INSTITUT FÜR INFORMATIK

UND PRAKTISCHE MATHEMATIK

Deciding Properties of Contract-Signing Protocols

Detlef Kähler and Ralf Küsters and Thomas Wilke

Bericht Nr. 0409
September 2004

CHRISTIAN-ALBRECHTS-UNIVERSITÄT

KIEL

Institut für Informatik und Praktische Mathematik der
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40
D – 24098 Kiel

Deciding Properties of Contract-Signing Protocols

Detlef Kähler and Ralf Küsters and Thomas Wilke

Bericht Nr. 0409
September 2004

e-mail: {kaehler,kuesters,wilke}@ti.informatik.uni-kiel.de

Deciding Properties of Contract-Signing Protocols

Detlef Kähler Ralf Küsters
Thomas Wilke
Christian-Albrechts-Universität zu Kiel
{kaehler,kuesters,wilke}@ti.informatik.uni-kiel.de

Abstract

We show that for infinite transition systems induced by cryptographic protocols in the Rusinowitch/Turuani style with finite number of sessions, unbounded message size, and the Dolev-Yao intruder certain fundamental branching properties are decidable. As a consequence, we obtain that crucial properties of contract-signing protocols such as balance are decidable.

1 Introduction

Automatic analysis of cryptographic protocols has been studied intensively in the recent past (see, e.g., [10, 14] for an overview) and has led to industrial-strength debugging tools (see, e.g., [2]). One of the central results of the area is that the security of cryptographic protocols when analyzed w.r.t. a finite number of sessions, without a bound on the message size, and in presence of the so-called Dolev-Yao intruder is decidable (see, e.g., [16, 1, 5, 15]). This result (and all the related ones) is, however, restricted to security properties such as authenticity and secrecy, which are reachability properties of the transition system associated with a protocol: Is a state, in which the intruder possesses a certain secret, such as a session key, reachable? In contrast, crucial properties required of contract-signing and related protocols [12, 3, 4, 18], for instance abuse-freeness [12] and balance [6], are properties of the structure of the transition system associated with a protocol. Balance, for instance, requires that in no stage of a protocol run, the intruder (or a dishonest party) has both a strategy to abort the run and a strategy to successfully complete the run and thus obtain a valid contract.

The main goal of this paper is to show that the aforementioned result extends in a natural way to branching properties, such as balance, and similar properties of contract-signing protocols. In other words, the goal is to show that these branching properties are decidable w.r.t. a finite number of sessions, without a bound on the message size, and in presence of the so-called Dolev-Yao intruder. This can potentially lead to fully automatic analyzers for contract-signing protocols that are much more precise than the existing ones (which consider only drastically scaled down finite-state versions of the protocols in question).

The protocol and intruder model that we suggest to use is a "finite session" version of a model proposed in [6];[1]it contains different features important for contract-signing protocols, which are absent in the models for authentication and key exchange protocols referred to above. First, as in

[1]Our technical exposition though is closer to the term-rewriting approach [16] than to the multi-set rewriting framework employed in [6] (see also [8]).

[6], we include private contract signatures in our model as these signatures are used for instance in the contract-signing protocol proposed in [12]. Second, as in [6], we model write-protected channels which are *not* under the control of the intruder. In this paper, we call these channels *secure channels* for simplicity, although this notion is also used in cryptography with a different meaning. Third, for protocols in our model we explicitly define the induced transition systems. These transition systems have infinitely many states and are infinitely branching, but have paths of bounded length, and allow us to state crucial properties of contract-signing properties.

Our main technical result is that for the transition systems induced by a cryptographic protocol properties expressing the existence of certain strategies for the intruder are decidable. We show how to reduce balance to these properties. We also consider other important properties of contract-signing protocols, namely effectiveness and fairness, and show that these properties can be reduced to reachability properties slightly more general than those mentioned above. These properties are also shown to be decidable.

The basic technique used in our proofs is the same as the one first introduced in [16], where they show that to find an attack on a protocol only reasonably small substitutions have to be considered. In our framework, we extend and modify this idea. We have to overcome the problem that in the nature of a strategy there is a requirement that *all* possible counter-measures of an opponent have to be taken into account, no matter how small or large the substitutions involved are.

In several papers, contract-signing and related protocols have been analyzed using finite-state model checkers (see, e.g., [17, 13]). Due to the restriction to a finite state set, the Dolev-Yao intruder is, however, only approximated. A much more detailed model has been considered by Chadha et al. [6], who analyzed the contract-signing protocol proposed by Garay, Jakobsson, and MacKenzie [12], even taking into account an unbounded number of sessions. However, the analysis was carried out by hand and without tool support. As mentioned, our model is the "finite session" version of the model by Chadha et al. (Allowing an unbounded number of sessions would render the decision problems we consider undecidable.) Hence, our results show that when restricted to a finite number of sessions, the analysis carried out in [6] can be fully automated (given a specification of the protocols) without resorting to finite-state models as done in the works mentioned above. Drielsma and Mödersheim [11] apply an automatic tool originally intended for authentication and key exchange protocols in the analysis of the ASW protocol [3]. Their analysis is, however, restricted to reachability properties as branching properties cannot be handled by their tool. Also, secure channels are not modeled explicitly in that paper.

Structure of our paper. In Section 2 we introduce the protocol and intruder model, with an example provided in Section 3. We then define the properties of induced transition systems we show to be decidable and state decidability (Section 4). In Section 5, we show how the security problems for concret security properties of contract-signing protocols can be reduced to the abstract problems.

Full proofs of our results can be found in the appendix.

2 The Protocol and Intruder Model

As mentioned in the introduction, in essence, our model is the "finite session" version of the model proposed in [6]. When it comes to the technical exposition, our approach is, however, inspired by the term-rewriting approach of [16] rather than the multi-set rewriting approach of [6].

In our model, a protocol is a finite set of principals and every principal is a finite tree, which represents all possible behaviours of the principal. Each edge of such a tree is labelled by a rewrite rule, which describes the receive-send action that is performed when the principal takes this edge in a run of the protocol.

When a principal carries out a protocol, it traverses its tree, starting at the root. In every node, the principal takes its current input, chooses one of the edges leaving the node, matches the current input with the left-hand side of the rule the edge is labelled with, sends out the message which is determined by the right-hand side of the rule, and moves to the node the chosen edge leads to. Whether or not a principal gets an input and which input it gets is determined by the intruder (or the secure channel, see below), who receives every message sent by a principal, can use all the messages he has received to construct new messages, and can provide input messages to any principal he wants—this is the usual Dolev-Yao model (see, e.g., [16]).

The above is very similar to the approach in [16]. There are, however, three important ingredients that are not present in [16]: secure channels, private contract signatures, and an explicit branching structure.

Secure channels. Unlike in the standard Dolev-Yao model, in our model the input of a principal may not only come from the intruder but also from a so-called secure channel. While a secure channel is not read-protected (the intruder can read the messages written onto this channel), the intruder does not control this channel. That is, he cannot delay, duplicate, or remove messages, or write messages onto this channel under a fake identity (unless he has corrupted a party).

Branching structure. As mentioned in the introduction, unlike authentication and key-exchange protocols, properties of contract-signing and related protocols cannot be stated as reachability properties, i.e., in terms of single runs of a protocol alone. One rather has to consider branching properties. We therefore describe the behavior of a protocol as an infinite-state transition system which comprises all runs of a protocol. To be able to express properties of contract-signing protocols we distinguish several types of transitions: there are intruder transitions (just as in [16]), there are ε-transitions, which can be used to model that a subprotocol is spawned without waiting for input from the intruder, and secure channel transitions, which model communication via secure channels. Since the intruder can construct an infinite number of messages, the transition system will have an infinite number of states, but it will have paths of a bounded length.

2.1 Terms and Messages

We have a finite set $\mathcal{V}$ of variables, a finite set $\mathcal{A}$ of atoms, a finite set $\mathcal{K}$ of public and private keys, an infinite set $\mathcal{A}_I$ of intruder atoms, and a finite set $\mathcal{N}$ of principal addresses. All of them are assumed to be disjoint.

The set $\mathcal{K}$ is partitioned into a set $\mathcal{K}_{pub}$ of public keys and a set $\mathcal{K}_{priv}$ of private keys. There is a bijective mapping $\cdot^{-1}\colon \mathcal{K} \to \mathcal{K}$ which assigns to every public key the corresponding private key and to every private key its corresponding public key.

Typically, the set $\mathcal{A}$ contains names of principals, atomic symmetric keys, and nonces (i.e., random numbers generated by principals). We note that we will allow non-atomic symmetric keys as well. The atoms in $\mathcal{A}_I$ are the nonces, symmetric keys, etc. the intruder may generate. The elements of $\mathcal{N}$ are used as addresses of principals in secure channels.

We define two kinds of terms by the following grammar: *plain terms*, which model (patterns of) messages that are actually processed, and *secure channel terms*, which besides a plain message

also contain a sender and a receiver address:

$$\begin{array}{rcl} \textit{plain-terms} & ::= & \mathcal{V} \mid \mathcal{A} \mid \mathcal{A}_I \mid \langle \textit{plain-terms}, \textit{plain-terms} \rangle \mid \{\textit{plain-terms}\}^s_{\textit{plain-terms}} \mid \\ & & \{\textit{plain-terms}\}^a_{\mathcal{K}} \mid \mathsf{hash}(\textit{plain-terms}) \mid \mathsf{sig}_{\mathcal{K}}(\textit{plain-terms}) \mid \\ & & \mathsf{PCS}_{\mathcal{K}}(\textit{plain-terms}, \mathcal{K}, \mathcal{K}) \mid \mathsf{PCS}^f_{\mathcal{K}}(\textit{plain-terms}, \mathcal{K}, \mathcal{K}) \mid \\ & & \mathsf{ssig}_{\mathcal{K}}(\textit{plain-terms}, \mathcal{K}, \mathcal{K}) \mid \mathsf{tsig}_{\mathcal{K}}(\textit{plain-terms}, \mathcal{K}, \mathcal{K}) \\ \textit{terms} & ::= & \textit{plain-terms} \mid \mathsf{sc}(\mathcal{N}, \mathcal{N}, \textit{plain-terms}) \mid \mathcal{N} \end{array}$$

Plain terms, secure channel terms, and terms without variables (i.e., ground terms) are called *plain messages*, *secure channel messages*, and *messages*, respectively. As usual, $\langle t, t' \rangle$ is the pairing of t and t', the term $\{t\}^s_{t'}$ stands for the symmetric encryption of t by t' (note that the key t' may be any plain term), $\{t\}^a_k$ is the asymmetric encryption of t by k, the term $\mathsf{hash}(t)$ stands for the hash of t, and $\mathsf{sig}_k(t)$ is the signature on t which can be verified with the public key k. A term of the form $\mathsf{PCS}_{k_O}(t, k_R, k_T)$ stands for the private contract signature on the contract t computed using the private key ${k_O}^{-1}$ corresponding to the public key k_O. While it can be verified by all principals (who know the public keys), only O, R, and T can tell who of O or R actually computed the signature.[2] Given $\mathsf{PCS}_{k_O}(t, k_R, k_T)$, O and T (but not R) can turn this private contract signature into a universally verifiable signature. More precisely, O can turn $\mathsf{PCS}_{k_O}(t, k_R, k_T)$ into $\mathsf{ssig}_{k_O}(t, k_R, k_T)$ and T can turn $\mathsf{PCS}_{k_O}(t, k_R, k_T)$ into $\mathsf{tsig}_{k_O}(t, k_R, k_T)$ (this models that T is accountable, since $\mathsf{tsig}_{k_O}(t, k_R, k_T)$ is different from $\mathsf{ssig}_{k_O}(t, k_R, k_T)$). Also, O can generate a "fake" signature $\mathsf{PCS}^f_{k_R}(t, k_O, k_T)$ which by principals other than O, R, and T will be recognized as valid. See [12, 6] for more details on private contract signatures.

A secure channel term of the form $\mathsf{sc}(n, n', t)$ stands for feeding the secure channel from n to n' with t. A principal may only generate such a term if he knows n and t but not necessarily n'. This guarantees that a principal cannot impersonate other principals on the secure channel. Hence, knowing n grants access to secure channels with sender address n.

Note that the above explanations are just to give some intuition on what these messages can be used for and how they can be used. In our formel model, there will be no restriction on the use of these messages in protocol descriptions except for the rules by which the intruder can derive messages.

A *substitution* assigns terms to variables. The *domain* of a substitution is denoted by $\mathsf{dom}(\sigma)$ and defined by $\mathsf{dom}(\sigma) = \{x \in \mathcal{V} \mid \sigma(x) \neq x\}$. Substitutions are required to have finite domains and it is required that $\sigma(x)$ is a ground term for each $x \in \mathsf{dom}(\sigma)$. Given two substitutions σ and σ' with disjoint domains, their union $\sigma \cup \sigma'$ is defined in the obvious way. Given a term t, the term $t\sigma$ is obtained from t by simultaneously substituting each variables x occurring in t by $\sigma(x)$.

2.2 Principals and Protocols

Principal rules are of the form $R \Rightarrow S$ where R is a term or ε and S is a term.

A *principal* $\Pi = (V, E, r, \ell)$ is a finite tree rooted at $r \in V$ where ℓ maps every edge $(v, v') \in E$ of Π to a principal rule $\ell(v, v')$ in such a way that every variable occurring on the right-hand side of $\ell(v, v')$ also occurs on the left-hand side of $\ell(v, v')$ or on the left-hand side of a principal rule on the path from r to v. In other words, every variable occurring on the right-hand side of a principal

[2] When we say O, R, or T, we mean the principals who possess the private keys ${k_O}^{-1}$, ${k_R}^{-1}$, and ${k_T}^{-1}$.

rule also occurs on the left-hand side of this or a preceeding principal rule. We call Π a *rule tree* if the last restriction on the variables is not required to be satisfied.

For $v \in V$, we write $\Pi{\downarrow}v$ to denote the subtree of Π rooted at v. For a substitution σ, we write $\Pi\sigma$ for the principal obtained from Π by substituting all variables x occurring in the principal rules of Π by $\sigma(x)$.

A *protocol* $P = ((\Pi_1, \ldots, \Pi_n), \mathcal{I})$ consists of a finite set of principals and a finite set $\mathcal{I}$ of messages, the *initial intruder knowledge*. We require that each variable occurs in the rules of only one principal, i.e., different principals must have disjoint sets of variables. We assume that intruder atoms, i.e., elements of $\mathcal{A}_I$, do not occur in P.

2.3 Intruder

Given a set $\mathcal{I}$ of messages, the (infinite) set $d(\mathcal{I})$ of messages the intruder can derive from $\mathcal{I}$ is the smallest set satisfying the following conditions:

1. $\mathcal{I} \subseteq d(\mathcal{I})$.
2. *Composition and decomposition*: If $m, m' \in d(\mathcal{I})$, then $\langle m, m' \rangle \in d(\mathcal{I})$. Conversely, if $\langle m, m' \rangle \in d(\mathcal{I})$, then $m \in d(\mathcal{I})$ and $m' \in d(\mathcal{I})$.
3. *Symmetric encryption and decryption*: If $m, m' \in d(\mathcal{I})$, then $\{m\}^s_{m'} \in d(\mathcal{I})$. Conversely, if $\{m\}^s_{m'} \in d(\mathcal{I})$ and $m' \in d(\mathcal{I})$, then $m \in d(\mathcal{I})$.
4. *Asymmetric encryption and decryption*: If $m \in d(\mathcal{I})$ and $k \in d(\mathcal{I}) \cap \mathcal{K}$, then $\{m\}^a_k \in d(\mathcal{I})$. Conversely, if $\{m\}^a_k \in d(\mathcal{I})$ and $k^{-1} \in d(\mathcal{I})$, then $m \in d(\mathcal{I})$.
5. *Hashing*: If $m \in d(\mathcal{I})$, then $\mathsf{hash}(m) \in d(\mathcal{I})$.
6. *Signing*: If $m \in d(\mathcal{I}), k^{-1} \in d(\mathcal{I}) \cap \mathcal{K}$, then $\mathsf{sig}_k(m)$. (The signature contains the public key but can only be generated if the corresponding private key is known.)
7. *Private contract signature*: If $m \in d(\mathcal{I}), {k_0}^{-1}, k_1, k_2 \in d(\mathcal{I}) \cap \mathcal{K}$, then $\mathsf{PCS}_{k_0}(m, k_1, k_2)$.
8. *Fake private contract signature*: If $m \in d(\mathcal{I}), {k_0}^{-1}, k_1, k_2 \in d(\mathcal{I}) \cap \mathcal{K}$, then $\mathsf{PCS}^f_{k_1}(m, k_0, k_2)$.
9. *Converting private contract signatures*: If $\mathsf{PCS}_{k_0}(m, k_1, k_3) \in d(\mathcal{I})$ and ${k_0}^{-1} \in d(\mathcal{I})$, then $\mathsf{ssig}_{k_0}(m, k_1, k_3) \in d(\mathcal{I})$. Also, if $\mathsf{PCS}_{k_0}(m, k_1, k_3) \in d(\mathcal{I})$ and ${k_3}^{-1} \in d(\mathcal{I})$, then $\mathsf{tsig}_{k_0}(m, k_1, k_3) \in d(\mathcal{I})$.
10. *Writing onto and read from the secure channel:* If $m \in d(\mathcal{I})$, $n_1 \in d(\mathcal{I}) \cap \mathcal{N}$, and $n_2 \in \mathcal{N}$, then $\mathsf{sc}(n_1, n_2, m) \in d(\mathcal{I})$. If $\mathsf{sc}(n_1, n_2, m) \in d(\mathcal{I})$, then $m \in d(\mathcal{I})$.
11. *Generating fresh constants*: $\mathcal{A}_I \subseteq d(\mathcal{I})$.

Each of the above rules only applies when the resulting expression is a term according to the grammar stated above. For instance, a hash of a secure channel term is not a term, so rule 5 does not apply when m is of the form $\mathsf{sc}(n, n', m')$.

Intuitively, $n \in d(\mathcal{I}) \cap \mathcal{N}$ means that the intruder has corrupted the principal with address n and therefore can impersonate this principal when writing onto the secure channel. Also, the intruder can extract m from $\mathsf{sc}(n, n', m)$ since, just as in [6], the secure channel is not read-protected. (However, our results hold independent of whether or not the secure channel is read-protected.)

2.4 The Transition Graph Induced by a Protocol

We define the transition graph $\mathcal{G}_P$ induced by a protocol P and start with the definition of the states and the transitions between these states.

A *state* is of the form $((\Pi_1, \ldots, \Pi_n), \sigma, \mathcal{I}, \mathcal{S})$ where

1. σ is a substitution.

2. For each i, Π_i is a rule tree such that $\Pi_i\sigma$ is a principal.
3. $\mathcal{I}$ is a finite set of messages, the *intruder knowledge*.
4. $\mathcal{S}$ is a finite multi-set of secure channel messages, the *secure channel*.

The idea is that when the transition system gets to such a state, then the substitution σ has been performed, the accumulated intruder knowledge is what can be derived from $\mathcal{I}$, the secure channels hold the messages in $\mathcal{S}$, and for each i, Π_i is the "remaining protocol" to be carried out by principal i. This also explains why $\mathcal{S}$ is a multi-set: messages sent several times should be delivered several times.

Given a protocol $P = ((\Pi_1, \ldots, \Pi_n), \mathcal{I})$ the *initial state of* P is $((\Pi_1, \ldots, \Pi_n), \sigma, \mathcal{I}, \emptyset)$ where σ is the substitution with empty domain.

We have three kinds of transitions: intruder, secure channel, and ε-transitions. In what follows, let $\Pi_i = (V_i, E_i, r_i, \ell_i)$ and $\Pi_i' = (V_i', E_i', r_i', \ell_i')$ denote rule trees. We define under which circumstances there is a transition

$$((\Pi_1, \ldots, \Pi_n), \sigma, \mathcal{I}, \mathcal{S}) \xrightarrow{\tau} ((\Pi_1', \ldots, \Pi_n'), \sigma', \mathcal{I}', \mathcal{S}')$$

with τ an appropriate label.

1. *Intruder transitions:* The above transition with label i, m, I exists if there exists $v \in V_i$ with $(r_i, v) \in E_i$ and $\ell_i(r_i, v) = R \Rightarrow S$, and a substitution σ'' of the variables in $R\sigma$ such that
   (a) $m \in d(\mathcal{I})$,
   (b) $\sigma' = \sigma \cup \sigma''$,
   (c) $R\sigma' = m$,
   (d) $\Pi_j' = \Pi_j$ for every $j \neq i$, $\Pi_i' = \Pi_i{\downarrow}v$,
   (e) $\mathcal{I}' = \mathcal{I} \cup \{S\sigma'\}$,
   (f) $\mathcal{S}' = \mathcal{S}$ if $S \neq \mathsf{sc}(\cdot,\cdot,\cdot)$, and $\mathcal{S}' = \mathcal{S} \cup \{S\sigma'\}$ otherwise.

   This transition models that principal i reads the message m from the intruder (i.e., the public network).
2. *Secure channel transitions:* The above transition with label i, m, sc exists if there exists $v \in V_i$ with $(r_i, v) \in E_i$ and $\ell_i(r_i, v) = R \Rightarrow S$, and a substitution σ'' of the variables in $R\sigma$ such that $m \in \mathcal{S}$, (b)–(e) from 1., and $\mathcal{S}' = \mathcal{S} \setminus \{m\}$ if $S \neq \mathsf{sc}(\cdot,\cdot,\cdot)$, and $\mathcal{S}' = (\mathcal{S} \setminus \{m\}) \cup \{S\sigma'\}$ otherwise.

   This transition models that principal i reads message m from the secure channel.
3. *ε-transitions:* The above transition with label i exists if there exists $v \in V_i$ with $(r_i, v) \in E_i$ and $\ell_i(r_i, v) = \varepsilon \Rightarrow S$ such that $\sigma' = \sigma$ and (d), (e), (f) from above.

   This transition models that i performs a step where neither a message is read from the intruder nor from the secure channel.

If $q \xrightarrow{\tau} q'$ is a transition where the first component of the label τ is i, then the transition is called an *i-transition* and q' an *i-successor* of q.

Given a protocol P, the *transition graph $\mathcal{G}_P$ induced by P* is the tuple (S_P, E_P, q_P) where q_P is the initial state of P, S_P is the set of states reachable from q_P by a sequence of transitions, and E_P is the set of all transitions among states in S_P. Formally, a transition $q \xrightarrow{\tau} q'$ is a tuple (q, τ, q').

We write $q \in \mathcal{G}_P$ if q is a state in $\mathcal{G}_P$ and $q \xrightarrow{\tau} q' \in \mathcal{G}_P$ if $q \xrightarrow{\tau} q'$ is a transition in $\mathcal{G}_P$.

Remark 1. *The transition graph $\mathcal{G}_P$ of P is a DAG since by performing a transition the size of the first component of a state decreases. While the graph may be infinite branching, the maximal length of a path in this graph is bounded by the total number of edges in the principals Π_i of P.*

For a state q, we denote the subgraph of $\mathcal{G}_P$ consisting of all states reachable from q by $\mathcal{G}_{P,q}$.

3 Modelling the Originator of the ASW Protocol

To demonstrate that our framework can actually be used to analyze contract-signing protocols, we show how the originator of the Asokan-Shoup-Waidner (ASW) protocol [3] can be modelled. In a similar fashion, other contract-signing protocols, such as the Garay-Jakobsson-MacKenzie protocol [12], can be dealt with.

3.1 Overview of the Protocol

Our informal description of the ASW protocol follows [17] (see this work or [3] for more details). For ease in notation, we will write $\mathsf{sig}[m, k]$ instead of $\langle m, \mathsf{sig}_k(m)\rangle$.

The ASW protocol enables two principals O (originator) and R (responder) to obtain each other's commitment on a previously agreed contractual text, say text, with the help of a trusted third party T, which, however, is only invoked in case of problems. In other words, the ASW protocol is an optimistic two-party contract-signing protocol.

There are two kinds of valid contracts: the standard contract, $\langle \mathsf{sig}[m_O, k_O], N_O, \mathsf{sig}[m_R, k_R], N_R\rangle$, and the replacement contract, $\mathsf{sig}[\langle \mathsf{sig}[m_O, k_O], \mathsf{sig}[m_R, k_R]\rangle, k_T]$, where $m_O = \langle k_O, k_R, k_T, \mathsf{text}, \mathsf{hash}(N_O)\rangle$, $m_R = \langle \mathsf{sig}[m_O, k_O], \mathsf{hash}(N_R)\rangle$, and k_T is the key of the trusted third party. The keys k_O, k_R, and k_T are used for identifying the principals. Note that a signed contractual text ($\mathsf{sig}[\mathsf{text}, k_O]$ or $\mathsf{sig}[\mathsf{text}, k_R]$) is not considered a valid contract.

The ASW protocol consists of three subprotocols: the exchange, abort, and resolve protocol. However, we can describe every principal—O, R, and T—in terms of a single tree as introduced in Section 2.2.

The basic idea of the exchange protocol is that O first indicates his/her interest to sign the contract. To this end, O hashes a nonce N_O and signs it together with text and the keys of the principals involved. The resulting message is the message $\mathsf{sig}[m_O, k_O]$ from above. By sending it to R, O commits to the contract. Then, similarly, R indicates his/her interest to sign the contract by hashing a nonce N_R and signing it together with text and the keys of the involved principals. This is the message m_R from above. By sending it to O, R commits to the contract. Finally, first O and then R reveal N_O and N_R, respectively. This is why a standard contract is only valid if N_O and N_R are included.

If, after O has sent the first message, R does not respond, O may contact T to abort. At any point, if one of O and R does not respond, the other may contact T to resolve. In case the protocol is successfully resolved, the replacement contract $\mathsf{sig}[\langle \mathsf{sig}[m_O, k_O], \mathsf{sig}[m_R, k_R]\rangle, k_T]$ is issued. While this version of the contract only contains the message indicating O's and R's intention to sign the contract (and neither N_O nor N_R), the signature of T validates the contract.

In the next subsection, the model of O is presented. The models for R and T as well as the security properties for the ASW protocol can be found in the Appendix A.

3.2 The Principal O

The principal O is defined by the tree Π_O depicted in Figure 1 where the numbers stand for the principal rules defined below. Rules 1, 2, and 3 belong to the exchange protocol, rules 4, 5, and 6 belong to the abort protocol, and rules 7, 8, and 9 belong to the resolve protocol.

Exchange protocol. The actions performed in the exchange protocol have informally been discussed above.

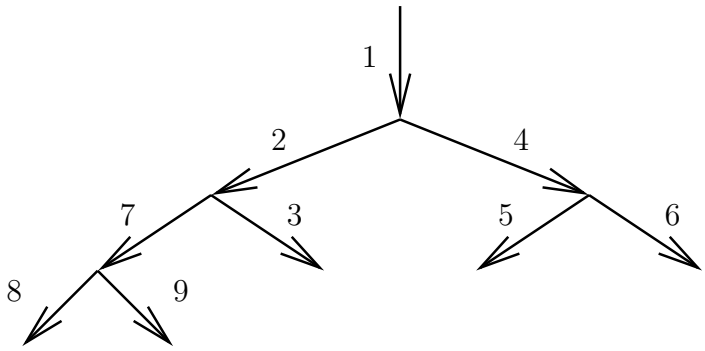

Figure 1: The Originator O

Abort protocol. If, after the first step of the exchange protocol, O does not get an answer back from R, the principal O may start the abort protocol, i.e., send an abort request via a secure channel to T (rule 4). Then, T will either confirm the abort of the protocol by returning an abort token—in this case O will continue with rule 5—or send a resolve token—in this case O will continue with rule 6. (The trusted third party T sends a resolve token if R previously contacted T to resolve the protocol run.)

Resolve protocol. If after rule 2, i. e., after sending N_O, the principal O does not get an answer back from R, then O can start the resolve protocol by sending a resolve request to T via the secure channel (rule 7). After that, depending on the answer returned from T (which again will return an abort or resolve token), one of the rules 8 or 9 is performed.

We now present the principal rules for O where the numbering corresponds to the one in Figure 1. Any occurrence of _ should be substituted by a new fresh variable, that is, the term which is matched is not used afterwards.

1. $\varepsilon \Rightarrow me_1$ where
$$me_1 = \mathsf{sig}[me_2, k_O] \quad \text{and} \quad me_2 = \langle k_O, k_R, k_T, \mathsf{text}, \mathsf{hash}(N_O)\rangle\,.$$
2. $\mathsf{sig}[me_3, k_R] \Rightarrow N_O$ where $me_3 = \langle me_1, \mathsf{hash}(x)\rangle\,.$
3. $x \Rightarrow \mathsf{OHasValidContract}$.
4. $\varepsilon \Rightarrow \mathsf{sc}(O, T, ma_1)$ where $ma_1 = \mathsf{sig}[\langle \mathsf{aborted}, me_1\rangle\,, k_O]$.
5. $\mathsf{sc}(T, O, ma_2) \Rightarrow \mathsf{OHasValidContract}$ where
$$ma_2 = \mathsf{sig}[\langle me_1, me_4\rangle\,, k_T] \quad \text{and} \quad me_4 = \mathsf{sig}[\langle me_1, \_\rangle\,, k_R].$$
6. $\mathsf{sc}(T, O, \mathsf{sig}[\langle \mathsf{aborted}, ma_1\rangle\,, k_T]) \Rightarrow \mathsf{OHasAbortToken}$.
7. $\varepsilon \Rightarrow \mathsf{sc}(O, T, \langle me_1, \mathsf{sig}[me_3, k_R]\rangle)$.
8. $\mathsf{sc}(T, O, \mathsf{sig}[\langle \mathsf{aborted}, ma_1\rangle\,, k_T]) \Rightarrow \mathsf{OHasAbortToken}$.
9. $\mathsf{sc}(T, O, mr_1) \Rightarrow \mathsf{OHasValidContract}$ where
$$mr_1 = \mathsf{sig}[\langle me_1, mr_2\rangle\,, k_T] \quad \text{and} \quad mr_2 = \mathsf{sig}[\langle me_1, \_\rangle\,, k_R].$$

4 Abstract Properties and Decidability Results

In what follows, we formulate abstract properties of transition graphs that request the existence of certain state or subgraphs (strategy graphs) and state decidability of these properties.

The abstract properties are divided into reachability properties and strategy properties.

4.1 Reachability Properties

Given a transition graph $\mathcal{G}_P = (S_P, E_P, q_P)$ of a protocol $P = ((\Pi_1, \ldots, \Pi_n), \mathcal{I})$, a *path* π in $\mathcal{G}_P$ is of the form

$$p_0 \xrightarrow{\lambda_0} p_1 \xrightarrow{\lambda_1} p_2 \xrightarrow{\lambda_2} \cdots \xrightarrow{\lambda_{l-1}} p_l$$

where $p_i \xrightarrow{\lambda_i} p_{i+1} \in E_P$ for every i. We say that π is *rooted* if $p_0 = q_P$.

A *reachability property* is a tuple (C, C', θ) where $C, C' \subseteq \mathcal{A} \cup \mathcal{K} \cup \mathcal{N}$ and $\theta \in \{2^{\{1,\ldots,n\}}, \text{noTrans}\}$. Elements in $\{1, \ldots, n\}$ denote the principals in P. Recall that these numbers are used as labels in transitions of the transition graph.

A rooted path π as above (or the final state p_l of this path) satisfies (C, C', θ) if the following conditions are satisfied:

1. $C \subseteq d(\mathcal{I})$ and $C' \cap d(\mathcal{I}) = \emptyset$ where $\mathcal{I}$ is the intruder's knowledge in state p_l, i.e., in state p_l, the intruder can derive all constants in C but cannot derive those in C'.
2. If $\theta \subseteq \{1, \ldots, n\}$, then no outgoing secure channel or ε-transition of p_l is labeled by one of the principals in θ. (Secure channel and ε-transitions of other principals are allowed though as well as arbitrary intruder transtions).
3. If $\theta = \text{noTrans}$, then p_l does not have outgoing transitions (of any kind).

Observe that $\theta = \emptyset$ just means that we don't care about which transitions leave the final state of the path.

The abstract reachability problem Reachability asks, given a protocol P and a reachability property (C, C', θ), whether there exists a rooted path in $\mathcal{G}_P$ which satisfies (C, C', θ).

We note that the secrecy problem usually considered for authentication and key exchange protocols is an instance of the path problem where the path property is defined to be $(\{\mathsf{secret}\}, \emptyset, *)$ for some fixed constant secret.

We show (see Appendix B):

Theorem 2. Reachability *is decidable.*

Similar to [16], to prove this theorem, we show how a given large path—a path with possibly large substitutions—which satisfies the given reachability property can be reduced in size. While the proof in [16] applies to reachability properties of the form $(C, C', \emptyset)$ (except that in [16] no secure channels or private contract signatures are considered), for properties where the last component is not the empty set, we need to make sure that, when reducing the size of the large path, no additional transitions are made applicable in the transition graph. Here we use that the intruder can generate new constants.

4.2 Strategy Graphs and the Strategy Property

We first define strategy graphs and then define the strategy property.

The notion of a strategy graph captures that the intruder has a way of acting such that regardless of how the other principals act, he achieves a certain goal, where goal in our context means that a state will be reached where the intruder can derive certain constants and cannot derive others.

A q-strategy graph $\mathcal{G}_q$ is a sub transition system of $\mathcal{G}_P$ where q is the initial state of $\mathcal{G}_q$ and such that for all states q' in $\mathcal{G}_q$, the following conditons, which are explained below, are satisfied.

1. If $q' \xrightarrow{j} q'' \in \mathcal{G}_P$, then $q' \xrightarrow{j} q'' \in \mathcal{G}_q$ for every j and q''.
2. If $q' \xrightarrow{j,m,\mathsf{sc}} q'' \in \mathcal{G}_P$, then $q' \xrightarrow{j,m,\mathsf{sc}} q'' \in \mathcal{G}_q$ for every m, j, and q''.
3. If $q' \xrightarrow{j,m,I} q'' \in \mathcal{G}_q$ and $q' \xrightarrow{j,m,I} q''' \in \mathcal{G}_P$, then $q' \xrightarrow{j,m,I} q''' \in \mathcal{G}_q$ for every m, j, q'', and q'''.

The first condition says that every ε-transition of the original transition system must be present in the strategy graph; this is because the intruder should not be able to prevent a principal from performing an ε-rule. The second condition is similar: the intruder should not be able to block the secure channels. The third condition says that though the intruder can choose to send a particular message to a particular principal, he cannot decide which transition this principal uses (if the message matches two rules).

A *strategy property* is a tuple $((C_1, C_1'), \ldots, (C_s, C_s'))$, where $C_i, C_i' \subseteq \mathcal{A} \cup \mathcal{K} \cup \mathcal{N}$. A state q satisfies $((C_1, C_1'), \ldots, (C_s, C_s'))$ if there exist q-strategy graphs $\mathcal{G}_1, \ldots, \mathcal{G}_s$ such that every $\mathcal{G}_i$ satisfies (C_i, C_i'), where $\mathcal{G}_i$ satisfies (C_i, C_i') if for all leaves v_i of $\mathcal{G}_i$ all elements from C_i can be derived by the intruder and all elements from C_i' cannot.

The abstract path problem Strategy asks, given a protocol P and a strategy property $((C_1, C_1'), \ldots, (C_s, C_s'))$, whether there exists a state q that satisfies the property.

We show (Appendix B):

Theorem 3. Strategy *is decidable.*

To prove this theorem, we show that given a possibly large state q and large q-strategy graphs satisfying the properties, we can reduce the size of the state and the strategy graphs, i.e., the size of the substitutions in the state and the graphs. For this purpose, we need to deal with all substitutions in all of the strategy graphs at the same time. The challenge is then to guarantee that the reduced strategy graphs are in fact strategy graphs, i.e., satisfy the required conditions. Also, we, again, make use of the fact that the intruder can generate new constants.

5 Properties of Contract-Signing Protocols

In this section, we formalize fundamental properties of contract-signing protocols, namely effectiveness, fairness, and balance, in our model and explain why these properties are decidable in our framework. The definition of these properties vary slighty from one paper to another (compare, for instance, [6], [17], and [13]). For concreteness, we follow the definitions from [6].

In [6], as in most other works on the formal analysis of contract-signing protocols (see, however, [7]), two-party optimistic contract-signing protocols have been studied. Beside the two parties (the contractual partners), say A and B, who want to sign the contract, a trusted third party T is involved in the protocol, and is consulted in case a problem occurs.

In order to be able to state the properties, we assume in what follows that the protocols are modelled in the following way. If A finishes its protocol, it indicates this by writing the message $\mathsf{ATerminated}$ into the network (i.e., adds it to the intruder's knowledge). In case different sessions of the protocol are considered, this message can contain details of the specific session. Similarly, if A has a valid contract, it writes $\mathsf{AHasValidContract}$ into the network, and if A has an abort token, it writes $\mathsf{AHasAbortToken}$ into the network. In many protocols, such as the ASW protocol (see Section 3 and Appendix A), the message $\mathsf{ATerminated}$ is not needed since if A outputs $\mathsf{AHasValidContract}$ or $\mathsf{AHasAbortToken}$ this also means that A terminated. We make analogous assumptions for B.

We now consider the different security properties. These properties are formulated under the assumption that one of the contractual parties is dishonest and the other party is honest, i.e., follows the protocol. The actions of the dishonest party are performed by the intruder, and hence, are arbitrary. The trusted third party is assumed to be honest. We denote the honest party by H; all properties are formulated for H.

5.1 Effectiveness

The property effectiveness, as formulated in [6], can be divided into two subproperties, soundness and termination.

According to [6], *soundness* means that there is a reachable state (in the transition graph $\mathcal{G}_P$ induced by the protocol P) such that H has completed the protocol and has a valid contract. Hence, in our model soundness asks whether there exists a state in $\mathcal{G}_P$ in which the intruder can derive HTerminated and HHasValidContract. This can be formulated as the reachability property $(\{\mathsf{HTerminated}, \mathsf{HHasValidContract}\}, \emptyset, \emptyset)$, and thus, by Theorem 2 can be decided.

According to [6], *termination* means that for every state q in $\mathcal{G}_P$, there exists a state q' in $\mathcal{G}_P$ reachable from q only by actions of H and T such that H has completed the protocol and either has a valid contract or an abort token. In other words, if a state is reached in which neither H nor T can perform an action, then H must have either a valid contract or an abort token. In our model, this means that for every state in $\mathcal{G}_P$ without outgoing secure channels and ε-transitions of H and T, the intruder can derive either HHasValidContract or HHasAbortToken. Hence, non-termination holds iff (a path in) $\mathcal{G}_P$ has reachability property $(\emptyset, \{\mathsf{HHasValidContract}, \mathsf{HHasAbortToken}\}, \{H, T\})$ or reachability property $(\{\mathsf{HHasValidContract}, \mathsf{HHasAbortToken}\}, \emptyset, \{H, T\})$, and is thus decidable. (The second property is only needed in case one wants to exclude that both HHasValidContract and HHasAbortToken occur.) Recall that the names H and T in the third component of the reachability properties stand for the principals as defined in the protocol. If the protocol description consists of n principals, H and T are some numbers between 1 and n, namely, the numbers used in the labels of the transitions.

A protocol is *effective* for H if it is sound and it terminates for H. By the above, this can be decided based on the reachability problem.

5.2 Fairness

According to [6], a protocol is *fair* for H if for every reachable state q, the following conditions are true:

1. If the intruder (and thus, the dishonest party) has a valid contract in q, then there exists a state q' reachable from q such that H has a valid contract in q'.
2. If H has an abort token in q, then in each state q' reachable from q, the intruder does not have a valid contract.

We will formulate the fact that a protocol is *not* fair (for H) in our model.

The first subproperty fails if there exists a state in $\mathcal{G}_P$ without outgoing transitions and such that H does not have a valid contract but the intruder (and thus, the dishonest party) has.

To formulate the fact that the intruder has a valid constract in our model, we add a principal to the protocol description which on receiving a valid contract outputs IntruderHasValidContract. (What a valid contract is depends on the particular protocol, and thus, the exact formulation

of the new principal depends on the protocol. An example of such a prinicpal is presented in Appendix A.4.)

Now, the fact that the first subproperty is not satisfied means in our model that there exists a state without outgoing transitions and such that the intruder cannot derive HHasValidContract but can derive IntruderHasValidContract. This can precisely be phrased as the reachability property $(\{\mathsf{IntruderHasValidContract}\}, \{\mathsf{HHasValidContract}\}, \text{noTrans})$, and thus is decidable by Theorem 2.

The fact that the second subproperty is not satisfied means that there exists a state in $\mathcal{G}_P$ where H has an abort token *and* the intruder has a valid contract. Again using the new principal, in our model this means that there exists a state in $\mathcal{G}_P$ such that a state can be reached in which the intruder can derive both HHasAbortToken and IntruderHasValidContract. This corresponds to the reachability property $(\{\mathsf{HHasAbortToken}, \mathsf{IntruderHasValidContract}\}, \emptyset, \emptyset)$, and thus, can be checked automatically.

We note that the way fairness is formulated above (and in [6]) does not assume termination. Hence, one could call this version of fairness partial fairness. Fairness defined in other papers often also requires termination, and thus correspondends to the conjunction of partial fairness and termination.

5.3 Balance

According to [6], balance means that there does not exist a state q in $\mathcal{G}_P$ in which the intruder has both the power to abort and the power to complete the protocol. The intruder has the *power to abort* at state q if there exists a strategy graph such that H has an abort token in all leaves of this strategy graph. The intruder has the *power to complete* at state q if there exists a strategy graph such that H has a valid contract in all leaves of this strategy graph. Hence, in our model, a protocol P is *not* balanced if P satisfies the strategy property $((\{\mathsf{HHasAbortToken}\}, \emptyset), (\{\mathsf{HHasValidContract}\}, \emptyset))$. By Theorem 3, this can be decided.

6 Conclusion

In this paper we have shown that effectiveness, fairness, and balance, a branching property of contract-signing protocols, is decidable when there is no bound on the message size for a Dolev-Yao intruder and when there are only a finite number of sessions. This extends known results on the decidability of reachability problems for cryptographich protocols in a natural way. Our approach is fairly generic; it should therefore be a good starting point for analyzing other game-theoretic properties of cryptographic protocols. From a practical point of view, our result may also be a good starting point for developing more precise analyzers for contract-signing protocols.

References

[1] R.M. Amadio, D. Lugiez, and V. Vanackere. On the symbolic reduction of processes with cryptographic functions. *TCS*, 290(1):695–740, 2002.

[2] A. Armando, D. Basin, M. Bouallagui, Y. Chevalier, L. Compagna, S. Mödersheim, M. Rusinowitch, M. Turuani, L. Viganò, and L. Vigneron. The AVISS Security Protocol Analysis Tool. In CAV 2002, volume 2404 of *LNCS*, pages 349–353. Springer, 2002.

[3] N. Asokan, V. Shoup, and M. Waidner. Asynchronous protocols for optimistic fair exchange. In *Proc. of the IEEE Symposium on Research in Security and Privacy*, pages 86–99, 1998.
[4] 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.
[5] M. Boreale. Symbolic trace analysis of cryptographic protocols. In *ICALP 2001*, volume 2076 of *Lecture Notes in Computer Science*, pages 667–681. Springer-Verlag, 2001.
[6] R. Chadha, M.I. Kanovich, and A.Scedrov. Inductive methods and contract-signing protocols. In *CCS 2001*, pages 176–185. ACM Press, 2001.
[7] R. Chadha, S. Kremer, and A. Scedrov. Formal analysis of multi-party contract signing. In *CSFW-17*, pages 266–279. IEEE Computer Society Press, 2004.
[8] R. Chadha, J.C. Mitchell, A. Scedrov, and V. Shmatikov:. Contract signing, optimism, and advantage. In R.M. Amadio and D. Lugiez, editors, *CONCUR 2003 - Concurrency Theory, 14th International Conference*, volume 2761 of *Lecture Notes in Computer Science*, pages 361–377. Springer, 2003.
[9] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR. In *LICS 2003*. IEEE, Computer Society Press, 2003.
[10] H. Comon and V. Shmatikov. Is it possible to decide whether a cryptographic protocol is secure or not? *Journal of Telecommunications and Information Technolog*, 2002.
[11] P. H. Drielsma and S. Mödersheim. The ASW protocol revisited: A unified view. In *Workshop on Automated Reasoning for Security Protocol Analysis (ARSPA)*, 2004.
[12] J.A. Garay, M. Jakobsson, and P. MacKenzie. Abuse-free optimistic contract signing. In *CRYPTO'99*, volume 1666 of *LNCS*, pages 449–466. Springer-Verlag, 1999.
[13] S. Kremer and J.-F. Raskin. Game analysis of abuse-free contract signing. In *Computer Security Foundations Workshop 2002 (CSFW 2002)*. IEEE Computer Society, 2002.
[14] C. Meadows. Formal Methods for Cryptographic Protocol Analysis: Emerging Issues and Trends. *IEEE Journal on Selected Areas in Communication*, 21(1):44–54, January 2003.
[15] J. K. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *CCS 2001*, pages 166–175. ACM Press, 2001.
[16] M. Rusinowitch and M. Turuani. Protocol insecurity with a finite number of sessions, composed keys is np-complete. *Theoretical Computer Science*, 299(1–3):451–475, 2003.
[17] V. Shmatikov and J.C. Mitchell. Finite-state analysis of two contract signing protocols. *Theoretical Computer Science*, 283(2):419–450, 2002.
[18] J. Zhou and D. Gollmann. A fair non-repudiation protocol. In *Proceedings of the IEEE Symposium on Research in S&P*, pages 55–61. IEEE Computer Society Press, 1996.

A Modelling the Asokan-Shoup-Waidner Protocol

An informal description of the protocol as well as the formal model of the originator is in the main part of this paper. We now state the models for the responder and the trusted third party, and formulate security properties.

A.1 The Responder R

The principal R is given by the tree Π_R depicted in Figure 2 where the numbers stand for the principal rules defined below. Rules 1 and 2 describe the exchange protocol, and 3 to 5 the resolve

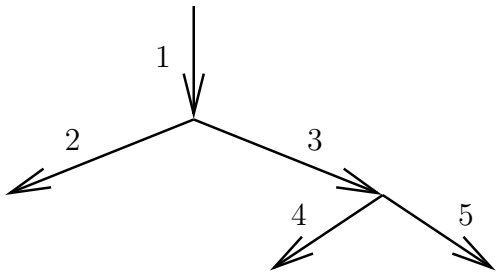

Figure 2: The Responder R

protocol. (Recall that the responder R has no abort protocol). In the exchange protocol, R responds to O's message sent in O's exchange protocol. If after performing rule 1, O does not return a message, R can launch the resolve protocol by sending a resolve request via a secure channel to T (rule 3). As above, T will return one of two possible answers, namely, T will return an abort or resolve token, which are handled by R using rules 4 and 5, respectively.

Descriptions of the principal rules follow.

1. $me_1 \Rightarrow me_2$ where

$$\begin{aligned} me_1 &= \mathsf{sig}[\langle k_O, k_R, k_T, \mathsf{text}, \mathsf{hash}(x)\rangle, k_O], \\ me_2 &= \mathsf{sig}[\langle me_1, \mathsf{hash}(N_R)\rangle, k_R]. \end{aligned}$$

2. $x \Rightarrow \langle N_R, \mathsf{RasValidContract}\rangle$.[3]
3. $\varepsilon \Rightarrow \mathsf{sc}(R, T, \langle me_1, me_2\rangle)$.
4. $\mathsf{sc}(T, R, \mathsf{sig}[\langle \mathsf{aborted}, ma_1\rangle, k_T]) \Rightarrow \mathsf{RHasAbortToken}$ where

$$ma_1 = \mathsf{sig}[\langle \mathsf{aborted}, me_1\rangle, k_T].$$

5. $\mathsf{sc}(T, R, \mathsf{sig}[\langle me_1, me_2\rangle, k_T]) \Rightarrow \mathsf{RasValidContract}$.

A.2 The Trusted Third Party T

The trusted third party handles resolve and abort requests. Each such request involves two steps: receiving the request and sending out a response (one rule). The different requests depend, however, on each other. For instance, if the TTP receives an abort request for a protocol run between two principals, then, after it has replied to it, the TTP will only allow a resolve request by the other principal for the same protocol run. Moreover, this resolve request will be acknowledged by sending an abort token. Similarly, if the TTP receives a resolve request, then, after it has replied to it, it will only allow a request by the other principal (abort or resolve), but no further request by the same principal. Moreover, this request will be acknowledged by a resolve token.

When, in the first situation, we assume that O is the principal sending the first request (abort), we can model the behaviour of the TTP by the following two rules, where the tree is the one involving the edges labeled 1 and 2 in Figure 3.

[3]Even though O only expects to obtain N_R, R also indicates that it has a valid contract. Since this message is sent to the intruder, the intruder could extract N_R from it and send it to O. In any case, the intruder decides what is actually sent to O.

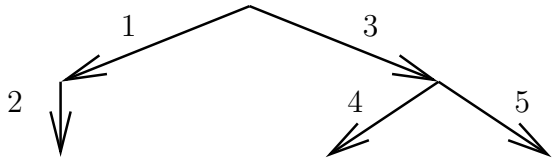

Figure 3: Partial Tree of the Trusted Third Party T

1. $\mathsf{sc}(O, T, ma_1) \Rightarrow \mathsf{sc}(T, O, ma_2)$ where

$$\begin{aligned} ma_1 &= \mathsf{sig}[\langle \mathsf{aborted}, me_1 \rangle, k_O], \\ me_1 &= \mathsf{sig}[\langle k_O, k_R, k_T, x_1, x_2 \rangle, k_O], \\ ma_2 &= \mathsf{sig}[\langle \mathsf{aborted}, ma_1 \rangle, k_T]. \end{aligned}$$

2. $\mathsf{sc}(R, T, \langle me_1, \mathsf{sig}[\langle me_1, \_ \rangle, R] \rangle) \Rightarrow \mathsf{sc}(T, R, ma_1)$ with the same terms as above.

When, in the other situation, we assume that R is the principal sending the first request (resume), we can model the behaviour of the TTP by the following three rules, where the tree is the one involving the edges labeled 3, 4, and 5 in Figure 3.

3. $\mathsf{sc}(R, T, mr_1) \Rightarrow \mathsf{sc}(T, R, mr_2)$ where

$$\begin{aligned} mr_1 &= \langle me_1, me_2 \rangle, \\ me_1 &= \mathsf{sig}[\langle k_R, k_O, k_T, x_1, x_2 \rangle, k_R], \\ me_2 &= \mathsf{sig}[\langle me_1, x_3 \rangle, k_R], \\ mr_2 &= \mathsf{sig}[\langle me_1, me_2 \rangle, k_T]. \end{aligned}$$

4. $\mathsf{sc}(O, T, \mathsf{sig}[\langle \mathsf{aborted}, me_1 \rangle, O]) \Rightarrow \mathsf{sc}(T, O, mr_2)$.
5. $\mathsf{sc}(O, T, mr_1) \Rightarrow \mathsf{sc}(T, O, mr_2)$.

The complete model of T is now obtained by joining the two trees (as done in Figure 3) and adding a copy of the resulting tree, but with the roles of O and R exchanged and the labels and variables renamed.

This completes the description of our model of the ASW protocol.

A.3 Modelling issues

In the following, we discuss some of the modelling issues.

Authentication through secure channel. From the rules we use for T it is clear that in our model—which follows [6] in this respect—the TTP makes use of the parameters of its secure channels in a crucial way: T only accepts a request by O if this request is received on the secure channel from O. In other words, T uses the channel for authentication purposes; it relies on the secure channel.

A different approach would be to stipulate that the protocol itself should provide evidence for authenticity and not rely on the secure channel. We can change our model accordingly. For instance, rule 1 of the intruder would then split into three different rules, each one of the form

$$\mathsf{sc}(z, T, ma_1) \Rightarrow \mathsf{sc}(T, z, ma_2)$$

where

$$\begin{aligned} ma_1 &= \mathsf{sig}[\langle \mathsf{aborted}, me_1 \rangle, y], \\ me_1 &= \mathsf{sig}[\langle y, x_1, T, x_2, x_3 \rangle, y], \\ ma_2 &= \mathsf{sig}[\langle \mathsf{aborted}, ma_1 \rangle, k_T]. \end{aligned}$$

To obtain the three rules one would have to replace z by any of the three possible principals, namely O, R, and T.

Resources of the TTP. In the above model, T can handle only one protocol instance (session). In case one wants it to handle more than one, one would have to have a model of the TTP that incorporates several copies of T from above. Clearly, the TTP would have to keep track of the abort tokens it has sent out, because it should not sent out a resolve token for the same session it has issued an abort token earlier. In [6], this kind of book keeping is realized by a slight extension of the protocol: each message in the protocol is augmented by a unique session id. In our framework, this could be mimicked by adding a unique session constant to each message.

A.4 Formulating Security Goals of the ASW Protocol

In this section, we formulate the security goals of the ASW protocol as modeled in the previous sections. All of these security goals can be decided as shown in Section 5. For concretness, we will consider the protocol $P = (\{\Pi_O, \Pi_T\}, \{R, k_R, {k_R}^{-1}, k_O, k_T, \mathsf{text}, \mathsf{aborted}\})$ with $R \in \mathcal{N}$, $k_R, {k_R}^{-1}, k_O, k_T \in \mathcal{K}$, and $\mathsf{text}, \mathsf{aborted} \in \mathcal{A}$. That is, we assume O and T to be honest, i.e., they follow their respective protocols. However, R is corrupted, and thus, R's actions are performed by the intruder, and hence, are arbitrary. Note that the intruder can impersonate R since he has R's private key ${k_R}^{-1}$ and also the address R used as a sender for communication on a secure channel.

The security goals are formulated for honest O. This is done exactly as described in Section 5. Since if O outputs $\mathsf{OHasValidContract}$ or $\mathsf{OHasAbortToken}$, then this implies that O has terminated, the message $\mathsf{OTerminated}$ is not needed. The additional principal needed to state (partial) fairness for O is defined as follows. Recall that the purpose of this principal is to return $\mathsf{IntruderHasValidContract}$ in case the intruder can derive a valid contract (for O and the agreed upon contractual text). Formally, the tree defining the principal has two edges, which both originate from the root. The principal rules labeling these edges are defined as follows:

1. $valid_1 \Rightarrow \mathsf{IntruderHasValidContract}$ where

$$\begin{aligned} valid_1 &= \langle me_1, x, me_2, y \rangle, \\ me_1 &= \mathsf{sig}[\langle k_O, k_R, k_T, \mathsf{text}, \mathsf{hash}(x) \rangle, k_O], \\ me_2 &= \mathsf{sig}[\langle me_1, \mathsf{hash}(y) \rangle, k_R]. \end{aligned}$$

2. $valid_2 \Rightarrow \mathsf{IntruderHasValidContract}$ where

$$\begin{aligned} valid_2 &= \mathsf{sig}[\langle me_1, me_2 \rangle, k_T], \\ me_1 &= \mathsf{sig}[\langle k_O, k_R, k_T, \mathsf{text}, y \rangle, k_O], \\ me_2 &= \mathsf{sig}[\langle me_1, z \rangle, k_R]. \end{aligned}$$

B Deciding Reachability and Strategy Properties

In this section, we prove Theorem 2 and 3. For this purpose, we show some important properties about the replacement of terms (Appendix B.1) and the derivation of messages by the intruder (Appendix B.2) and then consider the path and strategy problem in Appendix B.3 and B.4.

By $\mathrm{Sub}(t)$ we denote the set of subterms of t. We define $\mathrm{Sub}(E) = \bigcup\{\mathrm{Sub}(t) \mid t \in E\}$. We define $\mathrm{Sub}(P)$ to be the set of subterms occurring in P and also add the sets $\mathcal{A}$, $\mathcal{K}$, and $\mathcal{N}$.

B.1 Replacements

Let τ and τ' be terms. For a term t we denote by $t_{|\tau\to\tau'}$ the term obtained from t by replacing all occurrences of τ in t by τ'. For a set of terms E we define $E_{|\tau\to\tau'} = \{t_{|\tau\to\tau'} \mid t \in E\}$. For a substitution σ we define $\sigma_{|\tau\to\tau'}$ to be the substitution with $\sigma_{|\tau\to\tau'}(x) = \sigma(x)_{|\tau\to\tau'}$ for all variables $x \in \mathsf{dom}(\sigma)$. For a state $p = ((\Pi_1, \ldots, \Pi_n), \sigma, \mathcal{I}, \mathcal{S})$ we define

$$p_{|\tau\to\tau'} = ((\Pi_1, \ldots, \Pi_n), \sigma_{|\tau\to\tau'}, \mathcal{I}_{|\tau\to\tau'}, \mathcal{S}_{|\tau\to\tau'}).$$

For a substitution σ and terms t, t', we say that t is a *σ-match* of t' ($t \sqsubseteq_\sigma t'$) if t is not a variable and $t\sigma = t'$ (this notions was first introduced in [16]).

Obviously, we have the following lemma.

Lemma 4. *Let σ be a substitution, E be a set of terms, and τ be a message such that $t \not\sqsubseteq_\sigma \tau$ for all $t \in Sub(E)$. Then, for every $a \in \mathcal{A} \cup \mathcal{K} \cup \mathcal{A}_I$ and for all $t \in Sub(E)$, we have $t(\sigma_{|\tau\to a}) = (t\sigma)_{|\tau\to a}$.*

B.2 Derivations

We now prove some important properties about the derivation of messages by the intruder.

First, we characterize the set $d(E)$ in terms of what we call intruder rules. The rules include the same as those introduced in [16]. In addition, we need rules for hasing, signatures, private contract signatures, and generating new atoms. In what follows, we often write E, m and m, m' instead of $E \cup \{m\}$ and $\{m, m'\}$, respectively.

An intruder rule L is of the form $E \to m$ where E is a finite set of messages and m is a message. A rule of this form is also called m-rule since m is generated. Given a set E', L can be *applied to* E' if $E \subseteq E'$. The rule L induces a binary relation $\to_L$ on finite sets of messages: $\to_L = \{(E', E' \cup \{m\}) \mid L \text{ can be applied to } E'\}$. If $\mathcal{L}$ is a set of intruder rules, then $\to_{\mathcal{L}} = \bigcup_{L\in\mathcal{L}} \to_L$. For a binary relation $\to$ we write $E \to E'$ instead of $\to (E, E')$. The reflexive and transitive closure of $\to$ is denoted by $\to^*$.

To describe $d(E)$, we consider the following set of intruder rules. In what follows, the notion "intruder rule" will always refers to the rules introduced below. This set is partitioned into decomposition and composition rules. Accordingly, we call a rule decomposition and composition rule, respectively.

Decomposition rules are of one of the following forms, where m and m' are some messages and $k \in \mathcal{K}$ (and thus, $k^{-1} \in \mathcal{K}$):

1. $\langle m, m'\rangle \to m$ and $\langle m, m'\rangle \to m'$.
2. $\{m\}^s_{m'}, m' \to m$.
3. $\{m\}^a_k, k^{-1} \to m$.

Composition rules are of one of the following forms, where m, m' are some messages, $k, k_0, k_1, k_2 \in \mathcal{K}$, $n, n' \in \mathcal{N}$, $a_I \in \mathcal{A}_I$:

1. $m, m' \to \langle m, m' \rangle$.
2. $m, m' \to \{m\}^s_{m'}$.
3. $m, k \to \{m\}^a_k$.
4. $m \to \mathsf{hash}(m)$.
5. $m, k^{-1} \to \mathsf{sig}_k(m)$.
6. $m, {k_0}^{-1}, k_1, k_2 \to \mathsf{PCS}_{k_0}(m, k_1, k_2)$.
7. $m, k_0, {k_1}^{-1}, k_2 \to \mathsf{PCS}^f_{k_0}(m, k_1, k_2)$.
8. $\mathsf{PCS}_{k_0}(m, k_1, k_2), {k_0}^{-1} \to \mathsf{ssig}_{k_0}(m, k_1, k_2)$ and $\mathsf{PCS}_{k_0}(m, k_1, k_2), {k_2}^{-1} \to \mathsf{tsig}_{k_0}(m, k_1, k_2)$.
9. $m, n \to \mathsf{sc}(n, n', m)$ (for some $n' \in \mathcal{N}$).
10. $\to a_I$.

If $\mathcal{L}$ denotes the set of all (composition and decomposition) rules, then we obviously have that $d(E) = \bigcup\{E' \mid E \to^*_{\mathcal{L}} E'\}$.

A *derivation* is of the form $E_0 \to_{L_0} E_1 \to_{L_1} E_2 \to_{L_2} \cdots \to_{L_{n-1}} E_n$ where $E_i \to_{L_i} E_{i+1}$ for every i. We call n the length of the derivation. We know that for every $m \in d(E)$ there exists n, intruder rules $L_0, \ldots, L_{n-1}$, and sets $E_0, \ldots, E_n$ such that $E_0 = E$, $m \in E_n$, and there is a derivation from E_0 to E_n as above. We call such a derivation a derivation for m of length n. The derivation is *minimal* if no step can be removed such that the resulting sequence is still a derivation for m. Clearly, for every $m \in d(E)$ there exists a minimal derivation. We write $m \in d^c(E)$ if there exists a minimal derivation of m where the last rule is a composition rule. (Note that not for every $m \in d(E)$ such a derivation needs to exist.) We say that $m \in d_n(E)$ if there exists a (minimal) derivation of m from E of length $\leq n$. We write $m \in d^c_n(E)$ to say that $m \in d^c(E)$ and the minimal derivation of m is of length $\leq n$.

It has been proved in [16] for encryption that a a minimal derivation of m from E only contains subterms of E and m, i.e., every rule only produces a subterm of E or m. This easily carries over to our setting as well if with a public (private) key k, the set of subterms of a term also contains the private (public) key k^{-1}, and with a term $\mathsf{ssig}_{k_0}(m, k_1, k_2)$ or $\mathsf{tsig}_{k_0}(m, k_1, k_2)$ the set of subterms also contains $\mathsf{PCS}_{k_0}(m, k_1, k_2)$. Extending the set of subterms in this way is necessary since, for instance, to construct $\mathsf{ssig}_{k_0}(m, k_1, k_2)$ it may be necessary to first construct $\mathsf{PCS}_{k_0}(m, k_1, k_2)$ and then convert this message into $\mathsf{ssig}_{k_0}(m, k_1, k_2)$. Now, given that to derive m from E only subterms of m and E have to be considered, it is not hard to see that $m \in d(E)$ can be decided in polynomial time: One simply computes the closure of all subterms of m and E—of which there are only polynomially many in the size of E and m—derivable from E (see, e.g., [9]). Hence, with $\textsc{Derive} = \{(E, m) \mid m \in d(E)\}$ where E and m are given as DAGs we have:

Lemma 5. $\textsc{Derive}$ *can be decided in polynomial time.*

We now study which messages can be derived from a set of messages if certain terms are replaced by other terms.

Lemma 6. *Let E be a set of messages and τ, τ' be a message. Then, $\tau \in d^c(E \setminus \{\tau\})$ implies that $\tau \in d(E_{|\tau \to \tau'})$.*

Proof. By induction on n we show that $\tau \in d^c_n(E \setminus \{\tau\})$ implies $\tau \in d(E_{|\tau \to \tau'})$.

$\mathbf{n = 1}$: We have a derivation of the form $E \setminus \{\tau\} \to_L E \setminus \{\tau\}, \tau$ where L is a composition rule. If $L = m, m' \to \langle m, m' \rangle (= \tau)$, then of course $m, m' \in E_{|\tau \to \tau'}$, and hence, $\tau \in d(E_{|\tau \to \tau'})$. The

argument for the other composition rules, except for $\rightarrow a_I$, is the same. If $\tau \in \mathcal{A}_I$, the statement is obvious.

$\mathbf{n} \rightarrow n+1$: Let $E_0 = E \setminus \{\tau\}$ and $E_0 \rightarrow_{L_0} E_1 \rightarrow_{L_1} \cdots \rightarrow_{L_n} E_{n+1}$ be a minimal derivation of τ such that L_n is a composition rule. We know that $\tau \notin E_1$ (since otherwise the derivation would not be minimal). Thus, $\tau \in d_n^c(E_1 \setminus \{\tau\})$, and by induction, it follows $\tau \in d(E_{1|\tau\rightarrow\tau'})$. Assume that L_0 is an m-rule. Hence, $E_1 = E_0 \cup \{m\}$, and thus, $E_{1|\tau\rightarrow\tau'} = E_{0|\tau\rightarrow\tau'} \cup \{m_{|\tau\rightarrow\tau'}\}$. If we can show that $m_{|\tau\rightarrow\tau'} \in d(E_{0|\tau\rightarrow\tau'})$, then we immediately obtain that $\tau \in d(E_{0|\tau\rightarrow\tau'})$, which concludes our proof. To show that $m_{|\tau\rightarrow\tau'} \in d(E_{0|\tau\rightarrow\tau'})$, we distinguish different cases. Note that $m \neq \tau$ and $\tau \notin E_0$.

1. Assume that $L_0 = m', m'' \rightarrow \langle m', m''\rangle$. Then, ${m'}_{|\tau\rightarrow\tau'}, {m''}_{|\tau\rightarrow\tau'} \in E_{0|\tau\rightarrow\tau'}$, and thus, $m_{|\tau\rightarrow\tau'} = \langle {m'}_{|\tau\rightarrow\tau'}, {m''}_{|\tau\rightarrow\tau'}\rangle \in d(E_{0|\tau\rightarrow\tau'})$. A similar argument works for the other composition rules. For $\mathsf{PCS}_{k_0}(m, k_1, k_2), {k_0}^{-1} \rightarrow \mathsf{ssig}_{k_0}(m, k_1, k_2)$ and $\mathsf{PCS}_{k_0}(m, k_1, k_2), {k_2}^{-1} \rightarrow \mathsf{tsig}_{k_0}(m, k_1, k_2)$ we use that $\tau \neq \mathsf{PCS}_{k_0}(m, k_1, k_2)$. For $\rightarrow a_I$, we use that $\tau \neq a_I$.
2. Assume that $L_0 = \langle m, m'\rangle \rightarrow m$. We have $\langle m, m'\rangle_{|\tau\rightarrow\tau'} = \langle m_{|\tau\rightarrow\tau'}, {m'}_{|\tau\rightarrow\tau'}\rangle$ because $\langle m, m'\rangle \neq \tau$, . Hence, $m_{|\tau\rightarrow\tau'} \in d(E_{0|\tau\rightarrow\tau'})$. The argument for the other decomposition rules is similar. □

We use this lemma to prove:

Lemma 7. *Let E be a set of messages and τ, τ' be messages. Then, $\tau \in d^c(E \setminus \{\tau\})$ implies $d(E)_{|\tau\rightarrow\tau'} \subseteq d(E_{|\tau\rightarrow\tau'} \cup \{\tau'\})$.*

Proof. By induction on n we show that with $\tau \in d^c(E \setminus \{\tau\})$ it follows for every $m \in d_n(E)$ that $m_{|\tau\rightarrow\tau'} \in d(E_{|\tau\rightarrow\tau'} \cup \{\tau'\})$. For $n = 0$ this is obvious. For the induction step $(n \rightarrow n+1)$ assume that $E = E_0 \rightarrow_{L_0} E_1 \rightarrow_{L_1} \cdots \rightarrow_{L_n} E_{n+1}$ is a minimal derivation of m. It is easy to see that $m \in d_n(E_1)$ and $\tau \in d^c(E_1 \setminus \{\tau\})$. Induction yields that $m_{|\tau\rightarrow\tau'} \in d(E_{1|\tau\rightarrow\tau'} \cup \{\tau'\})$. Assume that L_0 is an m'-rule. Hence, $E_1 = E_0 \cup \{m'\}$ and $E_{1|\tau\rightarrow\tau'} = E_{0|\tau\rightarrow\tau'} \cup \{{m'}_{|\tau\rightarrow\tau'}\}$. If we can show that ${m'}_{|\tau\rightarrow\tau'} \in d(E_{0|\tau\rightarrow\tau'} \cup \{\tau'\})$, we are done. We distinguish different cases.

1. Assume that $L_0 = m'', m''' \rightarrow \langle m'', m'''\rangle (= m')$. If $\langle m'', m'''\rangle = \tau$, then we have of course $\langle m'', m'''\rangle_{|\tau\rightarrow\tau'} \in d(E_{0|\tau\rightarrow\tau'} \cup \{\tau'\})$. Otherwise, $\langle m'', m'''\rangle_{|\tau\rightarrow\tau'} = \langle {m''}_{|\tau\rightarrow\tau'}, {m'''}_{|\tau\rightarrow\tau'}\rangle$, and again, $\langle {m''}_{|\tau\rightarrow\tau'}, {m'''}_{|\tau\rightarrow\tau'}\rangle \in d(E_{0|\tau\rightarrow\tau'} \cup \{\tau'\})$. If $L_0 =\!\!\rightarrow a_I$, the argument is obvious. Analogously, one can show the statment for all other composition rules except for the rule $\mathsf{PCS}_{k_0}(m'', k_1, k_2), {k_0}^{-1} \rightarrow \mathsf{ssig}_{k_0}(m'', k_1, k_2)$ and $\mathsf{PCS}_{k_0}(m'', k_1, k_2), {k_2}^{-1} \rightarrow \mathsf{tsig}_{k_0}(m'', k_1, k_2)$ in case $\mathsf{PCS}_{k_0}(m'', k_1, k_2) = \tau$. However, by assumption, and Lemma 6, we know that $\tau \in d(E_{0|\tau\rightarrow\tau'})$. Thus, $m' = \mathsf{ssig}_{k_0}(m'', k_1, k_2) \in d(E_{0|\tau\rightarrow\tau'})$ $(m' = \mathsf{tsig}_{k_0}(m'', k_1, k_2) \in d(E_{0|\tau\rightarrow\tau'}))$. Since $\tau \neq m'$, we have ${m'}_{|\tau\rightarrow\tau'} = m'$. Hence, ${m'}_{|\tau\rightarrow\tau'} \in d(E_{0|\tau\rightarrow\tau'} \cup \{\tau'\})$.
2. Assume that $L_0 = \langle m', m''\rangle \rightarrow m'$. If $\langle m', m''\rangle = \tau$, then $\tau \in d(E_{0|\tau\rightarrow\tau'})$ by Lemma 6. Hence, ${m'}_{|\tau\rightarrow\tau'} = m' \in d(E_{0|\tau\rightarrow\tau'})$. If $\langle m', m''\rangle \neq \tau$, then $\langle {m'}_{|\tau\rightarrow\tau'}, {m''}_{|\tau\rightarrow\tau'}\rangle \in E_{0|\tau\rightarrow\tau'}$, and thus, ${m'}_{|\tau\rightarrow\tau'} \in d(E_{0|\tau\rightarrow\tau'})$. The argument for the other decomposition rules is similar. □

B.3 Deciding the Reachability Problem

The key to show that the reachability problem is decidable is the following lemma. It says that given a rooted path π in the transition graph $\mathcal{G}_P$ such that the substitution σ in the final state of this path (note that this substitution extends all other substitutions in states of the path) is not built from subterms of the protocol, one can construct another rooted path π' in $\mathcal{G}_P$ which has the same properties as π w.r.t. reachability properties as defined in Section 4.1 and in which the substitution

is build only from subterms of the protocol. Formally, a subsitution σ is built from subterms of the protocol if for every x there exists $t \in \mathrm{Sub}(P) \cup \mathcal{A}_I$ such that $t \sqsubseteq_\sigma \sigma(x)$. This lemma extends a lemma proved in [16], which was restricted to a setting without (private contract) signatures, secure channels, and the ability of the intruder to generate new atoms, and more importantly, in [16] only secrecy properties rather than the more general reachability properties have been considered.

Our lemma is stated slightly more generally than explained above. We show that for every message τ such that $t \not\sqsubseteq_\sigma \tau$ for every $t \in \mathrm{Sub}(P) \cup \mathcal{A}_I$, there exists a path with the same reachability properties as the previous path where every occurrence of τ is replaced by a *new* constant, and thus, in this path the substitutions do not contain τ. This more general version of the lemma is also needed in the following section. Note that if a subsitution is not built from subterms of the protocol, i.e., there exists x such that $t \not\sqsubseteq_\sigma \sigma(x)$ for every $t \in \mathrm{Sub}(P) \cup \mathcal{A}_I$, then using the lemma and setting $\tau = \sigma(x)$, we can find a path with a substitution where $t \sqsubseteq_\sigma \sigma(x)$ for some $t \in \mathcal{A}_I$. Hence, by repeated application of the lemma, we can turn the substitution into one that is built from subterms of the protocol.

Lemma 8. *Let*

$$\pi = p_0 \stackrel{\lambda_0}{\to} p_1 \stackrel{\lambda_1}{\to} p_2 \stackrel{\lambda_2}{\to} \cdots \stackrel{\lambda_{l-1}}{\to} p_l$$

be a rooted path in $\mathcal{G}_P$ where

$$p_j = ((\Pi_1^j, \ldots, \Pi_n^j), \sigma_j, \mathcal{I}_j, \mathcal{S}_j).$$

Let τ be some message such that $t \not\sqsubseteq_{\sigma_l} \tau$ for all $t \in Sub(P) \cup \mathcal{A}_I$. Furthermore, let $a_I \in \mathcal{A}_I$ be a constant that does not occur anywhere in π and P, and define

$$\pi' = p'_0 \stackrel{\lambda'_0}{\to} p'_1 \stackrel{\lambda'_1}{\to} p'_2 \stackrel{\lambda'_2}{\to} \cdots \stackrel{\lambda'_{l-1}}{\to} p'_l$$

where

$$p'_j = p_{j|\tau \to a_I}$$

and

$$\lambda'_j = \begin{cases} \lambda_j & \text{if } \lambda_j \text{ is an } \epsilon\text{-transition,} \\ (i, m', I) & \text{if } \lambda_j = (i, m, I) \text{ with } m' = m_{|\tau \to a_I}, \\ (i, m', \mathsf{sc}) & \text{if } \lambda_j = (i, m, \mathsf{sc}) \text{ with } m' = m_{|\tau \to a_I}. \end{cases}$$

Then, all of the following is true:

1) *π' is a rooted path in $\mathcal{G}_P$.*
2) *For all atoms $c \in \mathcal{A} \cup \mathcal{N} \cup \mathcal{K}$ we have $c \in d(\mathcal{I}_l)$ iff $c \in d(\mathcal{I}'_l)$.*
3) *If there is no intruder transition in $\mathcal{G}_P$ (for principal i) originating from p_l, then there is no intruder transition in $\mathcal{G}_P$ (for principal i) originating from p'_l.*
4) *If there is no secure channel transition in $\mathcal{G}_P$ (for principal i) originating from p_l, then there is no secure channel transition in $\mathcal{G}_P$ (for principal i) originating from p'_l.*
5) *If there is no ε-transition in $\mathcal{G}_P$ (for principal i) originating from p_l, then there is no ε-transition in $\mathcal{G}_P$ (for principal i) originating from p'_l.*

Proof. First, assume that τ does not occur as a subterm anywhere in π. Then, $\pi' = \pi$ and nothing is to show. In what follows, we show the properties claimed for π' under the assumption that τ occurs in π.

1) We call (*) the property of τ that $t \not\sqsubseteq_{\sigma_l} \tau$ for all $t \in \mathrm{Sub}(P) \cup \mathcal{A}_I$. Let

$$R_0 \Rightarrow S_0, \ldots, R_{l-1} \Rightarrow S_{l-1}$$

be the sequence of principal rules applied in the transitions of the path. We know that τ is not a subterm of $\mathcal{I}_0$ or some of the principal rules of the principals because of (*). But then, there must exist a j such that τ is a subterm of $R_j\sigma_l$ or $S_j\sigma_l$. Because of (*) it follows that there exists x in the domain of σ_l (and in R_j or S_j) such that τ is a subterm of $\sigma_l(x)$. By definition of principals, if a variable occurs on the right-hand side of some principal rule, then it also occurs on the left-hand of some preceeding rule. Therefore, there exists i such that x occurs in R_i. In particular, there exists q such that τ is a subterm of $R_q\sigma_l (= R_q\sigma_{q+1})$. We choose q minimal with this property, i.e., τ is not a subterm in $R_j\sigma_l$ for some $j < q$. We prove the following statements:

a) λ_q is an intruder transition.

Suppose λ_q is a secure channel transition. Then there is $u < q$ such that $R_q\sigma_{q+1} = S_u\sigma_u = \mathsf{sc}(n_1, n_2, m) \in S_q$. It follows that τ is a subterm of $S_u\sigma_u$. Because of (*) it follows that τ occurs as a subterm in $\sigma_u(y)$ for some variable y. By the definition of principals, it follows that y occurs in some R_i for $i \leq u$. Thus, τ is a subterm of $R_i\sigma_u$ in contradiction to the definition of q. Hence, λ_q cannot be a secure channel transition.
Trivially, the transition λ_q can not be an ε-transition since in this case $R_q = \varepsilon$.

b) $\tau \notin \mathrm{Sub}(\mathcal{I}_q)$.
The argument is similar to the above.

c) $\tau \in d^c(\mathcal{I}_q)$.
We first show that $\tau \in d(\mathcal{I}_q)$. We have $R_q\sigma_{q+1} \in d(\mathcal{I}_q)$, because λ_q is an intruder transition. Let $E_0 = \mathcal{I}_q$ and

$$E_0 \rightarrow_{L_0} E_1 \rightarrow_{L_1} \cdots \rightarrow_{L_{n-1}} E_n.$$

be a minimal derivation of $R_q\sigma_{q+1}$ from E_0. In particular, $R_q\sigma_{q+1} \in E_n$. We want to show that $\tau \in E_n$. Suppose $\tau \notin E_n$. There exists a minimal $j \geq 0$ such that $\tau \in \mathrm{Sub}(E_j)$ since $\tau \in \mathrm{Sub}(R_q\sigma_{q+1})$. We know that $j > 0$ since $\tau \notin \mathrm{Sub}(\mathcal{I}_q)$. We distinguish between the following cases:

- L_{j-1} is a composition rule which generates m, and thus, $\tau \in \mathrm{Sub}(m)$ and $\tau \neq m$. By the definition of composition rules, it is easy to check that then $\tau \in \mathrm{Sub}(E_{j-1})$, in contradiction to the choice of j.
- L_{j-1} is a decomposition rule which generates m, and thus, $\tau \in \mathrm{Sub}(m)$ and $\tau \neq m$. Then, by the definition of decomposition rules, we can conclude that $\tau \in \mathrm{Sub}(E_{j-1})$, in contradiction to the choice of j.

This implies that $\tau \in d(\mathcal{I}_q)$. Furthermore, using that $\tau \notin \mathrm{Sub}(\mathcal{I}_q)$ we can conclude that $\tau \in d^c(\mathcal{I}_q)$.

d) $\pi' = p'_0 \stackrel{\lambda'_0}{\rightarrow} p'_1 \stackrel{\lambda'_1}{\rightarrow} p'_2 \stackrel{\lambda'_2}{\rightarrow} \cdots \stackrel{\lambda'_{l-1}}{\rightarrow} p'_l$ is a path in $\mathcal{G}_P$. We have to show that all steps in π' are indeed transitions. We show the following statement by induction on j: For all $0 \leq j \leq l-1$ we have that p'_j and p'_{j+1} are states in $\mathcal{G}_P$ and λ'_j is a transition between them.
$\mathbf{j = 0}$: Of course we have $p'_0 = p_0$. If $q > 0$ then $p'_1 = p_1$ and $\lambda'_1 = \lambda_1$, and thus, we are done. If $q = 0$, we know that λ_0 is an intruder transition and

$$R_0\sigma_1 \in d(\mathcal{I}_0),$$

and we know by Lemma 4 that

$$R_0\sigma'_1 = R_0(\sigma_{1|\tau\to a_I}) = (R_0\sigma_1)_{|\tau\to a_I}$$

and we know $K'_0 = K_{0|\tau\to a_I} = K_0$ because $\tau \notin \mathrm{Sub}(\mathcal{I}_0)$.
Therefore, $R_0\sigma'_1 \in d(\mathcal{I}'_0)$ by Lemma 7.

$\mathbf{j} \to \mathbf{j+1}$: We distinguish different cases depending on the type of λ_{j+1}.

- λ_{j+1} is an ε-transition: If $\lambda_{j+1} = \xrightarrow{i}$ then there is an applicable rule in Π_i^{j+1} of the form $\varepsilon \Rightarrow S_{j+1}$. The principals in p'_{j+1} are the same as in p_{j+1} so the same ϵ-rule $\epsilon \Rightarrow S_{j+1}$ is applicable in p'_{j+1}. The substitution in p_{j+2} is the same as in p_{j+1} so the substitutions in p'_{j+2} and p'_{j+1} are the same, too. Since by Lemma 4 we have $S_{j+1}\sigma'_{j+2} = S_{j+1}(\sigma_{j+1|\tau\to a_I}) = (S_{j+1}\sigma_{j+1})_{|\tau\to a_I}$ the application of $\epsilon \Rightarrow S_{j+1}$ leads to p'_{j+2}.
- λ_{j+1} is an intruder transition: We know that $R_{j+1}\sigma_{j+2} \in d(\mathcal{I}_{j+1})$ and we know that

  $$R_{j+1}\sigma'_{j+2} = R_{j+1}(\sigma_{j+2|\tau\to a_I}) = (R_{j+1}\sigma_{j+2})_{|\tau\to a_I}$$

  and by definition $\mathcal{I}'_{j+1} = \mathcal{I}_{j+1|\tau\to a_I}$. Therefore by Lemma 7 we have $R_{j+1}\sigma'_{j+2} \in d(\mathcal{I}'_{j+1})$.
- λ_{j+1} is a secure channel transition: Let $m \in \mathcal{S}_{j+1}$ such that $m = R_{j+1}\sigma_{j+2}$. By definition $m_{|\tau\to a_I} \in \mathcal{S}'_{j+1}$. We have

  $$m_{|\tau\to a_I} = R_{j+1}\sigma_{j+2|\tau\to a_I} = R_{j+1}(\sigma_{j+2|\tau\to a_I})$$

  So λ'_{j+1} leads from p'_{j+1} to p'_{j+2}.

2) The implication from left to right, follows from Lemma 7 with $E = \mathcal{I}_l$ and $\tau' = a_I$. The implication in the other direction, follows from Lemma 7 if we set E to be $\mathcal{I}'_l$, τ (from Lemma 7) to be a_I, and τ' to be τ (from the lemma proved here).

3) We show that there is an intruder transition originating from p_l (for principal i) if there is an intruder transition originating from p'_l (for principal i).

Let $p' \in \mathcal{G}_P$ such that $p'_l \xrightarrow{(i,m',I)} p'$ is a transition in $\mathcal{G}_P$, where $p' = (\{\Pi'\}, \sigma', \mathcal{I}', \mathcal{S}')$. Let $R \Rightarrow S$ be the principle rule for this transition. Then we know that σ' is an extension of $\sigma_{l|\tau\to a_I}$ to $\mathsf{dom}(\sigma_{l|\tau\to a_I}) \cup \mathcal{V}(R)$ such that $R\sigma' \in d(\mathcal{I}'_l)$. Now we show that there is an intruder transition from p_l in $\mathcal{G}_P$. Let $\sigma = \sigma'_{|a_I\to\tau}$. We only need to show that σ is an extension of σ_l and $R\sigma \in d(\mathcal{I}_l)$. Since a_I does not occur in σ_l the substitution σ is an extension of σ_l and we know by Lemma 7 that

$$R\sigma = R(\sigma'_{|a_I\to\tau}) = (R\sigma')_{|a_I\to\tau} \in d(\mathcal{I}_{l|\tau\to a_I})_{|a_I\to\tau} \subseteq d((\mathcal{I}_{l|\tau\to a_I})_{|a_I\to\tau} \cup \{\tau\}) = d(\mathcal{I}_l \cup \{\tau\}) = d(\mathcal{I}_l).$$

This shows that there is an outgoing intruder transition from p_l (for principal i) in $\mathcal{G}_P$.

4) We show that there is a secure channel transition in $\mathcal{G}_P$ (for principal i) originating from p_l if there is a secure channel transition in $\mathcal{G}_P$ (for principal i) originating from p'_l.

Let $p' \in \mathcal{G}_P$ such that $p'_l \xrightarrow{(i,m',\mathsf{sc})} p'$ is a transition in $\mathcal{G}_P$, where $p' = (\{\Pi'\}, \sigma', \mathcal{I}', \mathcal{S}')$. Let $R \Rightarrow S$ be the principle rule for this transition. Then we know that σ' is an extension of $\sigma_{l|\tau\to a_I}$

to $\mathsf{dom}(\sigma_{l|\tau\to a_I}) \cup \mathcal{V}(R)$ such that $m' \in \mathcal{S}'_l$. Now we show that there is a secure channel transition from p_l in $\mathcal{G}_P$. Let $\sigma = {\sigma'}_{|a_I\to\tau}$. We only need to show that σ is an extension of σ_l and $R\sigma \in \mathcal{S}_l$. Since a_I does not occur in σ the substitution σ is an extension of σ_l. We know that ${m'}_{|a_I\to\tau} \in \mathcal{S}_l$ and by Lemma 7 we have

$$R\sigma = R({\sigma'}_{|a_I\to\tau}) = (R\sigma')_{|a_I\to\tau} = {m'}_{|a_I\to\tau}.$$

So there is an outgoing secure channel transition (for principal i) from p_l in $\mathcal{G}_P$.

5) This is obvious as the principals in p'_l and p_l are the same. □

As explained above, as a corollary of this lemma we obtain:

Corollary 9. *If π is a rooted path in $\mathcal{G}_P$ which satisfies the reachability property (C, C', θ), then there exists a rooted path π' in $\mathcal{G}_P$ such that π' satisfies (C, C', θ) and for the substitution σ' in the final state of π' we have that for every x in the domain of π' there exists $t \in Sub(P) \cup \mathcal{A}_I$ such that $t \sqsubseteq_{\sigma'} \sigma'(x)$.*

It has been shown in [16] that a substitution which can be built from subterms of a protocol P, i.e., a substitution such as σ' in the above corollary, can be represented as a DAG of size polynomial in the size of the protocol P (i.e., the number of subterms occurring in P and the number of nodes in the principals of P). However, since in our setting subsitutions may contain atoms from the *infinite* set $\mathcal{A}_I$, bounding the size of subsitutions still allows an infinite number of possible substitutions. But it is clear that the number of different atoms from $\mathcal{A}_I$ we need to consider can be bounded by the DAG size of the substitution as well. Hence, we can simply take some fixed set of $p(|P|)$ atoms where $p(\cdot)$ is the polynomial that bounds the size of the substitution.

From this and Lemma 5 it is now easy to see that the reachability problem can be decided by the following algorithm: Given a reachability property $(C, C', \emptyset)$, the algorithm works in three steps. It first guesses a "symbolic" path through $\mathcal{G}_P$ starting from the initial state. Second, it guessses a substitution of size polynomially bounded in the size of the protocol. Finally, it checks if the path guessed is actually a path in $\mathcal{G}_P$ and if the given reachability property is satisfied when substituting all variables in the path according to the guessed substitution. In what follows, the steps are further explain:

1. The algorithm guesses a symbolic path π_s—a path where the subsitution is not determined yet—through $\mathcal{G}_P$ starting from the initial state of $\mathcal{G}_P$ by guessing transitions until it non-deterministically decides not to extend the path anymore or no principals rules are left. (Recall that the length of such a path is bounded by the number of principal rules occurring in the protocol.) A transition is guessed as follows: First, one edge originating at the root of some principal (if any) is chosen. Say the edge is labeled $R \Rightarrow S$ and the principal is i. In case $R = \varepsilon$, the algorithm extends the current path by an ε-transition labeled i and writes S into the intruder knowledge and if S is a secure channel term also into the secure channel. This determines the new "symbolic" state. Clearly, principal i is also updated according to the edge that was chosen. In case $R \neq \varepsilon$, the algorithm guesses whether the input should come from the intruder or whether it should come from the secure channel. In the former case, the transition is labeled (i, R, I) and in the latter case (i, R, sc). In case the transition is determined to be a secure channel transition, the algorithm also guesses which of the secure channel terms, say S', in the secure channel will be read in order to apply the transition.

Since the substitution is not determined yet, the terms may have variables. Now, once the transition is guessed, the new state is obtained as in the case of ε-transitions.

2. Guess a subsitution σ of the variables occurring in π_s of size polynomially bounded in the size of P.
3. Check that a) $\pi_s\sigma$—the path obtained from π_s when substituting the variables according to σ—is in fact a path in $\mathcal{G}_P$ and that b) $\pi_s\sigma$ satisfies the given reachability property.
   (a) We need to check whether the transitions can actually be applied. By construction of π_s, ε-transitions can be applied. A secure channel transition say with label $(i, R\sigma, \mathsf{sc})$ can be applied if $R\sigma$ coincides with the term $S'\sigma$ (the term determined to be read from the secure channel). An intruder transition can be applied if $R\sigma \in d(\mathcal{I}\sigma)$ where $\mathcal{I}$ is the current knowledge of the intruder in the symbolic state, and hence, $\mathcal{I}\sigma$ is the actual knowledge of the intruder in the current state. By Lemma 5, we can decide $R\sigma \in d(\mathcal{I}\sigma)$ in deterministic polynomial time in the size of P.
   (b) Assume that $\mathcal{I}\sigma$ is the current knowledge of the intruder.
      i. By Lemma 5, we can decide $c \in d(\mathcal{I}\sigma)$ for every $c \in C$ and $c' \notin d(\mathcal{I}\sigma)$ for every $c' \in C'$ in deterministic polynomial time in the size of P and the reachability property.
      ii. It is obvious how to decide (in polynomial time) whether or not the current state has an outgoing ε-transition.
      iii. As for secure channel transitions, we only need to check whether there exists an edge originating from a root of a tree describing some principal—the number of such edges is linearly bounded in the size of the protocol—labeled say with $R \Rightarrow S$ such that $R\sigma$ matches with some secure channel message in the secure channel. Note that $R\sigma$ may contain variables since there may be new variables in R not subsitutionted by σ yet. Clearly, this can be decided in polynomial time.
      iv. Similarly, to decide whether the current state has an outgoing intruder transition, we check whether there exists an edge as above labeled say with $R \Rightarrow S$ such that there exists a substitution σ' of the variables in $R\sigma$ such that $(R\sigma)\sigma' \in d(\mathcal{I}\sigma)$. This problem is obviously equivalent to the following problem: Let P' be a protocol which consists of one principal which has only one edge labeled with $R\sigma \Rightarrow \mathsf{secret}$ where secret is some new atom added to $\mathcal{A}$. The initial intruder knowledge of P' is $\mathcal{I}\sigma$. We ask whether there exists a path in $\mathcal{G}_{P'}$ which satisfies $(\{\mathsf{secret}\}, \emptyset, \emptyset)$. From the above we know that the algorithm consisting of the steps 1. to 3.,(b),i decides this problem.

It follows Theorem 2. Similar as in [16], we obtain that the reachability problem is NP-hard. Also, when restricted to reachability properties of the form $(C, C', \emptyset)$, the above algorithm is a non-deterministic polynomial time algorithm. For more general reachability properties we need to decide the presence or absense of intruder transitions, which can be done in coNP, since as shown above, this problem can be reduced to the question whether in a given protocol there is *no* path which satisfies $(\{\mathsf{secret}\}, \emptyset, \emptyset)$. Thus, REACHABILITY is in NP^{coNP}.

B.4 Deciding the Strategy Problem

The goal of this section is to prove Theorem 3. The basic idea is similar to the decidability of the reachability problem: Given a sequence of strategy graphs which satisfies the given strategy property we show that we can turn these graphs into strategy graphs where in addition all subsitutions are built from subterms of terms occurring in the protocol. However, there is a crucial different:

For the reachability problem we have shown that we can find a path such that for the substitution σ of this path (i.e., the substitution in the final state of this path) we have that for every variable x in the domain of σ there exists $t \in \mathrm{Sub}(P) \cup \mathcal{A}_I$ such that $t \sqsubseteq_\sigma \sigma(x)$. For the strategy problem, we only require that for every leaf e in the (union of the) strategy graphs and every substitution σ_e in such a leaf and every variable x in the domain of σ_e there exists $t \in \mathrm{Sub}(P) \cup \mathcal{A}_I$ and a substitution $\sigma_{e'}$ in a possibly different leaf e' such that $t \sqsubseteq_{\sigma_{e'}} \sigma_e(x)$. In the follow lemma, we show that if there is one leaf e such that its substitution σ_e does not fulfill this property, i.e., there exists x in the domain of σ_e such that for no e' and no $t \in \mathrm{Sub}(P) \cup \mathcal{A}_I$ we have $t \sqsubseteq_{\sigma_{e'}} \sigma_e(x)$, then $\sigma_e(x)$ can be replaced in all strategy graphs by a new atom $a_I \in \mathcal{A}_I$. Repeated application of this lemma thus yields a strategy graphs with subsitutions of the desired form. Before we can state the lemma, we need some notation.

For a subgraph $\mathcal{G} = (S, E, p)$ of $\mathcal{G}_P$ and messages τ and τ' we denote by $\mathcal{G}_{|\tau\to\tau'}$ the tuple $(S_{|\tau\to\tau'}, E_{|\tau\to\tau'}, p_{|\tau\to\tau'})$. Here $S_{|\tau\to\tau'}$ is the set consisting of all $q_{|\tau\to\tau'}$ where $q \in S$. The set E consists of all transitions of the form $p_{|\tau\to\tau'} \xrightarrow{\lambda_{|\tau\to\tau'}} p'_{|\tau\to\tau'}$, for $p \xrightarrow{\lambda} p'$ in E. Here $\lambda_{|\tau\to\tau'}$ is defined by

$$\lambda_{|\tau\to\tau'} = \begin{cases} \lambda & \text{if } \lambda \text{ is an } \varepsilon\text{-transition,} \\ (i, m', I) & \text{if } \lambda = (i, m, I) \text{ with } m' = m_{|\tau\to\tau'}, \\ (i, m', \mathsf{sc}) & \text{if } \lambda = (i, m, \mathsf{sc}) \text{ with } m' = m_{|\tau\to\tau'}. \end{cases}$$

For subgraphs $\mathcal{G}^1, \ldots, \mathcal{G}^s$ of $\mathcal{G}_P$ with $\mathcal{G}^i = (S_i, E_i, q)$, we denote by $\mathcal{G}^1 \cup \cdots \cup \mathcal{G}^s$ the union of $\mathcal{G}^1$ to $\mathcal{G}^s$, i.e., the tuple $(S_1 \cup \cdots \cup S_s, E_1 \cup \cdots \cup E_s, q)$.

Lemma 10. *Let P be a protocol and $\kappa = ((C_1, C'_1), \ldots, (C_s, C'_s))$ be a strategy property. Let $\mathcal{G}^1, \ldots, \mathcal{G}^s$ be q-strategy graphs of $\mathcal{G}_P$ such that they satisfy κ. Let $e = ((\Pi_1, \ldots, \Pi_n), \sigma_e, \mathcal{I}_e, \mathcal{S}_e)$ be a leaf of $\mathcal{G}^1 \cup \cdots \cup \mathcal{G}^s$ and x be a variable such that*

i) x is in the domain of $\sigma_e(x)$ and

ii) for all $t \in Sub(P) \cup \mathcal{A}_I$ and all leafs e' of $\mathcal{G}^1 \cup \cdots \cup \mathcal{G}^s$ we have $t \not\sqsubseteq_{\sigma_{e'}} \sigma_e(x)$ where $\sigma_{e'}$ is the substitution of e'.

Let $a_I \in \mathcal{A}_I$ be an intruder atom that does not occur in $\mathcal{G}^1 \cup \cdots \cup \mathcal{G}^s$ and P. Then,

$$\mathcal{G}^1{}_{|\tau\to a_I}, \ldots, \mathcal{G}^s{}_{|\tau\to a_I}$$

also satisfy κ.

Proof. In what follows, define $\tau = \sigma_e(x)$. We have to prove the following:

1) $q_{|\tau\to a_I}$ is a state in $\mathcal{G}_P$,
2) $\mathcal{G}^i{}_{|\tau\to a_I}$ is a subgraph of $\mathcal{G}_P$ for all $i = 1, \ldots, s$,
3) $\mathcal{G}^i{}_{|\tau\to a_I}$ satisfies the strategy graph conditions for all $i = 1, \ldots, s$,
4) $C_i \subseteq d(\mathcal{I}_{e'})$ and $C'_i \cap d(\mathcal{I}_{e'}) = \emptyset$ for all $i = 1, \ldots, s$ and all leafs e' of $\mathcal{G}^i{}_{|\tau\to a_I}$.

In what follows, we prove the above statements.

1) Since $q = ((\Pi_1, \ldots, \Pi_s), \sigma, \mathcal{I}, \mathcal{S})$ lies on a path from the root of $\mathcal{G}_P$ to e we know by Lemma 8 that $q_{|\tau\to a_I}$ is a state in $\mathcal{G}_P$. Note that the preconditions of this lemma are satisfied.

2) This easily follows from Lemma 8.

4) We know that every path from q to a leaf in $\mathcal{G}^i$ satisfies the reachability property $(C_i, C'_i, \emptyset)$. By Lemma 8, this carries over to the paths of $\mathcal{G}^i{}_{|\tau\to a_I}$.

3) We have to show that for $i = 1, \ldots, s$ the graph ${\mathcal{G}^i}_{|\tau \to a_I}$ is a strategy graph, that is, for all states $p_{|\tau \to a_I} \in {\mathcal{G}^i}_{|\tau \to a_I}$ we have to show the following conditions. Let $p = ((\Pi_1, \ldots, \Pi_n), \sigma, \mathcal{I}, \mathcal{S})$. (Recall that $p_{|\tau \to a_I}$ is obtained from this state by replacing every occurrence of τ by a_I.)

- If $p_{|\tau \to a_I} \xrightarrow{j} p' \in \mathcal{G}_P$, then $p_{|\tau \to a_I} \xrightarrow{j} p' \in {\mathcal{G}^i}_{|\tau \to a_I}$ for every j.

  Proof. Let $p' = ((\Pi'_1, \ldots, \Pi'_n), \sigma', \mathcal{I}', \mathcal{S}') \in \mathcal{G}_P$ such that $p_{|\tau \to a_I} \xrightarrow{j} p' \in \mathcal{G}_P$. Let $\varepsilon \Rightarrow S$ be the principal rule applied in the principal j. Because the principals in $p_{|\tau \to a_I}$ and p are the same this rule is also applicable in p, and because $\mathcal{G}^i$ is a strategy graph there is a transition $p \xrightarrow{j} p'' \in \mathcal{G}^i$ involving this rule. There is a leaf in $\mathcal{G}^i$ reachable by a rooted path in $\mathcal{G}_P$ through q, p, and p''. The subsitution of such a leaf extends σ. Now, by property ii) and Lemma 4, we can conclude that $S(\sigma_{|\tau \to a_I}) = (S\sigma)_{|\tau \to a_I}$. Hence, $p' = {p''}_{|\tau \to a_I}$, and thus, the transition $p_{|\tau \to a_I} \xrightarrow{j} p'$ is present in ${\mathcal{G}^i}_{|\tau \to a_I}$.
- If $p_{|\tau \to a_I} \xrightarrow{j,m,\mathsf{sc}} p' \in \mathcal{G}_P$, then $p_{|\tau \to a_I} \xrightarrow{j,m,\mathsf{sc}} p' \in {\mathcal{G}^i}_{|\tau \to a_I}$ for every m and j.

  Proof. Let $p_{|\tau \to a_I} \xrightarrow{j,m,\mathsf{sc}} p' \in \mathcal{G}_P$ where $p' = ((\Pi'_1, \ldots, \Pi'_n), \sigma', \mathcal{I}', \mathcal{S}')$. Let $R \Rightarrow S$ be the principal rule applied in this transition. Then σ' is an extension of $\sigma_{|\tau \to a_I}$ to $\mathsf{dom}(\sigma_{|\tau \to a_I}) \cup \mathcal{V}(R)$ and we have $m = R\sigma' \in \mathcal{S}_{|\tau \to a_I}$. Since the intruder atom a_I does not occur in any of the strategy graphs or in P, and in particular, in any of the messages of $\mathcal{S}$, we have $m_{|a_I \to \tau} \in (\mathcal{S}_{|\tau \to a_I})_{|a_I \to \tau} = \mathcal{S}$. The substitution ${\sigma'}_{|a_I \to \tau}$ is an extension of σ to $\mathsf{dom}(\sigma) \cup \mathcal{V}(R)$, and since a_I does not occur in R, we have

  $$m_{|a_I \to \tau} = (R\sigma')_{|a_I \to \tau} = R({\sigma'}_{|a_I \to \tau}).$$

  So the rule $R \Rightarrow S$ can be applied in state p and we have $p \xrightarrow{j,m_{|a_I \to \tau},\mathsf{sc}} p_1 \in \mathcal{G}^i$ with $p_1 = ((\Pi'_1, \ldots, \Pi'_n), {\sigma'}_{|a_I \to \tau}, \mathcal{I}'', \mathcal{S}'')$. We now have to show that for the message $S({\sigma'}_{|a_I \to \tau})$, which is sent in this transition, we have $(S({\sigma'}_{|a_I \to \tau}))_{|\tau \to a_I} = S\sigma'$, and that $(m_{|a_I \to \tau})_{|\tau \to a_I} = (R({\sigma'}_{|a_I \to \tau}))_{|\tau \to a_I} = m$, since then $p' = {p_1}_{|\tau \to a_I}$ and $p_{|\tau \to a_I} \xrightarrow{j,m,\mathsf{sc}} p' \in {\mathcal{G}^i}_{|\tau \to a_I}$.
  There is a leaf in $\mathcal{G}^i$ reachable by a rooted path in $\mathcal{G}_P$ through q, p, and p_1. The substitution of such a leaf extends ${\sigma'}_{|a_I \to \tau}$. Now, by property ii) and Lemma 4, we can conclude that $(S({\sigma'}_{|a_I \to \tau}))_{|\tau \to a_I} = S\sigma'$ and $(m_{|a_I \to \tau})_{|\tau \to a_I} = (R({\sigma'}_{|a_I \to \tau}))_{|\tau \to a_I} = m$.
- If

  $$p_{|\tau \to a_I} \xrightarrow{j,m_{|\tau \to a_I},I} {p'}_{|\tau \to a_I} \in {\mathcal{G}^i}_{|\tau \to a_I}$$

  and

  $$p_{|\tau \to a_I} \xrightarrow{j,m_{|\tau \to a_I},I} q_1 \in \mathcal{G}_P \text{ for some } q_1,$$

  then

  $$p_{|\tau \to a_I} \xrightarrow{j,m_{|\tau \to a_I},I} q_1 \in {\mathcal{G}^i}_{|\tau \to a_I}.$$

  Proof. Let $p_{|\tau \to a_I} \xrightarrow{j,m_{|\tau \to a_I},I} {p'}_{|\tau \to a_I} \in {\mathcal{G}^i}_{|\tau \to a_I}$ and $p_{|\tau \to a_I} \xrightarrow{j,m_{|\tau \to a_I},I} q_1 \in \mathcal{G}_P$ where $q_1 = ((\Pi^1_1, \ldots, \Pi^1_n), \sigma^1, \mathcal{I}^1, \mathcal{S}^1)$
  It suffices to show that there is a $p_1 \in \mathcal{G}_P$ such that
  - ${p_1}_{|\tau \to a_I} = q_1$ and
  - $p \xrightarrow{j,m,I} p_1 \in \mathcal{G}_P$.

Let $R \Rightarrow S$ be the principle rule that was applied for the above transition from $p_{|\tau \to a_I}$ to q_1. Then σ^1 is an extension of $\sigma_{|\tau \to a_I}$ to $\mathsf{dom}(\sigma_{|\tau \to a_I}) \cup \mathcal{V}(R)$. Consequently, ${\sigma^1}_{|a_I \to \tau}$ is an extension of σ to $\mathsf{dom}(\sigma) \cup \mathcal{V}(R)$. We know that

$$R({\sigma^1}_{|a_I \to \tau}) = (R\sigma^1)_{|a_I \to \tau} = (m_{|\tau \to a_I})_{|a_I \to \tau} = m.$$

Hence, the rule $R \Rightarrow S$ can be applied in state p, and since $\mathcal{G}^i$ is a strategy graph and $p \xrightarrow{j,m,I} p' \in \mathcal{G}^i$, the rule will be applied. Let the resulting state be p_1. To show that $p_{1|\tau \to a_I} = q_1$ it suffices to show that $(S({\sigma^1}_{|a_I \to \tau}))_{|\tau \to a_I} = S\sigma_1$. Here we apply the same argument as in the case of secure channel transitions above. □

Using this lemma, we obtain:

Lemma 11. *Let P be a protocol and $\kappa = ((C_1, C_1'), \ldots, (C_s, C_s'))$ be a strategy property such that P satisfies κ. Then, there exists a state q in $\mathcal{G}_P$ and q-strategy graphs $\mathcal{G}^1, \ldots, \mathcal{G}^s$ such that the following is true (for every $i = 1, \ldots, s$):*

1. *$\mathcal{G}^i$ satisfies (C_i, C_i') and for every state in $\mathcal{G}^i$, the number of its outgoing intruder, secure channel, and ε-transitions is bounded by the size of P, respectively.*
2. *For every leaf e in $\mathcal{G}^1 \cup \cdots \cup \mathcal{G}^s$ and every variable x in the domain of σ_e there exists $t \in Sub(P) \cup \mathcal{A}_I$ and a leaf $e' \in \mathcal{G}^1 \cup \cdots \cup \mathcal{G}^s$ such that $t \sqsubseteq_{\sigma_{e'}} \sigma_e(x)$.*

Proof. Since P satisfies $\kappa = ((C_1, C_1'), \ldots, (C_s, C_s'))$, there exists a state $q \in \mathcal{G}_P$ and q-strategy graphs $\mathcal{G}^1, \ldots, \mathcal{G}^s$ which satisfy $(C_1, C_1'), \ldots, (C_s, C_s')$, respectively.

The first statement is clear for secure channel and ε-transitions, since this is even satisfied in $\mathcal{G}_P$, and thus, in particular for every $\mathcal{G}^i$ since they are subgraphs of $\mathcal{G}_P$.

As for intruder transitions, first observe that if in a q-strategy graph which satisfies (C, C'), say, there exists a node with at least two outgoing intruder edges with different labels (i.e., differing messages or principal names), say one of the labels is λ, then if all transitions with label other than λ are removed, the resulting graph (consisting only of states reachable from q) will still be a q-strategy graph and will still satisfy (C, C'). Hence, we may assume that for every state in $\mathcal{G}^i$ all outgoing intruder transitions have the same label. This means that in all intruder transitions the same message is sent to the same principal. Now, the number of different edges originating from the root of this principal is bounded by the size of P. Moreover, applying the same principal rule (on the same edge of the principal) to the same message, yields the same resulting state. Thus, in effect, the number of outgoing intruder transitions is bounded by the size of P.

In summary, we may assume that the q-strategy graphs $\mathcal{G}^i$ satisfy property 1. This property implies that the number of outgoing transitions (of any kind) of every state of a strategy graph $\mathcal{G}^i$ is polynomially bounded in the size of P. Since the length of the longest path in $\mathcal{G}_P$, and thus $\mathcal{G}^i$, is also bounded by the size of P, we obtain that the number of leaves of $\mathcal{G}^i$ is bounded exponentially in P.

Now, if for these strategy graphs property 2. is not satisfied, we can repeatedly apply Lemma 10 until property 2. is satisfied. Note that since the number of leaves in the strategy graphs $\mathcal{G}^i$ is exponentially bounded in the size of P, Lemma 10 only needs to be applied a finite number of times. Also, observe that the transformation of the graphs and the state q performed in Lemma 10 preserve property 1. □

Based on Lemma 11, we can now define an algorithm which decides STRATEGY. We will first consider the case that the strategy property is of the form $((C, C'))$. As we will see, it is easy to extend this to the more general case $((C_1, C_1'), \ldots, (C_s, C_s'))$.

By (the proof of) Lemma 11, we know that the number of leaves of (the union of) the strategy graphs can be bounded exponentially in the size of the protocol and the strategy property. With this, property 2. of Lemma 11 guarantees that the DAG size of the family of substitutions at the leaves of the strategy graphs can be bounded by an exponential in the size of the protocol and the strategy property.

Given $((C, C'))$, our algorithm works as follows: In a nutshell, the algorithm first guesses a path to some symbolic state and from there it guesses a symbolic graph. Then, it guesses a substitution, and finally, based on this subsitution, checks whether the symbolic state can in fact be reached, whether the graph is in fact a strategy graph, and whether the strategy property is satisfied. More precisely, the algorithm performs the following steps:

1. The algorithm guesses a symbolic path—just as the algorithm for the reachability problem (Section B.3)—from the root of $\mathcal{G}_P$ to some symbolic state q_s. While constructing the path, the algorithm decides non-deterministically when to stop.
2. Starting from q_s, the algorithm now guesses zero, one, or more symbolic outgoing transitions, where every symbolic transition is constructed just as in the case of constructing symbolic paths. By Lemma 11, we know that the number of outgoing transistions of one type can be bounded by the size of P. At the newly generated symbolic states (if any) the algorithm proceeds in the same way (by processing each of these states one at a time). If no new and unprocessed symbolic states are left, this phase of the algorithm stops. The symbolic structure constructed by the algorithm thus far is considered a tree. The number of leaves of this tree is exponentially bounded in the size of P and the given strategy graph property.
3. For every leaf of the symbolic tree the algorithm guesses a substitution where the domain of the substitution is the set of variables occurring on the path in the tree to this leaf starting from the root of the symbolic tree (which coincides with the root of $\mathcal{G}_P$). This family of subsitutions is represented as a DAG. As mentioned above, we know that the size of this DAG can be bounded exponentially in the size of P and the strategy properties. When applying the substitutions we obtain a graph which we call $\mathcal{G}$. To obtain this graph, we may need to merge some of the states and transitions of the tree since after applying the substitution some symbolic states and transitions may coincide. We call the state obtained when applying the corresponding substitution to a symbolic state, the instance of this state.
4. The algorithm now checks whether the path from the initial state of $\mathcal{G}$ (which is also the initial state of $\mathcal{G}_P$) to the instance of q_s is in fact a path in $\mathcal{G}_P$ and that the paths in $\mathcal{G}$ starting from the instance of q_s to the leaves of $\mathcal{G}$ are in fact paths in $\mathcal{G}_P$. This is done as in the case of the algorithm for the reachability problem. If this check is successful, it follows that $\mathcal{G}$ is a subgraph of $\mathcal{G}_P$.
5. The algorithm checks whether the subgraph $\mathcal{G}'$ of $\mathcal{G}$ rooted at the instance of q_s is in fact a strategy graph by checking for every state of $\mathcal{G}'$ the required conditions. It is easy to see that this can be decided for every such state. The argument is similar to the one for checking reachability property.
6. Check whether $\mathcal{G}'$ satisfies (C, C'). This is decided as in the case of the reachability property.

With Lemma 11, it is easy to see that this algorithm decides whether P satisfies the strategy property $((C, C'))$. The algorithm can easily be extended to handle strategy properties of the

form $((C_1, C'_s), \ldots, (C_s, C'_s))$ as follows: Step 1. is exactly the as above. In step 2. , the algorithm does esstentially the same as above. However, the number of outgoing transitions of one kind at one symbolic state may be s times the size of P since the structure guessed by the algorithm is meant to represent (symbolic versions of) the union of s strategy graphs. To be able to extract the different graphs, colors $1, \ldots, s$ are assigned to the states. States may have several colors. The state q_s has all colors. The colors are assigned in such a way that if a symbolic state (except q_s) is assigned a certain color, then the preceeding state has been assigned this color as well. Step 3. is unchanged. If to construct $\mathcal{G}$ states have to be merged, the union of the corresponding sets of colors is taken. Step 4. stays the same as well. Step 5. is performed on every subgraph of $\mathcal{G}'$ induced by a certain color. Step 6. is again done for every subgraph of $\mathcal{G}'$ induced by a certain color. Using Lemma 11, it easily follows that this algorithm decides STRATEGY. Hence, we obtain Theorem 3. A complexity analysis of this algorithm reveals that the algorithms runs in at most deterministic double exponential time. We note that if we consider a rooted version of the strategy problem, i.e., a version where q is given, the problem can be shown to be PSPACE-hard by a straightforward reduction from QBF.