationally feasible for him to also learn $t_1, \ldots, t_n$. For example, here are the familiar "Dolev-Yao" [13, 25] rules given in [10]:

$$P(x), P(k) \longrightarrow P(\text{encrypt}(k, x))$$
$$P(\text{encrypt}(k, x)), P(k^{-1}), \text{Keypair}(k, k^{-1}) \longrightarrow P(x)$$

In the remainder of the paper, we assume some fixed theory **Possess** of rules that characterize the computationally feasible operations on messages. As a disclaimer, we emphasize that the results in this paper are accurate statements about a protocol using cryptographic primitives only to the extent that **Possess** accurately characterizes the computationally feasible operations. In particular, protocols that distinguish between

low-order polynomial computation and high-order polynomial computation, or rely on probabilistic operations in some essential way, may fall outside the scope of our analysis and may conceivably violate some of our results.

### 2.2 Protocol model

We say that a protocol $P$ is a *contract signing protocol* if it involves three parties, $O$ (originator), $R$ (responder), and $T$ (trusted third party), and the goal of the protocol is to enable $O$ (respectively, $R$) to obtain $R$'s signature (respectively, $O$'s signature) on some pre-agreed text. For brevity, we will say *signature* as a shorthand for "signature on the pre-agreed text," use terms *contract signing* and *signature exchange* interchangeably, and refer to $O$ and $R$ as *signers*. We specify the protocol by a set of MSR rules, which we call a *theory*. Any sequence of rules consistent with the theory corresponds to a valid execution trace of a protocol instance. If execution traces are naturally arranged in trees, then the MSR theory defines the set of all possible execution traces as a forest of trees. To obtain the impossibility result, we choose *any* contract signing protocol $P$ and fix it. We assume that the contract text for each instance contains a unique identifier, and consider only a single instance of $P$.

A protocol theory $\mathbf{P}$ for the given protocol is the disjoint union of six theories: $\mathbf{O}, \mathbf{R}, \mathbf{T_0}, \mathbf{O_{timeouts}}, \mathbf{R_{timeouts}}$, and $\mathbf{T_{timeouts}}$. We will refer to $\mathbf{O}, \mathbf{R}, \mathbf{T_0}$ as *role theories*. Each role theory specifies one of the protocol roles by giving a finite list of *role state predicates* that define the internal states of the participant playing that role and the rules for advancing from state to state. Role theory also contains another, disjoint list of *timer predicates* describing the rules for the participant's timers. A participant may advance his state by "looking" at the state of his timers or the network (*i.e.*, a timer or a network predicate appears on the left side of the rule). He may also set his timer by changing the timer's state from $unset$ to $set$, but he may not change it to $timed\_out$.

A *timeout rule* is a rule of the form $Z(k, set) \rightarrow Z(k, timed\_out)$, where $k$ is the public key of the participant associated with timer $Z$. In the protocol theory, $\mathbf{O_{timeouts}}$, $\mathbf{R_{timeouts}}$, and $\mathbf{T_{timeouts}}$ are the sets of *timeout rules* for all timers of $O$, $R$, and $T$, respectively. For simplicity, we will combine the role theory and the timeouts of $T$, and call it $\mathbf{T} = \mathbf{T_0} \cup \mathbf{T_{timeouts}}$.

*Communication.* Following the standard assumption that the adversary controls the network and records all messages, we model communication between $O$ and $R$ by a unary *network predicate* $N$ whose argument is of the sort $mssg$. Once a fact $N(m)$ for some $m$ is added to the state, it is never removed. As in contract signing protocols in the literature [3, 18], we assume that channels between signers and $T$ are inaccessible to the adversary and separate from the network between $O$ and $R$ (by contrast, [20] considers security of contract signing protocols under relaxed assumptions about channel security). Channels between signers and $T$ are modeled by ternary $TTPchannel$ *predicates*, whose arguments are of the sort $public\_key$, $public\_key$ and $mssg$, respectively. For example, $tc_o(k_o, k_t, m)$ models the channel between $O$ and $T$ carrying message $m$.

*Threat model.* We are interested in guarantees provided by contract signing protocols when one of the signers misbehaves in an arbitrary way. $T$ is assumed to be honest. We will call the misbehaving signer the *adversary*. The adversary does not necessarily

follow the protocol, and may ignore the state of the timers or stop prematurely. He may gather messages from the network, store them, decompose them into fragments and construct new messages from the fragments. These abilities are formalized by theories $\mathbf{O_{threat}}$ and $\mathbf{R_{threat}}$ containing *dishonest rules* for $O$ and $R$, respectively. Each rule models a particular dishonest operation.

A *protocol definition* consists of the protocol theory $\mathbf{P}$, theories $\mathbf{O_{threat}}$ and $\mathbf{R_{threat}}$, **Possess** theory which models computationally feasible operations on messages, and the *initial set of facts* $S_0$ which contains the initial states of all participants and timers. Formal definition of protocol theory can be found in appendix A. Non-probabilistic fixed-round contract signing protocols in the literature such as [3, 18] can all be defined in this way.

### 2.3 Traces and continuation trees

A *state* is a finite multiset of facts. For example, the initial state $S_0$ may include facts $O_0(k_o, k_o^{-1}, k_r, p)$ and $R_0(k_r, k_r^{-1}, k_o, p)$ modeling the initial states of the originator and the responder in protocol $p$: each knows his own public and private keys, and the opponent's public key. A *trace* from state $S$ is a chain of nodes, with the root labeled by $S$, each node labeled by a state, and each edge labeled by a triple $\langle t, \sigma, \mathbf{Q}\rangle$. Here $Q$ is one of $\{\mathbf{O}, \mathbf{R}, \mathbf{T}, \mathbf{O_{timeouts}}, \mathbf{R_{timeouts}}, \mathbf{O_{threat}}, \mathbf{R_{threat}}\}$, $t \in \mathbf{Q}$ is a state transition rule, and $\sigma$ is a ground substitution. If $\langle t, \sigma, \mathbf{Q}\rangle$ labels the edge from a node labeled by $S_1$ to a node labeled by $S_2$, it must be the case that the application of $t$ to $S_1\sigma$ produces $S_2$. Any state labeling a node in this chain is said to be *reachable from S*. We will simply say that a state is *reachable* if it is reachable from the initial state $S_0$.

An edge is a *dishonest move of* $O$ if it is labeled by some $t \in \mathbf{O_{threat}}$. $O$ is said be *honest in the trace* if there are no dishonest moves of $O$ in the trace. If $S$ is reachable by a trace in which $O$ is honest, then $S$ is *reachable by honest* $O$. The definitions for $R$ are symmetric. Assuming that dishonest participants, if any, make only a finite number of dishonest moves, let *continuation tree* $ctr$ at state $S$ be the finite tree of all possible traces from $S$. This tree can be thought of as a game tree that represents the complete set of possible plays. Let $ctr_{[O]}$ be the tree obtained from $ctr$ by removing all edges in $\mathbf{O} \cup \mathbf{O_{threat}}$ along with their descendants. Intuitively, $ctr_{[O]}$ is the set of all possible plays if $O$ stops participating in the protocol. Definition of $ctr_{[R]}$ is symmetric. We will say that any edge $e$ in $ctr$ that is labeled by a rule in $\mathbf{O}$ or $\mathbf{O_{threat}}$ (respectively, $\mathbf{R}$ or $\mathbf{R_{threat}}$), is under *$O$'s control* (respectively, $R$'s control). To model optimism of honest signers (see section 4), we will also assume that some edges in $\mathbf{O_{timeouts}} \cup \mathbf{R_{timeouts}}$ are under control of the dishonest participant.

## 3 Properties of Contract Signing Protocols

MSR definition of the protocol defines the set of all possible execution traces in the form of a continuation tree. To define protocol properties such as fairness, optimism, timeliness, and advantage, we view the continuation tree as a game tree containing all possible plays, and adapt the notion of strategy from classical game theory.

For the remainder of the paper, we will assume that only one of the signers is honest. We will use $A$ to refer to the honest signer, *i.e.*, $A$ refers to either $O$, or $R$, depending on which of them is honest. We'll use $B$ to refer to the other, dishonest signer.

### 3.1 Strategies

Following [11], we formalize strategies as truncated continuation trees. Given a set of edges $E$, let $ctr \backslash E$ be the tree obtained from continuation tree $ctr$ by removing the edges in $E$ along with their descendants. Intuitively, if $E$ is a subset of edges of $ctr$ under $A$'s control, then $ctr \backslash E$ is the set of possible plays that result if $A$ does not use transitions in $E$. Similarly, we can define $ctr_{[A]} \backslash E$ (recall that $ctr_{[A]}$ is the tree of all plays if $A$ stops participating in the protocol).

**Definition 1.** *Let $S$ be a reachable state and let $ctr$ be the continuation tree from $S$. Let $X \subseteq \{A, B, T\}$.*
*1. If $E$ is a subset of edges of $ctr$ such that each edge in $E$ is under the control of some $p \in X$, then $ctr \backslash E$ is said to be a* strategy for the coalition $X$. *If there are no dishonest moves of any $p \in X$ in $ctr \backslash E$, then $ctr \backslash E$ is said to be an* honest strategy.
*2. If $E$ is a subset of edges of $ctr_{[A]}$ such that each edge in $E$ is under the control of some $p \in X$, then $ctr_{[A]} \backslash E$ is said to be an $A$-silent* strategy for the coalition $X$.

This definition corresponds to the standard game-theoretic notion of strategy. $E$ represents the plays that the coalition $X$ considers unfavorable, and $ctr \backslash E$ represents the continuations that $X$ prefers. At any given state $S'$ in $ctr \backslash E$, an edge coming out of the node labeled by $S'$ indicates the next move for $X$ in accordance with the strategy $ctr \backslash E$. If the edge is not under $X$'s control, then the next move for $X$ is idling, *i.e.*, waiting for others to move.

To define fairness and other properties, we are interested in strategies in which the coalition $X$ drives the protocol to a state in which some property holds:

**Definition 2.** *If there is a strategy $ctr \backslash E$ from $S$ for coalition $X$ such that all leaf nodes of $ctr \backslash E$ are labeled by states $S'$ that satisfy some property $\phi(S')$, then $X$* has a strategy from $S$ to reach a state in which $\phi$ holds.

The definition for $A$-silent strategies is similar.

Since the players' objective in the game is to obtain each other's signatures, we are interested in the states where $A$ possesses $B$'s signature and the ones where $B$ possesses $A$'s signature. Formally, $B$ *possesses* some term $u$ in a reachable state $S$ if $u$ is derivable, using the rules in **Possess**, from the terms in $B$'s internal role state predicate $B_i$ in $S$ and $B$'s additional memory in $S$ given to him by the threat model. Possession is always monotonic. The definition for $A$ is symmetric, except that the threat model does not have to be considered.

**Definition 3.** *If there is a strategy for coalition $X$ such that all leaf nodes in the strategy are labeled by states in which $A$ possesses $B$'s signature, then $X$* has a strategy from $S$ to give $A$ $B$'s signature. *Moreover, if $X = \{A\}$, then $A$ is said to have a* strategy to obtain $B$'s signature.

### 3.2 Fairness, optimism, timeliness, and advantage

We now use the notion of strategy to define what it means for a contract signing protocol to be fair, optimistic, and timely, and what it means for a participant to enjoy an advantage. The definitions are quite subtle. For example, we need to draw the distinction between a *strategy* for achieving some outcome, and a *possibility* that the outcome will happen under the right circumstances. This requires introduction of a four-valued variable to characterize the degree of each player's control over the protocol game.

*Fairness.* Fairness is the basic symmetry property of an exchange protocol. There is a known impossibility result [14, 24] demonstrating that no deterministic two-party protocol can be fair. Therefore, fairness requires introduction of at least one other party, *e.g.,* the trusted third party $T$. Our definition is equivalent to a common definition of fairness in terms of state reachability [18, 11]. Intuitively, a protocol is fair for the honest signer $A$, if, whenever $B$ has obtained $A$'s signature, $A$ has a strategy in coalition with $T$ to obtain $B$'s signature.

**Definition 4.** *A protocol is* fair *for honest $A$ if, for each state $S$ reachable by honest $A$ such that $B$ possesses $A$'s signature in $S$, the coalition of $A$ and $T$ has an honest strategy from $S$ to give $A$ $B$'s signature for all bounds on the number of moves that a dishonest $B$ makes.*

*Advantage.* Intuitively, fairness says that either both players obtain what they want, or neither does. This is not always sufficient, however. A player's ability to decide unilaterally *whether* the transaction happens or not can be of great value in scenarios where resource commitment is important, such as online trading and auction bidding.

To characterize the degree to which each participant controls the outcome of the protocol in a given state, we now define a pair of values $rslv_A, rslv_B$ associated with each reachable state. We are interested in what a participant *may* do in the worse possible case. Therefore, despite our assumption that $A$ is honest, we will consider $A$'s dishonest moves when reasoning about $A$'s ability to control the outcome.

**Definition 5.** *Define $rslv_A$ for any reachable state $S$ as follows:*

$rslv_A(S) = 2$, *if $A$ has a strategy to obtain $B$'s signature for all bounds on the number of dishonest moves of $B$,*

$= 1$, *if $rslv_A(S) \neq 2$, but $A$ has a $B$-silent strategy to reach state $S'$ such that $rslv_A(S') = 2$,*

$= \frac{1}{2}$, *if $rslv_A(S) \neq \{1, 2\}$, but there is state $S'$ reachable from $S$ such that $rslv_A(S') = 2$, and no transition on the $S \rightarrow S'$ path is in $\mathbf{B} \cup \mathbf{B_{threat}}$,*

$= 0$, *otherwise.*

*The strategies need not be honest. Definition of $rslv_B$ is symmetric.*

Intuitively, $rslv_B(S) = 2$ if $B$ can obtain $A$'s signature no matter what $A$ does, 1 if $B$ can obtain $A$'s signature provided $A$ stops communicating and remains silent, $\frac{1}{2}$ if there is a possibility (but no strategy) for $B$ to obtain $A$'s signature when $A$ is silent, and 0 means that $B$ cannot obtain $A$'s signature without $A$'s involvement. The difference between 1 and $\frac{1}{2}$ is essential. For example, $rslv_B(S) = 1$ if $B$ can obtain $A$'s signature

by sending a message to $T$ as long as $A$ is silent, while $rslv_B(S) = \frac{1}{2}$ if $A$ is silent, but some previously sent message is already on the channel to $T$, and the outcome of the protocol depends on the race condition between this message and $B$'s message.

Given an initial state $S_0$, we assume that $rslv_A(S_0) = rslv_B(S_0) = 0$. The signature exchange problem is not meaningful otherwise.

**Definition 6.** *$B$ has an* abort strategy *in $S$ if $B$ has a strategy to reach a state $S'$ such that $rslv_A(S') = 0$. $B$ has a* resolve strategy *in $S$ if $B$ has a strategy to reach a state $S''$ such that $rslv_B(S'') = 2$. $B$ has an* advantage *in $S$ if $B$ has both an abort strategy and a resolve strategy.*

If $B$ has an advantage in $S$, then $A$ does not have an advantage in $S$, and vice versa.

*Optimism.* Intuitively, a protocol is optimistic if it enables two honest parties to exchange signatures without involving the trusted third party, assuming they do not time out waiting for each other's messages. Such protocols potentially provide a practical means of fair exchange between mistrusting agents without relying on a third party in most instances.

We say that $A$ *does not send a message to* $T$ in the transition from $S$ to $S'$ if (i) the transition is an application of a rule in $\mathbf{A} \cup \mathbf{A_{threat}}$, and (ii) no fact created by the transition matches a term in the left hand side of a rule in $\mathbf{T}$.

**Definition 7.** *A fair protocol is* optimistic *for $B$ if, assuming $A$ is honest and $B$ controls the timeouts of both $A$ and $B$, $B$ has an honest strategy at $S_0$ such that*
*1) no messages are sent by any signer to $T$;*
*2) every leaf node is labeled by a state in which $B$ possesses $A$'s signature;*
*3) there is a trace from $S_0$ to a leaf node that involves only the transitions in $\mathbf{A} \cup \mathbf{B}$.*

*Any trace in this strategy is an* optimistic trace. *Definition of* optimistic for $A$ *is symmetric. A protocol is optimistic if it is optimistic for both signers.*

Our definition of optimism implies that the protocol specification does not permit honest participants to contact $T$ nondeterministically, *i.e.*, every rule that results in a message sent to $T$ is conditional on some timer timing out.

*Timeliness.* We now formalize the following intuition [3]: "one player cannot force the other to wait for any length of time — a fair and timely termination can always be forced by contacting the third party." Timeliness has been emphasized by the designers of fair exchange protocols, since it is essential for practical use. In any state of the protocol, each participant should be able to terminate the exchange unilaterally. If he has not been able to obtain the other's signature, he can always reach a terminal state where he can stop and be sure that the opponent will not be able to obtain his signature, either.

**Definition 8.** *A fair, optimistic protocol is* timely *for $B$ if in every state on an optimistic trace $B$ has an $A$-silent strategy to reach a state $S'$ such that $rslv_A(S') = 0$ or $rslv_B(S') = 2$. A protocol is timely if it is timely for both signers.*

To illustrate the importance of timeliness, consider a protocol that is *not* timely, *e.g.*, Boyd-Foo protocol [8]. In this protocol, originator $O$ releases some information that can be used by responder $R$ to obtain $O$'s signature from $T$ at some later point. If $R$ stops

communicating, $O$ is at his mercy. He may have to wait, possibly forever, before he learns whether the exchange has been successful.

For the rest of this paper, we assume that the protocol is fair, timely, and optimistic for both signers.

## 4 Impossibility of Balance in Optimistic Protocols

As explained in the introduction, optimistic contract signing protocols are only valuable insofar as they offer benefit to an optimistic participant. We say that the honest participant $A$ is *optimistic* if, in any state where he is permitted by the protocol specification to contact trusted third party $T$, he waits for $B$'s response before contacting $T$.

The propensity of the optimistic participant to wait for the opponent's response before contacting $T$ can be exploited by the opponent. Recall that definition 7 implies that an honest participant only contacts $T$ after some timer times out. We use this to model optimism of $A$ by giving $B$ the ability to schedule the timeout rules of $A$ by an "out-of-band" signal. In any implementation of the protocol, $B$ does not actually schedule $A$'s timers. This is simply a technical device to restrict the set of execution traces under consideration to those that may occur when one of the participants is optimistic.

Definition 6 can thus be extended to cases where $A$ is optimistic by permitting $B$'s strategy to include control over timeouts of $A$ and $B$. If $B$ does not have a strategy for reaching a state where he has an advantage over an optimistic $A$, we say that the protocol is *balanced* for an optimistic $A$. As we will now show, balance cannot be achieved by any fair, timely, optimistic protocol.

The first observation underlying our proof is that, in the interleaving semantics of concurrency used by our model, the order of application of state transition rules that affect independent parts of the system can be commuted. The second observation is that the strategies available to the dishonest player are not negatively affected by messages sent to him by the honest player or by the honest player's timeouts because the dishonest player is free to ignore both.

We start with an auxiliary proposition, which follows directly from definition 5.

**Proposition 1.** *Let $S \to S'$ be a state transition* not *in $\mathbf{B} \cup \mathbf{B_{threat}}$. If $rslv_B(S) = 2$, then $rslv_B(S') = 2$. If $rslv_A(S) = 0$, then $rslv_A(S') = 0$.*

Proposition 1 implies that if $rslv_A(S) = 0$ and $rslv_A(S') > 0$, then the $S \to S'$ transition must be in $\mathbf{B} \cup \mathbf{B_{threat}}$. Similarly, if $rslv_B(S) = 0$ and $rslv_B(S') > 0$, then $S \to S'$ is in $\mathbf{A} \cup \mathbf{A_{threat}}$. Intuitively, a player acquires some degree of control over the outcome of the protocol for the first time only because of the other player's move.

Just like we defined $ctr_{[A]}$ to be the tree obtained from $ctr$ by removing all edges in $\mathbf{A} \cup \mathbf{A_{threat}}$, we define $ctr_{[A+]}$ to be the tree obtained from $ctr$ by removing all edges in $\mathbf{A} \cup \mathbf{A_{threat}} \cup \mathbf{A_{timeouts}}$. If $E$ is a selection of edges in $ctr_{[A+]}$ under $B$'s control, then $ctr_{[A+]} \backslash E$ is a strategy available to $B$ if $A$ remains silent *and* no timers time out. We will call such a strategy *weak $A$-silent strategy*.

**Proposition 2.** *Let $S \to S'$ be a state transition in $\mathbf{A_{timeouts}}$. $B$ has a weak $A$-silent abort [resolve] strategy at $S'$ if and only if $B$ has a weak $A$-silent abort [resolve] strategy at $S$.*

The proof of proposition 2 relies on the fact that the moves of $B$ and $T$ that constitute a weak $A$-silent strategy cannot depend on the state of $A$'s timers.

**Proposition 3.** *$B$ has an $A$-silent abort [resolve] strategy at $S$ if and only if $B$ has a weak $A$-silent abort [resolve] strategy at $S$.*

In the proof, we use proposition 2 to construct an $A$-silent strategy from a weak $A$-silent strategy by induction on the height of the continuation tree. Proposition 3 establishes that the strategies available to the dishonest player are not negatively affected by the honest player's timeouts. We now show that they are not affected by the honest player's messages to the dishonest player.

**Lemma 1.** *Let $S \rightarrow S'$ be a transition in $\mathbf{A} \cup \mathbf{A_{threat}}$. If $B$ has an $A$-silent abort [resolve] strategy in $S$, and $A$ does not send a message to $T$ in the $S \rightarrow S'$ transition, then $B$ has an $A$-silent abort [resolve] strategy in $S'$.*

*Proof.* The proof, illustrating our general proof techniques, is in appendix B.

We use lemma 1 to show that for each strategy conditional on $A$ remaining silent, there is an equivalent strategy in which $A$ is not silent, but $B$ simply ignores $A$'s messages. The strategy works as long as $A$ does not try to talk to $T$.

**Lemma 2.** *If $B$ has an $A$-silent abort [resolve] strategy at $S$, and $A$ does not send any messages to $T$, then $B$ has an abort [resolve] strategy.*

*Proof.* (Omitted for space reasons).

We now show that a strategy conditional on $A$ not talking to $T$ works against an optimistic $A$ since he waits for $B$'s messages instead of trying to contact $T$.

**Lemma 3.** *Let $S$ be a state that does not contain $Z(k, timed\_out)$ for any timer predicate $Z$. If $B$ has an $A$-silent abort [resolve] strategy in state $S$, then $B$ has an abort [resolve] strategy against optimistic $A$ in $S$.*

*Proof.* (Sketch) Definition 7 implies that an optimistic $A$ contacts $T$ only when some timer times out. $B$ controls the timeouts of an optimistic $A$. Hence $B$ can prevent $A$ from sending any message to $T$. We then apply lemma 2.

**Theorem 1 (Impossibility of Balance).** *Let $\mathbf{P}$ be a fair, optimistic, timely protocol between signers $A$ and $B$. If $A$ is optimistic, then there is a non-initial state $S^*$ such that $B$ has an advantage against an optimistic $A$ at $S^*$.*

*Proof.* (Sketch) By definition 7, there is an optimistic trace from the initial state $S_0$ which contains only the transitions in $\mathbf{A} \cup \mathbf{B}$ and leads to $S'$ such that $rslv_B(S') = 2$. Consider the first transition $S \rightarrow S^*$ on this trace such that $rslv_B(S) = 0, rslv_B(S^*) > 0$. Proposition 1 implies that this must be a transition in $\mathbf{A} \cup \mathbf{A_{threat}}$. By definition 7, $A$ does not send a message to $T$ anywhere in the trace, including this transition.

By definition 8, $B$ has an $A$-silent strategy to reach a state $S''$ such that $rslv_A(S'') = 0$ or $rslv_B(S'') = 2$. Since $rslv_B(S) = 0$, it must be the case that $rslv_A(S'') = 0$, *i.e.*,

$B$ has an $A$-silent abort strategy. By lemma 1, $B$ has an $A$-silent abort strategy in $S^*$. Therefore, by lemma 3, $B$ has an abort strategy against optimistic $A$ in $S^*$.

By definition 7, $B$ has a strategy at $S_0$ to obtain $A$'s signature since $B$ controls the timeouts of $A$ and $B$. Because $S^*$ is reached a part of this strategy (recall that the $S \rightarrow S^*$ transition is on an optimistic trace), $B$ also has a strategy to obtain $A$'s signature at $S^*$. Hence $B$ has a resolve strategy against optimistic $A$ in $S^*$. Since $B$ has both abort and resolve strategies, $B$ has an advantage against an optimistic $A$ in $S^*$. □

We'd like to emphasize that the result of theorem 1 is *not* a trivial "second-mover" advantage. $A$ and $B$ are not protocol roles, but simply notation for the honest and dishonest participant, respectively. An optimistic participant is at a disadvantage *regardless* of the role he plays in the protocol. Even if he chooses the responder role, he will lose control over the outcome of the protocol at some point as long as he remains optimistic. For example, in Garay *et al.*'s abuse-free contract signing protocol [18], the originator enjoys the advantage over the responder, even though the responder is the first to receive information that potentially enables him to obtain the originator's signature.

## 5 Abuse-Free Protocols and Provable Advantage

Theorem 1 states that any fair, optimistic, timely protocol necessarily provides a dishonest participant with control over the outcome against an optimistic opponent. The problem may be alleviated by ensuring that no participant can *prove* to an outside party that he controls the outcome. Such protocols have been called *abuse-free* in the literature [18], and concrete protocols [3, 18] have been constructed using zero-knowledge cryptographic techniques such as verifiable signature escrows and designated-verifier proofs. To formalize "ability to prove," we rely on a knowledge-theoretic framework borrowed from epistemic logic [19, 16].

*Reasoning about knowledge.* Given a participant $P$ and a reachable state $S$, let $P$'s *view* of $S$ be the submultiset of $S$ containing all the facts corresponding to role states in the role theory of $P$, timers of $P$ and messages on $P$'s channels to other participants. Intuitively, this set represents all that $P$ may observe in $S$. Given a trace $tr$ from the initial state $S_0$ to $S$, construct a new labeled chain by relabeling the nodes by $P$'s view of $S$. Relabel the edges not associated with $P$ by $\epsilon$, which indicates that somebody other than $P$ may have moved. Since $P$ cannot observe other players' moves, insert an $\epsilon$ between any two consecutive edges labeled by rules of $P$ (duplicate the node connecting these edges) as well as at the start and end of the trace. If there are two or more consecutive $\epsilon$ edges, but $P$'s view does not change when moving across one of them, then delete that edge. The resulting chain $tr'$ is called $P$'s *observation* of the protocol, $Obsv_P(S, tr)$. Intuitively, $P$'s observation is just $P$'s own history in the trace.

In the spirit of algorithmic knowledge [16, 22], observations $Obsv_P(S, tr)$ and $Obsv_P(S^*, tr^*)$ are equivalent if they are computationally indistinguishable by $P$.

**Definition 9.** *Given a trace $tr$ from $S_0$ ending in $S$, we say that $P$* knows *in $(S, tr)$ that logical formula $F$ is true if*
*i) $F$ is true in $S$, and*

*ii) for each trace $tr^*$ from $S_0$ to $S^*$ such that $Obsv_P(S^*, tr^*)$ is indistinguishable by $P$ from $Obsv_P(S, tr)$, $F$ is true in $S^*$.*

Intuitively, $P$ *knows* that $F$ is true if $F$ holds in all possible executions of the protocol consistent with $P$'s observations.

*Abuse-freeness.* To reason about abuse-freeness, we augment the protocol with an outside party $C$ and consider his knowledge at different stages of the protocol. $C$ does not possess the signers' or the third party's private keys, and obtains all of his evidence about the protocol from one of the protocol participants, *e.g.*, $B$, who forwards arbitrary messages to $C$ in an attempt to cause $C$ to *know* that $A$ is participating in the protocol.

**Definition 10.** *$B$ has* provable advantage *against $A$ in state $S$ if*
*i) $B$ has an advantage over $A$ at $S$, and*
*ii) $B$ can provide information, derived from the protocol execution up to $S$, that causes $C$ to* know *that $A$ is participating in the protocol.*
*A protocol is* abuse-free *for $A$ if $B$ has no provable advantage in any reachable state.*

Definition 10 is weaker than one might expect. If $B$ enjoys an advantage at $S$, then in order for $B$ to enjoy provable advantage, $B$ merely has to prove $A$'s participation in the protocol. $B$ may succeed even if his protocol with $A$ is already over. But since we are concerned with making the protocol as safe as possible for an optimistic $A$, the weaker definition is acceptable since it makes abuse-freeness (its negation) stronger. Combining theorem 1 and definition 10, we obtain

**Corollary 1.** *In any fair, optimistic, timely, abuse-free protocol between $A$ and $B$, there is a trace $tr$ from $S_0$ to state $S$ such that*
*i) $B$ has an advantage over optimistic $A$ at $S$,*
*ii) $C$ does not know in $(S, tr)$ that $A$ is participating in the protocol,* i.e., *there is another trace $tr^*$ from $S_0$ to some $S^*$ such that $Obsv_C(S^*, tr^*)$ is indistinguishable by $C$ from $Obsv_C(S, tr)$, and $A$ is not participating in $tr^*$.*

## 6 Related work

Previous game-theoretic approaches to the study of fair exchange [11, 20, 21] focused on formalizing fairness for the strongest possible honest player without taking optimism into account. In [20], fairness is formalized as the existence of a defense strategy for the honest player, which is not sufficient if the honest player faces nondeterministic choices in the protocol, as is the case in the abuse-free protocol of Garay *et al.* [18]. Another game-theoretic model was developed in [9], but it focuses mainly on economic equilibria in fair exchange. Cryptographic proofs of correctness by protocol designers [2, 3, 18] focus on basic fairness and ignore the issues of optimism and fundamental asymmetry of communication between the signers and the trusted third party.

To the best of our knowledge, we are the first to apply an epistemic logic framework to formalize the "ability to prove" and thus abuse-freeness. In [27], belief logic SVO is used to reason about correctness of the non-repudiation protocol [26], but it is not clear how belief logics might apply to fairness and abuse-freeness. [21] models advantage, but not the concepts of proof and knowledge, which we believe provide a more compelling characterization of abuse-freeness.

## 7 Conclusions and Further Work

We have studied contract signing protocols in a game-theoretic model, giving precise, formal definitions of properties such as fairness and timeliness. We characterized optimism of honest protocol participants using a form of out-of-band signal that forces the optimistic player to wait for the opponent. While the out-of-band signal does not correspond to any realistic mechanism in distributed computation, it accurately reduces the set of protocol traces to those where the optimistic player waits for the opponent instead of contacting the trusted third party.

Our main result is that in any fair, optimistic, timely protocol, an optimistic player yields an advantage to his opponent. This means that the opponent has both a strategy to complete the signature exchange and a strategy to keep the players from obtaining each other's signatures. Since the protocol is fair, the outcome for both players is the same, but the player with an advantage can choose what this outcome is. This holds regardless of whether the optimistic player is the first or second mover.

Since advantage cannot be eliminated, the best a protocol can do to protect optimistic participants is prevent the opponent from proving to any outside party that he has reached a position of advantage. This property is known as abuse-freeness. We define abuse-freeness using the concept of algorithmic knowledge adapted from epistemic logic to formalize what it means to "prove" something to an outside observer.

One direction for further investigation involves the notion of trusted third party accountability. The relationship between our definitions and the cryptographic definitions of fairness [3] may also merit further study. Finally, we believe that our techniques will prove useful for investigating multi-party contract signing protocols.

## References


1. M. Abadi and A. Gordon. A calculus for cryptographic protocols: the spi-calculus. *Information and Computation*, 143:1–70, 1999.
2. N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for fair exchange. In *Proc. 4th ACM Conf. on Computer and Communications Security*, pages 7–17, 1997.
3. N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18(4):593–610, 2000.
4. J. Banatre and D. Le Metayer. Computing by multiset transformation. *Communications of the ACM (CACM)*, 36(1):98–111, 1993.
5. M. Ben-Or, O. Goldreich, S. Micali, and R. L. Rivest. A fair protocol for signing contracts. *IEEE Transactions on Information Theory*, 36(1):40–46, 1990.
6. G. Berry and D. Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96(1):217–248, 1992.
7. D. Boneh and M. Naor. Timed commitments and applications. In *Proc. CRYPTO 'OO*, pages 236–254, 2000.
8. C. Boyd and E. Foo. Off-line fair payment protocols using convertible signatures. In *Proc. ASIACRYPT '98*, pages 271–285, 1998.
9. L. Buttyán and J.-P. Hubaux. Toward a formal model of fair exchange — a game theoretic approach. Technical Report SSC/1999/39, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, December 1999.

10. I. Cervesato, N. Durgin, P. D. Lincoln, J. C. Mitchell, and A. Scedrov. A meta-notation for protocol analysis. In *Proc. 12th IEEE Computer Security Foundations Workshop*, pages 55–69, 1999.
11. R. Chadha, M. Kanovich, and A. Scedrov. Inductive methods and contract signing protocols. In *Proc. 8th ACM Conf. on Computer and Communications Security*, pages 176–185, 2001.
12. I. B. Damgård. Practical and provably secure release of a secret and exchange of signatures. *J. Cryptology*, 8(4):201–222, 1995.
13. D. Dolev and A. Yao. On the security of public-key protocols. In *Proc. 22nd Annual IEEE Symposium on Foundations of Computer Science*, pages 350–357, 1981.
14. S. Even and Y. Yacobi. Relations among public key signature schemes. Technical Report 175, Computer Science Dept. Technion, Israel, March 1980.
15. F.J. Thayer Fábrega, J. Herzog, and J. Guttman. Strand spaces: Why is a security protocol correct? In *Proc. IEEE Symposium on Security and Privacy*, pages 160–171, 1998.
16. R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
17. M. Fischer, N. Lynch, and M. Patterson. Impossibility of distributed consensus with one faulty process. *JACM*, 32(2):374–382, 1985.
18. J. Garay, M. Jakobsson, and P. MacKenzie. Abuse-free optimistic contract signing. In *Proc. CRYPTO '99*, pages 449–466, 1999.
19. J. Hintikka. *Knowledge and Belief*. Cornell University Press, 1962.
20. S. Kremer and J.-F. Raskin. A game-based verification of non-repudiation and fair exchange protocols. In *Proc. CONCUR '01*, pages 551–565, 2001.
21. S. Kremer and J.-F. Raskin. Game analysis of abuse-free contract signing. In *Proc. 15th IEEE Computer Security Foundations Workshop*, pages 206–220, 2002.
22. R. Pucella and J. Halpern. Modeling adversaries in a logic for security protocol analysis. In *Formal Aspects of Security, 2002 (FASec '02)*.
23. O. Markowitch and S. Saeednia. Optimistic fair exchange with transparent signature recovery. In *Proc. 5th International Conf. on Financial Cryptography*, pages 339–350, 2001.
24. H. Pagnia and F. Gaertner. On the impossibility of fair exchange without a trusted third party. Technical Report TUD-BS-1999-02, Department of Computer Science, Darmstadt University of Technology, Germany, March 1999.
25. T.Y.C. Woo and S.S. Lam. A semantic model for authentication protocols. In *Proc. IEEE Symposium on Security and Privacy*, pages 178–194, 1993.
26. J. Zhou and D. Gollmann. A fair non-repudiation protocol. In *Proc. IEEE Symposium on Security and Privacy*, pages 55–61, 1996.
27. J. Zhou and D. Gollmann. Towards verification of non-repudiation protocols. In *Proc. International Refinement Workshop and Formal Methods Pacific*, pages 370–380, 1998.

## A Role and Protocol Theories

We assume that the vocabulary contains the following basic sorts: $PK$ (for public keys), $M$ (for messages), $C$ (for pre-agreed contract texts), $PI$ (for protocol instances), and $UI$ (for globally unique instance identifiers, since we assume that each protocol instance has such an identifier). We also assume a function $\langle \_, \_, \_, \_, \_ \rangle : PK \times PK \times PK \times C \times UI \rightarrow PI$, *i.e.*, a protocol instance is determined by the signers' public key, the key of the trusted third party, pre-agreed contract text, the and unique identifier. For example, $p = \langle k_o, k_r, k_t, m, n \rangle$ describes a protocol instance, identified as $n$, in which signers with public keys $k_o$ and $k_r$ exchange signatures on the pre-agreed text $m$ with the help of the trusted third party whose key is $k_t$.

**Definition 11.** *Theory* $\mathbf{A}$ *is a* role theory for participant $A$ with public key $k_a$, *where* $k_a$ *is a constant of the sort* $PK$, *if it satisfies the following:*
*i)* $A$ *includes a finite list of predicates* $A_0, \ldots, A_n$, *called* role state predicates, *and a finite list of timer predicates, called* timers *of* $A$. *The two lists are disjoint.*
*ii)* $A_0$ *is a binary predicate whose arguments are of the sort* $PK$ *and* $PI$, *respectively. We call* $A_0$ *the* initial role state predicate.
*iii) For each rule* $l \rightarrow r$ *in* $\mathbf{A}$,

1. *There is exactly one occurrence of a role state predicate in* $l$, *say* $A_i$, *and exactly one occurrence of a role state predicate in* $r$, *say* $A_j$. *Furthermore, it is the case that* $i < j$. *If* $A_0$ *occurs in* $l$, *then* $A_0(k_a, p) \in l$ *for some term* $p$ *of the sort* $PI$.
2. *If* $A_j$ *is a* $k$*-ary role state predicate occurring in* $l$, *and* $A_j$ *is an* $m$*-ary role state predicate occurring in* $r$, *then* $m > k$. *Furthermore, if* $A_i(u_1, \ldots, u_k) \in l$ *and* $A_j(v_1, \ldots, v_m) \in l$, *then* $u_q$ *and* $v_q$ *are the same terms for all* $1 \leq q \leq k$.
3. *Let* $A_i(u_1, \ldots, u_m) \in l$, $A_j(v_1, \ldots, v_m) \in r$. *Let* $\mathcal{MSG}$ *be the set of terms* $u$ *such that* $N(u)$ *or* $tc(k_1, k_2, u) \in l$ *for some TTPchannel predicate* $tc$. *For each* $q$, $v_q$ *is derivable from* $u_1, \ldots, u_m$ *and* $\mathcal{MSG}$ *using the rules in* ***Possess***.
4. *For each timer* $Z$ *of* $A$,
   *i)* $l$ *and* $r$ *each contain at most one occurrence of* $Z$. *Occurrences are of the form* $Z(k_a, ts)$, *where* $ts$ *is a constant of the sort* $timer\_state$. *If* $Z$ *occurs in* $r$, *then it occurs in* $l$.
   *ii) If* $Z(k_a, unset) \in l$, *then either* $Z(k_a, unset) \in r$, *or* $Z(k_a, set) \in r$.
   *iii) If* $Z(k_a, set) \in l$, *then* $Z(k_a, set) \in r$.
   *iv) If* $Z(k_a, timed\_out) \in l$, *then* $Z(k_a, timed\_out) \in r$.
5. *If* $N(u) \in l$, *where* $N$ *is a network predicate and* $u$ *is term of the sort* $M$, *then* $N(u) \in r$. *If* $tc(k_1, k_2, u) \in l$, *where* $tc$ *is a TTPchannel predicate, and terms* $k_1, k_2, u$ *are of the sort* $PK, PK, M$, *respectively, then* $tc(k_1, k_2, u) \in r$.
6. *For any predicate* $\mathcal{P}$ *other than a role state, timer, network, or TTPchannel predicate, atomic formula* $\mathcal{P}(t_1, ..., t_n)$ *has the same occurrences in* $l$ *as in* $r$.

**Definition 12.** *If* $Z$ *is a timer of the participant with public key* $k_a$, *then* $Z(k_a, set) \rightarrow Z(k_a, timed\_out)$ *is the* timeout rule *of* $Z$.

**Definition 13.** *Theory* $\mathbf{P}$ *is a* protocol theory *for signers* $O$ *and* $R$ *and trusted third party* $T$ *with public keys* $k_o, k_r, k_t$, *respectively, where* $k_o, k_r, k_t$ *are constants of the sort* $PK$, *if* $\mathbf{P} = \mathbf{O} \uplus \mathbf{R} \uplus \mathbf{T_0} \uplus \mathbf{O_{timeouts}} \uplus \mathbf{R_{timeouts}} \uplus \mathbf{T_{timeouts}}$, *where*

1. $\mathbf{O}, \mathbf{R}, \mathbf{T_0}$ *are role theories for, respectively,* $O, R, T$ *with public keys* $k_o, k_r, k_t$.
2. *At most one TTPchannel predicate, say* $tc_o$, *occurs in* $\mathbf{O}$. *Each occurrence of* $tc_o$ *is of the form* $tc_o(k_o, k_t, m)$, *where* $m$ *is of the sort* $M$, *and* $tc_o$ *may not occur in* $\mathbf{R}$.
3. *At most one TTPchannel predicate, say* $tc_r$, *occurs in* $\mathbf{R}$. *Each occurrence of* $tc_r$ *is of the form* $tc_r(k_r, k_t, m)$, *where* $m$ *is of the sort* $M$, *and* $tc_r$ *may not occur in* $\mathbf{O}$.
4. *If some TTPchannel predicate occurs in* $\mathbf{T_0}$, *then it also occurs in* $\mathbf{O}$ *or* $\mathbf{R}$.
5. *The role state predicates and the timers of* $O$ *(respectively, R) do not occur in* $\mathbf{R}$ *(respectively,* $\mathbf{O}$*) and* $\mathbf{T_0}$. *The role state predicates and the timers of* $T$ *do not occur in* $\mathbf{O}$ *or* $\mathbf{R}$.
6. $\mathbf{O_{timeouts}}, \mathbf{R_{timeouts}}$, *and* $\mathbf{T_{timeouts}}$ *are the sets of timeout rules of all timers of* $O$, $R$, *and* $T$, *respectively.*

## B Proof of Lemma 1

*Proof.* We rely on the observation that state transition rules affecting independent parts of the system may be commuted. Intuitively, moves of $B$ and $T$ are independent of $A$'s internal state. Therefore, as long as $A$ does not send any messages to $T$, $B$ may ignore any message sent to him by $A$ and follow the same strategy in $S'$ as in $S$. In light of proposition 3, all we need to show is that $B$ has a weak $A$-silent abort [resolve] strategy at $S'$ if $B$ has a weak $A$-silent abort [resolve] strategy at $S$. We prove this by induction on the height of the continuation tree at $S$.
*Base case:* The height of the continuation tree at $S$ is 0. The lemma is vacuously true.
*Induction hypothesis:* Suppose the lemma is true for all states $S$ such that the height of the continuation tree at $S$ is $\leq n$.
*Induction step:* Consider state $S$ such that i) the height of the continuation tree at $S$ is $n+1$, and ii) $B$ has a weak $A$-silent abort [resolve] strategy at $S$.

Consider the continuation tree at $S'$, and remove all edges that are in $\mathbf{A} \cup \mathbf{A_{threat}} \cup \mathbf{A_{timeouts}}$ along with their descendants. For each remaining edge $e$ from $S'$ to some state $S''$, let $t$ be the state transition rule labeling $e$ and consider the following cases:

*Case* 1*:* $t \in \mathbf{T}$. Since no message is sent to $T$ in the $S \rightarrow S'$ transition, $t$ can be applied at $S$ as well, resulting in some state $\hat{S}$. Observe that:
i) the height of the continuation tree at $\hat{S}$ is $\leq n$;
ii) $B$ has a weak $A$-silent strategy at $\hat{S}$;
iii) $S''$ can be obtained from $\hat{S}$ by the same transition that labels $S \rightarrow S'$: simply commute $S \rightarrow S'$ and $S' \rightarrow S''$ transitions.
By the induction hypothesis, $B$ has a weak $A$-silent strategy at $S''$. Replace the continuation tree at $S''$ by this strategy.

*Case* 2*:* $t \in \mathbf{B} \cup \mathbf{B_{threat}}$. There are three possibilities:
2.1) $t$ cannot be applied at $S$. Remove edge $e$ along with its descendants.
2.2) $t$ can be applied at $S$, but it is not a part of the $A$-silent strategy at $S$. Remove edge $e$ along with its descendants.
2.3) $t$ can be applied at $S$, and it is a part of the $A$-silent strategy at $S$. Then, as in Case 1, replace the continuation tree at $S''$ by this strategy.

*Case* 3*:* $t \in \mathbf{B_{timeouts}}$. If $t$ is not a part of the $A$-silent strategy at $S$, remove edge $e$ along with its descendants. If it is a part of the $A$-silent strategy, replace the continuation tree at $S''$ by this strategy.

By constructing the right continuation tree for any immediate descendant of $S'$, we have constructed a weak $A$-silent strategy at $S'$. It remains to show that it is indeed an abort [resolve] strategy. There are two possibilities :

*Case A*: The height of the constructed strategy is 0. From the construction, it follows that the height of the weak $A$-silent abort [resolve] strategy at $S$ is also 0. Therefore, $rslv_A(S) = 0$ [$rslv_B(S) = 2$]. By proposition 1, $rslv_A(S') = 0$ [$rslv_B(S') = 2$].

*Case B*: The height of the constructed strategy is $> 0$. By construction, all leaf nodes are labeled by states $S^*$ such that $rslv_A(S^*) = 0$ [$rslv_B(S^*) = 2$].

We conclude that $B$ has a weak $A$-silent abort [resolve] strategy at $S'$, which completes the induction. □