Infinite State AMC-Model Checking for Cryptographic Protocols*

Detlef Kähler
University of Kiel, Germany
kaehler@ti.informatik.uni-kiel.de

Ralf Küsters
ETH Zurich, Switzerland
ralf.kuesters@inf.ethz.ch

Tomasz Truderung
Wroclaw University, Poland
tomasz.truderung@ii.uni.wroc.pl

Abstract

Only very little is known about the automatic analysis of cryptographic protocols for game-theoretic security properties. In this paper, we therefore study decidability and complexity of the model checking problem for AMC-formulas over infinite state concurrent game structures induced by cryptographic protocols and the Dolev-Yao intruder. We show that the problem is NEXPTIME-complete when making reasonable assumptions about protocols and for an expressive fragment of AMC, which contains, for example, all properties formulated by Kremer and Raskin in fair ATL for contract-signing and non-repudiation protocols. We also prove that our assumptions on protocols are necessary to obtain decidability, unless other restrictions are imposed on protocols.

1 Introduction

The design of cryptographic protocols is highly error-prone as these protocols have to achieve their security goals even in presence of an adversary who controls part of the communication network and in presence of dishonest parties who deviate from the protocol specification. Rigorous analysis of these protocols is therefore indispensable. Several algorithms and tools for the (fully) automatic analysis of cryptographic protocols have been developed and successfully applied (see, e.g., [16, 3]). One of the fundamental results in the area is that the security of protocols can be decided for a bounded number of sessions and w.r.t. the so-called Dolev-Yao intruder, with no restrictions put on the size of messages (see, e.g., [17, 16, 5]). However, these results are restricted to reachability properties, such as secrecy and authentication. They do not apply to cryptographic protocols with more complex, game-theoretic security requirements, such as those for non-repudiation and contract-signing protocols, including, for example, different versions of fairness, timeliness, balance, and abuse-freeness (see, e.g., [14, 15]). For instance, one version of fairness for non-repudiation protocols taken from [14] requires that (dishonest) Bob does not have a strategy (in collaboration with certain communication channels) to reach a state in which he has a proof of origin but (honest) Alice does not have a strategy (against the other players) to obtain her proof of receipt.

Only recently a first decidability result for a *specific* game-theoretic security property, namely balance, has been obtained [12, 10] (see the related work). The goal of the present work is to study decidability and complexity of cryptographic protocol analysis in a much more general setting in which game-theoretic security properties are expressed in terms of the Alternating-time μ-Calculus (AMC), which strictly contains ATL*, and hence, provided a suitable set of propositional variables, also fair ATL [2].

More precisely, in this paper we formalize the possible executions of protocols along with the Dolev-Yao intruder in terms of a certain class of infinite-state concurrent game structures [2], which we call *security-specific concurrent game structures*. These concurrent game structures have an *infinite* state space since at every execution step the Dolev-Yao intruder can choose messages to be sent to principals among an *infinite* set of possible messages. Similar to [13], we model the realistic situation that (honest and dishonest) principals may take actions at the same time and may receive/write several messages from/to other principals at the same time. Since many cryptographic protocols with game-theoretic security requirements assume resilient channels (also called secure channels here), i.e., channels that, unlike the network, are not under the control of the Dolev-Yao intruder, our model comprises such channels. We distinguish between direct and scheduled secure channels: A direct se-

*This work was partially supported by the DFG under Grant KU 1434/4-1, the SNF under Grant 200021-116596/1, and the Polish Ministry of Science and Education under Grant 3 T11C 042 30.

cure channel is a direct link between principals. Messages sent on scheduled secure channels are first sent to a buffer before being delivered to the intended recipient. The buffer is a player in the security-specific concurrent game structure and may team up with (honest or dishonest) principals or other scheduled secure channels, as can be specified by an AMC-formula. Honest principals are specified by finite edge-labeled trees where an edge is labeled by a rule which describes a possible receive-send action of a principal at the current step. Vertices in these trees may have self-loops to allow a principal to stay in the current state.

Based on the security-specific concurrent game structures that we define, game-theoretic security requirements for protocols can conveniently be expressed in terms of AMC-formulas (or alternatively, ATL*-formulas). In order to decide whether a given protocol satisfies a given security property, expressed as AMC-formula, one has to decide the AMC-model checking problem over the security-specific concurrent game structures, where the input to the problem is the protocol (which together with the Dolev-Yao intruder induces the security-specific concurrent game structure) and the AMC-formula.

Our main technical results are as follows: We show that the above model checking problem is undecidable for a class of protocols in which honest principals may be what we call *non-greedy*, i.e., they may ignore received messages even though they conform to the protocol specification. The undecidability result holds for a relatively simple, fixed AMC-formula. Fortunately, in typical protocol specifications, honest principals are greedy, i.e., they do not ignore messages that conform to the protocol specification. Hence, requiring honest principals to be greedy is reasonable from a practical point of view. We also exhibit another source of undecidability, namely protocols that involve *scheduled* secure channels *from the Dolev-Yao intruder* (i.e., dishonest principals) to honest principals. This undecidability result holds for greedy principals and again a fixed, simple AMC-formula. Since we allow the Dolev-Yao intruder to send messages over direct secure channels to principals, disallowing scheduled secure channels from the Dolev-Yao intruder to honest principals does not limit the power of the intruder. These undecidability results show that to obtain decidability it is necessary to consider only protocols with greedy principals and without scheduled secure channels from the Dolev-Yao intruder to honest principals, unless other restrictions are imposed on protocols. For this class of protocols we indeed obtain decidability, more accurately (co-)NEXPTIME-completeness, of the model checking problem for an expressive fragment of AMC, consisting of what we call $\mathcal{I}$-positive ($\mathcal{I}$-negative) AMC-formulas, where $\mathcal{I}$ is the name of the Dolev-Yao intruder in the concurrent game structure. An AMC-formula φ is $\mathcal{I}$-positive if all subformulas of φ of the form $\langle\langle A\rangle\rangle\bigcirc\psi$ with $\mathcal{I} \in A$ fall under an even number of negations and all subformulas of φ of the form $\langle\langle A\rangle\rangle\bigcirc\psi$ with $\mathcal{I} \notin A$ fall under an odd number of negations; a formula is $\mathcal{I}$-negative if its negation is $\mathcal{I}$-positive. We subsume the set of $\mathcal{I}$-positive and $\mathcal{I}$-negative formulas under the notion $\mathcal{I}$-monotone formulas. The same terminology can be applied to ATL*-formulas. It is easy to see that the property of being $\mathcal{I}$-positive/-negative is invariant under the translation from ATL* to AMC as described in [2]. Kremer and Raskin were the first to express game-theoretic security properties in terms of fair ATL [14, 15]. It turns out that all the properties that they have formulated, including for instance various forms of fairness, timeliness, balance, and abuse-freeness, fall into the $\mathcal{I}$-monotone fragment of ATL*, and hence, the $\mathcal{I}$-monotone fragment of AMC, indicating that the $\mathcal{I}$-monotone fragment suffices for most properties of interest.

The complexity upper bound is proved by a novel combination of techniques from the theory of infinite games, such as parity games and memoryless strategies, and techniques from cryptographic protocol analysis for reachability properties.

Related work. In [19, 14, 15], specific protocols have been analyzed w.r.t. game-theoretic security properties using the finite-state model checkers Murphi and MOCHA, where in [14, 15] several game-theoretic properties have been formulated in fair ATL. The disadvantage of using finite-state model checking is that the Dolev-Yao intruder has to be approximated and actions of dishonest principals have to be anticipated to some extent. The present work shows that fully automated analysis of game-theoretic security requirements is possible also w.r.t. a fine-grained infinite-state model and the standard Dolev-Yao intruder.

As already mentioned, a first decidability result for a specific security property, namely balance, was proved in [12] (see also [10] for a constraint-based algorithm). The present work considerably generalizes [12, 10] in terms of the security properties that can be checked, as here we consider a comprehensive class of security properties, expressed as $\mathcal{I}$-monotone AMC-formulas. Also, unlike [12, 10], the present work contains undecidability and (tight) complexity-theoretic results. Finally, following [7], [12, 10] use a model with interleaving semantics, and hence, unlike our real concurrent model, at every time only one principal may be active and a principal can only receive/send one message at a time. As demonstrated in [13], taking real concurrency into account yields a more realistic model.

In [8], Corin et al. proposed a procedure for deciding trace-based properties in a variant of LTL with only past temporal operators. While they cannot express game-theoretic security properties, it seems that the (trace-based) properties they have formulated in their logic can also be formulated in the $\mathcal{I}$-monotone fragment of AMC. Complexity-theoretic results are not provided by Corin et

al. and they consider a model with interleaving semantics, rather than real concurrency.

Structure of this paper. In the next section, we recall the definition of concurrent game structures and AMC. The security-specific model that we use, in particular the infinite-state concurrent game structures induced by protocols and the Dolev-Yao intruder are introduced in Section 3. The main results are summarized in Section 4. We conclude in Section 5. Full definitions and proofs can be found in our technical report [11].

2. Concurrent Game Structure and AMC

Following [1, 2], in this section we recall the definition of concurrent game structures and the Alternating-time μ-Calculus (AMC). Our definition of a concurrent game structure differs from the one in [2] in two aspects: First, the structures we consider may have an *infinite* state space and in one state players may have an *infinite* number of possible moves. Second, for convenience, in our setting moves are not identified with natural numbers; they may for example be terms or vertices of trees.

Concurrent Game Structures. A *concurrent game structure (CGS)* is a tuple $S = \langle \Sigma, Q, \mathbb{P}, \pi, \Delta, \delta \rangle$ where

- Σ is a non-empty, finite set of *players*,
- Q is a (possibly infinite) set of *states*,
- $\mathbb{P}$ is a finite set of *propositional variables/propositions*,
- $\pi : Q \to 2^{\mathbb{P}}$ is a *labeling function* (which assigns every state to the set of propositions true in this state),
- Δ is a function which for each state $q \in Q$ and each player $a \in \Sigma$ returns a (possibly infinite) set $\Delta(q, a)$ of *moves* available at state q to player a.

  For $A \subseteq \Sigma$ and $q \in Q$, an (A, q)-*move* is a function c which maps every $a \in A$ to a move $c(a) \in \Delta(q, a)$. Given $A \subseteq \Sigma$ and a state q, we write $\Delta^A(q)$ for the set of (A, q)-moves. An (A, q)-move is called a *total move* if $A = \Sigma$.
- δ is a *transition function* which, for each state q and each total move $c \in \Delta^\Sigma(q)$, returns a state $\delta(q, c) \in Q$ (the state obtained when in state q all players simultaneously perform their moves according to c).

For a state q, a set of players $A \subseteq \Sigma$, and an (A, q)-move $c \in \Delta^A(q)$, we say that a state $q' \in Q$ is a *c–successor of q* if $q' = \delta(q, c')$, for some total move $c' \in \Delta^\Sigma(q)$ such that $c'(a) = c(a)$, for all $a \in A$.

Syntax of AMC-Formulas. An AMC-formula over the set $\mathbb{P}$ of propositions, the set $\mathcal{V}$ of variables, and the set Σ of players is one of the following: $p \in \mathbb{P}$; $X \in \mathcal{V}$; $\neg\varphi$ if φ is an AMC-formula; $\varphi_1 \vee \varphi_2$ if φ_1 and φ_2 are AMC-formulas; $\langle\langle A \rangle\rangle \bigcirc \varphi$ if $A \subseteq \Sigma$ and φ is an AMC-formula; $\mu X.\varphi$ if φ is an AMC-formula and all free occurrences of X (i.e., those that are not bound by μX) fall under an even number of negations.

We use the following common abbreviations: $\varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi)$, and $\nu X.\varphi = \neg\mu X.\neg\varphi[X/\neg X]$ where $\varphi[X/\neg X]$ is obtained from φ by replacing every free occurrence of X in φ by $\neg X$ and vice versa.

An AMC-formula is a *sentence* if it does not contain free variables, i.e., all variables are bound by fixed-point operators (μX and νX). The *size* of an AMC-formula φ, denoted by $|\varphi|$, is defined inductively in the natural way.

Semantics of AMC. AMC-formulas are interpreted over concurrent games structures. To define the semantics of AMC-formulas, we need some definitions and notations. Given a concurrent game structure $S = \langle \Sigma, Q, \mathbb{P}, \pi, \Delta, \delta \rangle$, a *valuation* F is a function from the set of variables $\mathcal{V}$ to 2^Q, i.e., the power set of Q. For F, a variable X, and a set $M \subseteq Q$, we denote by $F[X := M]$ the valuation that maps X to M and agrees with F on all other variables.

An AMC-formula φ is interpreted as a mapping φ^S from valuations to state sets. Intuitively, $\varphi^S(F)$ denotes the set of states in which φ is satisfied under the valuation F in the structure S. The mapping φ^S is defined inductively as follows:

- $p^S(F) = \{q \in Q \mid p \in \pi(q)\}$ for $p \in \mathbb{P}$.
- $X^S(F) = F(X)$ for $X \in \mathcal{V}$.
- $(\neg\varphi)^S(F) = Q \setminus \varphi^S(F)$.
- $(\varphi_1 \vee \varphi_2)^S(F) = \varphi_1^S(F) \cup \varphi_2^S(F)$.
- $(\langle\langle A \rangle\rangle \bigcirc \varphi)^S(F) = \{q \in Q \mid$ there exists $c \in \Delta^A(q)$ such that every c-successor of q belongs to $\varphi^S(F)\}$.
- $(\mu X.\varphi)^S(F) = \bigcap\{M \subseteq Q | \varphi^S(F[X := M]) \subseteq M\}$, i.e., $(\mu X.\varphi)^S(F)$ is the least fixed-point of the function that maps $M \subseteq Q$ to $\varphi^S(F[X := M])$. (Note that this function is monotonic.)

Note that if φ is a sentence, then the interpretation of φ in the structure S is uniquely determined independently of a valuation function F. We may therefore simply write φ^S instead of $\varphi^S(F)$ for some F. Given a state q of a CGS S and a sentence φ, we write $(S, q) \models \varphi$ if $q \in \varphi^S$.

Deciding $(S, q) \models \varphi$ for a given finitely represented CGS S, a state q in S, and a sentence φ is an AMC-model checking problem. The main purpose of this paper is to study this problem for a class of CGSs induced by cryptographic protocols and the Dolev-Yao intruder.

Fact 1 *[2] AMC is more expressive than ATL*, and hence, (fair) ATL (provided a suitable set of propositional variables).*

3 The Protocol and Intruder Model

We now introduce our protocol and intruder model. As mentioned in the introduction, similar to [13], we model the realistic situation that honest principals and the Dolev-Yao intruder may take actions at the same time and may receive/write several messages from/to other principals at the same time. Principals are connected via network channels and resilient channels (also called secure channels here), where the latter can be direct and scheduled.

In what follows, we define (i) terms and messages, (ii) how the Dolev-Yao intruder can derive new messages from a given set of messages, (iii) principals and protocols, and (iv) the concurrent game structures induced by protocols and the Dolev-Yao intruder.

Terms and Messages. Let $\mathcal{V}$ be a finite set of variables, $\mathcal{A}$ be a finite set of *atoms* (*atomic messages*, like names of principals, atomic symmetric keys, and nonces, i.e. random numbers generated by principals), $\mathbb{K}$ be a finite set of *public* and *private keys*, and $\mathcal{A}_I$ be an infinite set of *intruder atoms* (the nonces, symmetric keys, etc. the intruder may generate). These sets are assumed to be pairwise disjoint. The set $\mathbb{K}$ is partitioned into a set $\mathbb{K}_{pub}$ of public keys and a set $\mathbb{K}_{priv}$ of private keys, with a bijective mapping $\cdot^{-1}\colon \mathbb{K} \to \mathbb{K}$ which assigns to every public key the corresponding private key and vice versa. The set $\mathcal{T}$ of *terms* is defined as follows:

$$\mathcal{T} ::= \mathcal{V} \mid \mathcal{A} \mid \mathcal{A}_I \mid \langle \mathcal{T}, \mathcal{T} \rangle \mid \{\mathcal{T}\}^s_{\mathcal{T}} \mid \{\mathcal{T}\}^a_{\mathbb{K}_{pub}} \mid \mathsf{hash}(\mathcal{T}) \mid \mathsf{sig}(\mathbb{K}_{pub}, \mathcal{T}).$$

Let $\mathcal{V}(t)$ denote the set of variables in term t. Terms without variables (i.e., ground terms) are called *messages*. The set of messages is denoted by $\mathcal{M}$. 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 term, i.e., we allow complex keys), $\{t\}^a_k$ 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 (generated using k^{-1}) which can be verified with the public key k. One could also add other primitives, such as *private contract signatures (PCSs)* as in [12]. While our results also hold when, for example, PCSs are added, for simplicity of presentation we omit further primitives here.

We define $\mathcal{T}_\circ = \mathcal{T} \cup \{\circ\}$ and $\mathcal{M}_\circ = \mathcal{M} \cup \{\circ\}$ where '$\circ$' is a new symbol which stands for 'no message'. This symbol will be used in case there is no message on a channel. *Substitutions* σ and their application $t\sigma$ to terms t are defined as usual.

Derivation of Messages. Given a set $\mathcal{K}$ of messages, the (infinite) set $d(\mathcal{K})$ of messages the Dolev-Yao intruder can derive from $\mathcal{K}$ is defined as usual as the smallest set satisfying the following conditions, where $m, m' \in \mathcal{M}$: (1) $\mathcal{K} \subseteq d(\mathcal{K})$. (2) *Composition and decomposition*: If $m, m' \in d(\mathcal{K})$, then $\langle m, m' \rangle \in d(\mathcal{K})$. If $\langle m, m' \rangle \in d(\mathcal{K})$, then $m \in d(\mathcal{K})$ and $m' \in d(\mathcal{K})$. (3) *Symmetric encryption and decryption*: If $m, m' \in d(\mathcal{K})$, then $\{m\}^s_{m'} \in d(\mathcal{K})$. If $\{m\}^s_{m'} \in d(\mathcal{K})$ and $m' \in d(\mathcal{K})$, then $m \in d(\mathcal{K})$. (4) *Asymmetric encryption and decryption*: If $m \in d(\mathcal{K})$ and $k \in d(\mathcal{K}) \cap \mathbb{K}_{pub}$, then $\{m\}^a_k \in d(\mathcal{K})$. If $\{m\}^a_k \in d(\mathcal{K})$ and $k^{-1} \in d(\mathcal{K}) \cap \mathbb{K}_{priv}$, then $m \in d(\mathcal{K})$. (5) *Hashing*: If $m \in d(\mathcal{K})$, then $\mathsf{hash}(m) \in d(\mathcal{K})$. (6) *Signing*: If $m \in d(\mathcal{K}), k^{-1} \in d(\mathcal{K}) \cap \mathbb{K}_{priv}$, then $\mathsf{sig}(k, m) \in \mathcal{K}$. (The signature contains the public key but can only be generated if the corresponding private key is known.) (7) *Generating fresh constants*: $\mathcal{A}_I \subseteq d(\mathcal{K})$.

Channels, Principals, and Protocols. We denote by $\mathcal{P}$ the finite set of all principals. This set is partitioned into the set $\mathcal{H}$ of *honest* and the set $\mathcal{D}$ of *dishonest* principals. All dishonest principals will be subsumed by the Dolev-Yao intruder. The behavior of honest principals will be specified by certain trees (see below). Protocols will basically be defined by a set of such trees, specifying the behavior of all honest principals participating in a protocol run. First, we have to define how principals are connected via channels.

Channels and Multi Terms. We consider three types of communication channels between principals (including the intruder): (1) *network channels*, (2) *direct secure channels*, and (3) *scheduled secure channels*. Network channels are controlled by the intruder, i.e., every message sent on a network channel by an honest principal is immediately delivered to the intruder and every message received from a network channel by an honest principal was sent by the intruder. A direct secure channel is a direct link between principals, i.e., every message sent on such a channel by some principal to another principal will immediately be delivered to the latter principal without intervention by the intruder. Messages sent via a scheduled secure channel will first be sent to a buffer before they are delivered to the intended recipient. Such a buffer will be modeled as a player in a CGS and may team up with (honest or dishonest) principals or other channels, as can be specified by AMC-formulas, which describe security properties.

A network channel from a principal a to a principal b such that $a \neq b$ and not both a and b are dishonest will be denoted by $\mathsf{net}(a, b)$. Similarly, we use $\mathsf{dir}(a, b)$ and $\mathsf{sch}(a, b)$ to refer to direct and scheduled secure channels from a to b, respectively. The set of all the channels will be denoted by $\mathcal{C}$. For sets $A, B \subseteq \mathcal{P}$ of principals, we define $\mathsf{Net}(A, B) = \{\mathsf{net}(a, b) \mid a \in A, b \in B, a \neq b, \text{ and } (a \in \mathcal{H} \text{ or } b \in \mathcal{H})\}$. Similarly, we define $\mathsf{Dir}(A, B)$ and $\mathsf{Sch}(A, B)$ for direct and scheduled secure channels. We define $\mathcal{C}(A, B) = \mathsf{Net}(A, B) \cup \mathsf{Dir}(A, B) \cup \mathsf{Sch}(A, B)$. We will write, for example, $\mathcal{C}(a, B)$ instead of $\mathcal{C}(\{a\}, B)$.

For a set $C \subseteq \mathcal{C}$, we call a mapping $\mathbf{r} : C \to \mathcal{T}_\circ$ a *multi term* and a mapping $\mathbf{m} : C \to \mathcal{M}_\circ$ a *multi message*. We de-

note by $\mathsf{ch}(\mathbf{m})$ and $\mathsf{ch}(\mathbf{r})$ the domain C of $\mathbf{m}$ and $\mathbf{r}$, respectively, and by $\mathcal{V}(\mathbf{r})$ the set of variables occurring in $\mathbf{r}$, i.e., in the set $\{t \mid \mathbf{r}(c) = t \text{ for some } c \in C\}$. If σ is a substitution, we denote by $\mathbf{r}\sigma$ the multi term obtained by substituting every variable $x \in \mathcal{V}(\mathbf{r})$ in $\mathbf{r}$ by $\sigma(x)$, i.e., $\mathbf{r}\sigma(c) = \mathbf{r}(c)\sigma$ for every $c \in C$.

Let $\mathbf{m}$ be a multi message, $\mathbf{r}$ be a multi term, and σ be a substitution with domain $\mathcal{V}(\mathbf{r})$. We say that $\mathbf{m}$ *matches with* $\mathbf{r}$ *by* σ, if $\mathsf{ch}(\mathbf{r}) \subseteq \mathsf{ch}(\mathbf{m})$ and $\mathbf{m}(c) = \mathbf{r}(c)\sigma$ for each $c \in \mathsf{ch}(\mathbf{r})$. We say that $\mathbf{m}$ *matches with* $\mathbf{r}$, if there is a substitution σ such that $\mathbf{m}$ matches with $\mathbf{r}$ by σ.

Honest Principals. We now define honest principals; more precisely, we should say 'instances of honest principals' since a principal might run several copies of a protocol in possibly different roles. Informally speaking, an honest principal is defined by a finite edge-labeled tree which describes the behavior of this principal in a protocol run. Each edge of such a tree is labeled by a rule which describes the receive-send action that is performed when the principal takes this edge in a run of the protocol. As mentioned above, in *one* receive-send action a principal may receive/send several messages on different channels. The trees that we consider may have self-loops. These allow a principal to stay in the same state. When a principal carries out a protocol, it traverses its tree, starting at the root. In every node, the principal takes its current input (on all channels the principal has access to or wants to read), chooses one of the edges leaving the node such that the current inputs match with the left-hand side of the rule the edge is labeled with, sends out (possibly different) messages on (possibly different) channels as determined by the right-hand side of the rule, and moves to the node the chosen edge leads to. Edges have priorities which influence which edge may be taken in case several edges are applicable. However, if several edges with the same priority can be taken, one such edge is picked non-deterministically. Formally, principals are defined as follows:

For sets $C, D \subseteq \mathcal{C}$, we call $\mathbf{r} \Rightarrow \mathbf{s}$ with $\mathbf{r} : C \to \mathcal{T}_\circ$ and $\mathbf{s} : D \to \mathcal{T}_\circ$ a (C, D)*–rule*. For an honest principal $a \in \mathcal{H}$, an a*–rule* is a (C, D)–rule with $C \subseteq \mathcal{C}(\mathcal{P}, a)$ and $D \subseteq \mathcal{C}(a, \mathcal{P})$.

Let $a \in \mathcal{H}$ be an honest principal. Its behavior is specified by what we call an a-instance (or simply honest principal). An a*-instance* (*honest principal*) is defined by a finite tree $P = (V, E, r, \ell_p, \ell)$ where V is the set of vertices, E is the set of edges, $r \in V$ is the root of the tree, and ℓ_p maps every edge $e \in E$ of P to a natural number, the *priority of this edge*. The labeling function ℓ maps every edge $e = (v, v') \in E$ of P to an a-rule $\ell(e)$ in such a way that every variable occurring in $\mathcal{V}(\mathbf{s})$ with $\ell(e) = (\mathbf{r} \Rightarrow \mathbf{s})$ also occurs on the left-hand side of $\ell(e)$, i.e., is in $\mathcal{V}(\mathbf{r})$, or on the left-hand side of a rule on the path from the root r to v. In other words, every variable occurring on the right-hand side of a rule also occurs on the left-hand side of this or a preceding rule. Nodes of P may have *self-loops*, i.e., P may contain edges of the form $e = (v, v)$ for $v \in V$. In that case, we require that for $\ell(e) = (\mathbf{r} \Rightarrow \mathbf{s})$ the domains of $\mathbf{r}$ and $\mathbf{s}$ are empty, i.e., $\mathsf{ch}(\mathbf{r}) = \emptyset$ and $\mathsf{ch}(\mathbf{s}) = \emptyset$. In other words, a self-loop can always be applied and does not produce output.

For an a-instance P, we denote by $\mathsf{ch}(P)$ the set of all channels used by P, i.e., $\mathsf{ch}(P)$ consists of those channels c for which there exists an edge in P labeled with a rule of the form $\mathbf{r} \Rightarrow \mathbf{s}$ such that $c \in \mathsf{ch}(\mathbf{r})$ or $c \in \mathsf{ch}(\mathbf{s})$.

Protocols. A *protocol* is a tuple $Pr = (\mathcal{H}, \mathcal{D}, \mathcal{K}, \{P_a\}_{a \in \mathcal{H}})$ where $\mathcal{H}$ and $\mathcal{D}$ are disjoint sets of honest and dishonest principals, respectively, P_a is an a-instance for each $a \in \mathcal{H}$, and $\mathcal{K}$ is the *initial intruder knowledge*, i.e., a finite set of messages. W.l.o.g., we assume that the set of vertices of the trees P_a, $a \in \mathcal{H}$, are pairwise disjoint. For a protocol Pr, we denote by $\mathsf{ch}(Pr)$ the set of channels used in Pr, i.e. the set of all channels c such that $c \in \mathsf{ch}(P_a)$, for some $a \in \mathcal{H}$. The size of Pr, denoted by $|Pr|$, is defined according to some standard representation of Pr.

Example: The ASW Contract-signing Protocol. To illustrate the definition of honest principals and protocols introduced above, we show how one can specify the originator of the ASW protocol [4] in our model. First, we provide an informal overview of the protocol.

The objective of the ASW protocol is to enable two principals A (the originator) and B (the responder) to obtain each other's signature on a previously agreed contractual text $\mathsf{contract}$ with the help of a trusted third party (TTP) T, which however is only invoked in case of problems. In other words, the ASW protocol is an optimistic two-party contract-signing protocol.

In the following, we write $\mathsf{sig}[k, m]$ as an abbreviation for $\langle m, \mathsf{sig}(k, m)\rangle$ and we write $\langle m_1, \ldots, m_n\rangle$ instead of $\langle m_1, \langle m_2, \langle \cdots \langle m_{n-1}, m_n\rangle\rangle\rangle\rangle$. We denote the public or verification key of a principal A by k_A. There are two kinds of messages that are considered as a valid contract: the standard contract $\langle \mathsf{sig}[k_A, m_A], N_A, \mathsf{sig}[k_B, m_B], N_B\rangle$ and the replacement contract $\mathsf{sig}[k_T, \langle \mathsf{sig}[k_A, m_A], \mathsf{sig}[k_B, m_B]\rangle]$, where $m_A = \langle k_A, k_B, k_T, \mathsf{contract}, \mathsf{hash}(N_A)\rangle$, $m_B = \langle \mathsf{sig}[k_A, m_A], \mathsf{hash}(N_B)\rangle$, and N_A and N_B are nonces.

The ASW protocol consists of three subprotocols: the exchange, abort, and resolve protocol. These subprotocols are explained next.

Exchange protocol: The basic idea of the exchange protocol is that A first indicates her interest to sign the contract. To this end, she sends to B the message $\mathsf{sig}[k_A, m_A]$ as defined above, where N_A is a nonce generated by A. By sending this message, A "commits" to signing the contract. Then, similarly, B indicates his interest to sign the contract by generating a nonce N_B and sending the message

$\mathsf{sig}[k_B, m_B]$ to A. Finally, first A and then B reveal N_A and N_B, respectively. At this point both participants can built a standard contract.

Abort protocol: If, after A has sent her first message, B does not respond, A may contact T to abort, i.e., A runs the abort protocol with T. Note that A may wait as long as she wants before contacting T. (In our formal model, this is modeled as a non-deterministic choice of A.) In the abort protocol, A first sends the message $a_A = \mathsf{sig}[k_A, \langle \mathsf{aborted}, \mathsf{sig}[k_A, m_A]\rangle]$. If T has not received a resolve request before (see below), then T sends back to A the *abort token* $a_T = \mathsf{sig}[k_T, \langle \mathsf{aborted}, a_A\rangle]$. Otherwise (if T received a resolve request, which in particular involves the messages $\mathsf{sig}[k_A, m_A]$ and $\mathsf{sig}[k_B, m_B]$ from above), it sends the *replacement contract* $r_T = \mathsf{sig}[k_T, r]$ to A with $r = \langle \mathsf{sig}[k_A, m_A], \mathsf{sig}[k_B, m_B]\rangle$.

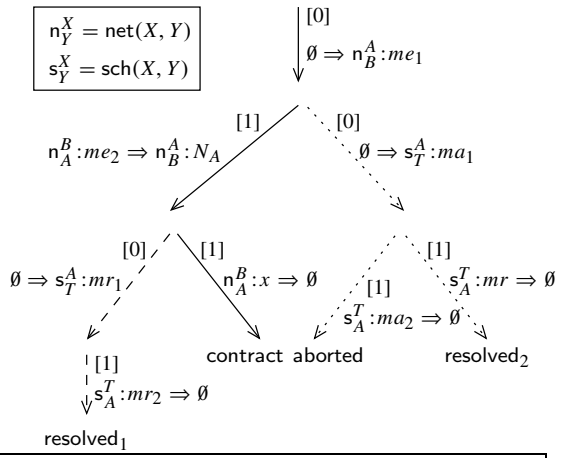
Resolve protocol: If, after A has sent the nonce N_A, B does not respond, A may contact T to resolve, i.e., A runs the resolve protocol with T. Again, A may wait for as long as she wants before contacting T (non-deterministic choice). In the resolve protocol, A sends the message r to T. If T has not sent out an abort token before, then T returns the replacement contract r_T, and otherwise T returns the abort token a_T. Analogously, if, after B has sent his commitment to sign the contract, A does not respond, B may contact T to resolve, i.e., B runs the resolve protocol with T similarly to the case for A (non-deterministic choice).

We note that, both in the abort and the resolve protocol, the communication with T is carried out over (direct or scheduled) secure channels.

Figure 1 depicts the formal specification P_A of A. Numbers in brackets denote the priorities of edges. Solid, dashed, and dotted edges correspond to steps in the exchange, resolve, and abort subprotocol, respectively. All vertices have a self-loop with priority 0; for simplicity, they are not drawn. The specification of B and T can be defined similarly as trees P_B and P_T, respectively (see [11] for P_T). The specification of the ASW protocol for the case that A and T are honest but B is dishonest (and hence, B's behavior is determined by the intruder) is the tuple $Pr_{ASW} = (\{A, T\}, \{B\}, \{A, B, T, k_A, k_B, k_B^{-1}, k_T\}, \{P_A, P_T\})$.

The Security-specific Concurrent Game Structure. The concurrent game structure S_{Pr} (the *security-specific concurrent game structure*) induced by a protocol $Pr = (\mathcal{H}, \mathcal{D}, \mathcal{K}, \{P_a\}_{a\in\mathcal{H}})$ and the Dolev-Yao intruder describes the possible moves of all players—honest principals, scheduled secure channels, and the Dolev-Yao intruder—in every state and the effects of these moves, and hence, the execution of the protocol along with the intruder. We now sketch the definition of S_{Pr} (see [11] for the full definition).

Players. The set of players of S_{Pr} consists of all honest principals, the scheduled secure channels, and the Dolev-Yao intruder, who subsumes the dishonest principals and is denoted by $\mathcal{I}$. (Direct secure channels immediately deliver the message written to them, and therefore, they do not need to be modeled as players.)



$$
\begin{array}{lcl}
me_1 & = & \mathsf{sig}[k_A, \langle k_A, k_B, k_T, \mathsf{contract}, \mathsf{hash}(N_A)\rangle] \\
me_2 & = & \mathsf{sig}[k_B, \langle me_1, \mathsf{hash}(x)\rangle] \\
ma_1 & = & \mathsf{sig}[k_A, \langle \mathsf{abort}, me_1\rangle] \\
ma_2 & = & \mathsf{sig}[k_T, \langle \mathsf{abort}, ma_1\rangle] \\
mr_1 & = & \langle me_1, me_2\rangle \\
mr_2 & = & \mathsf{sig}[k_T, mr_1] \\
mr & = & \mathsf{sig}[k_T, \langle me_1, \mathsf{sig}[k_B, \langle me_1, y\rangle]\rangle]
\end{array}
$$

Figure 1. The originator of the ASW protocol specified by a tree.

States. A state q of S_{Pr} describes the current state of the protocol execution (including the state of the intruder). More precisely, a state consists of the following components: (i) a finite set $\mathcal{K}^q$ of messages (the intruder knowledge in q); (ii) for each honest principal $a \in \mathcal{H}$, an a-instance P_a^q (the state of principal a); (iii) for each honest principal $a \in \mathcal{H}$ its current input, i.e., a multi message $\mathbf{m}_a^q$ with domain $\mathsf{ch}(\mathbf{m}_a^q) = \mathcal{C}(\mathcal{P}, a)$ (i.e. $\mathbf{m}_a^q$ is a mapping from $\mathcal{C}(\mathcal{P}, a)$ to $\mathcal{M}_\circ$); (iv) for each scheduled secure channel $c \in \mathcal{C}(Pr)$ a sequence s_c^q of messages (the sequence of messages on c) as well as a boolean (true/false) indicating whether in the previous step a message was delivered. The *initial state* q^0 of S_{Pr} is defined as follows: the intruder knowledge is $\mathcal{K}$, the state of a, $a \in \mathcal{H}$, is P_a, the input to a is the empty message '$\circ$' on all input channels to a, and the scheduled secure channels are empty, with the booleans being false.

Propositional variables and their evaluation. The set of propositional variables consists of (i) a propositional variable p_a for each constant $a \in \mathcal{A}$ which is true in state q if the intruder can derive a in q, i.e., $a \in d(\mathcal{K}^q)$; (ii) a propositional variable p_v for each vertex v of an honest principal P_a^q, $a \in \mathcal{H}$, which is true in state q if the root of P_a^q is v; (iii) propositional variables empty_c and $\mathsf{delivered}_c$ for every

scheduled secure channel c in Pr. They are true if c is empty and c delivered a message in the previous step, respectively.

Moves and the transition function. The possible moves of an honest principal $a \in \mathcal{H}$ in q are those edges leaving the root of P_a^q which are labeled with an a-rule whose left-hand side matches with the current input to a. The possible moves of a scheduled secure channel c are either to deliver a message (if any) or not to deliver a message. A possible move of the Dolev-Yao intruder in state q is a multi message $\mathbf{m}$ such that $\mathbf{m}(c) \in d(\mathcal{K}^q) \cup \{\circ\}$ (i.e., $\mathbf{m}(c)$ can be derived from the current intruder knowledge) for every channel c the intruder can write to. Note that the intruder has an infinite number of possible moves.

Given a total move at state q, i.e., the combination of the chosen moves in q of all players of S_{Pr} , the components of the successor state q' of q are determined as follows: The knowledge $\mathcal{K}^{q'}$ of the intruder in q' is $\mathcal{K}^q$ plus all messages sent on network channels (by honest principals) and all messages delivered by (direct and scheduled) secure channels to dishonest principals. The new state $P_a^{q'}$ of $a \in \mathcal{H}$ is obtained from P_a^q by making the vertex the edge picked by a is leading to the new root of the tree and applying the substitution obtained in the matching process. The new input of a are the messages delivered to a by the intruder and secure channels according to the total move. The scheduled secure channels are updated in the obvious way.

4 AMC-Model Checking and Main Results

In this section, we summarize the main results of this paper, which concern the decidability and complexity of the AMC-model checking problem induced by protocols and the Dolev-Yao intruder. Let us first define this problem.

Let $Pr = (\mathcal{H}, \mathcal{D}, \mathcal{K}, \{P_a\}_{a\in\mathcal{H}})$ be a protocol and $S_{Pr} = \langle \Sigma_{Pr}, Q_{Pr}, \mathbb{P}_{Pr}, \pi_{Pr}, \Delta_{Pr}, \delta_{Pr}\rangle$ be the concurrent game structure induced by Pr and the Dolev-Yao intruder. We call $\text{PAMC} = \{(Pr, \varphi) \mid Pr$ a protocol and φ an AMC-sentence over Σ_{Pr} and $\mathbb{P}_{Pr}$ such that $(S_{Pr}, q^0) \models \varphi$ where q^0 is the initial state of $S_{Pr}\}$ the *(general) protocol induced AMC-model checking problem*. The size of an instance (Pr, φ) of this problem is defined to be $|Pr| + |\varphi|$.

In a nutshell, our main results are as follows: In general, the above problem is undecidable and we identify two independent sources of undecidability: (1) honest principals that are *non-greedy*, i.e., they may ignore received messages even if these messages conform to the protocol specification, and (2) honest principals which may read messages from the Dolev-Yao intruder (i.e., dishonest principals) over scheduled secure channels; in this case, we speak of a *dishonest scheduled secure channel containing protocol* (*dssc-containing protocol*). In both cases we obtain undecidability even for very simple, fixed AMC-formulas. These results show that to obtain decidability it is necessary to restrict to dssc-free protocols with only greedy honest principals, unless other restrictions are imposed on protocols. For this class of protocols and for an expressive fragment of AMC (the $\mathcal{I}$*-monotone* fragment), we establish decidability and tight complexity bounds.

In the remainder of this section, we first define the classes of protocols mentioned above and the fragment of AMC. We then state the undecidability and complexity-theoretic results, along with proof ideas (see [11] for full proofs).

4.1 Classes of Protocols and Fragments of AMC

We first define greedy and dssc-free protocols and then the fragments of AMC we consider.

Greedy protocols. We first define greedy honest principals. Intuitively, an honest principal is greedy if it does not ignore messages in case they conform to the protocol specification. To define greedy honest principals formally, we need to introduce consuming rules. An a-rule $\mathbf{r} \Rightarrow \mathbf{s}$ is *consuming* if $\mathsf{ch}(\mathbf{r}) \neq \emptyset$. Intuitively, if principal a performs a consuming rule, then the form of the incoming messages matters. Now, an a-instance (honest principal) P of Pr is *greedy* if for all vertices v of P the outgoing edges of v labeled with consuming rules have priorities strictly higher than the priority of the self-loop of v (if any). Informally speaking, when a greedy principal can read a message using some consuming rule, then he has to apply such a rule (rather than the self-loop), and hence, as a result moves to another vertex. A protocol Pr is *greedy* if all of its a-instances (honest principals), for $a \in \mathcal{H}$, are. Assuming a protocol to be greedy is a realistic assumption, since in typical protocol specifications honest principals will not ignore messages if they conform to the messages they expect to receive. We note that the ASW protocol as specified in Section 3 is greedy.

Dssc-free protocols. We call a protocol Pr *dishonest scheduled secure channel free (dssc-free)* if no honest principal in Pr uses a scheduled secure channel from a dishonest principal as an input channel, i.e., $\mathsf{ch}(Pr) \cap \mathsf{Sch}(\mathcal{D}, \mathcal{H}) = \emptyset$. Otherwise, we call a protocol *dssc-containing*. Note that scheduled secure channels from honest to dishonest and honest principals are still allowed as well as direct secure channels between dishonest/honest principals. Since direct secure channels from dishonest principals are completely controlled by the adversary, these channels provide the adversary with more power than scheduled secure channels. Hence, excluding scheduled secure channels from dishonest principals is not a restriction in terms of the power of the adversary. We note that the ASW protocol as specified in Section 3 is dssc-free.

$\mathcal{I}$-Monotone AMC-formulas. The set of $\mathcal{I}$-monotone AMC-formulas consists of the set of $\mathcal{I}$-positive and $\mathcal{I}$-negative AMC-formulas. An AMC-formula φ is *$\mathcal{I}$-positive* if all subformulas of φ of the form $\langle\langle A\rangle\rangle\psi$ with $\mathcal{I} \in A$ fall under an even number of negations and all subformulas of φ of the form $\langle\langle A\rangle\rangle\psi$ with $\mathcal{I} \notin A$ fall under an odd number of negations, where $\mathcal{I}$ denotes the Dolev-Yao intruder, a player in security-specific CGSs. An AMC-formula φ is *$\mathcal{I}$-negative* if $\neg\varphi$ is $\mathcal{I}$-positive and *$\mathcal{I}$-monotone* if it is $\mathcal{I}$-positive or $\mathcal{I}$-negative.

$\mathcal{I}$-positive, -negative, and -monotone ATL and ATL* formulas are defined in the same way. As shown by Alur et al. [2] (see also Fact 1), every ATL*-formula can be translated into an equivalent AMC-formula. It is not hard to see that the translation preserves the property of being $\mathcal{I}$-positive/-negative, i.e., the translation of an $\mathcal{I}$-positive ($\mathcal{I}$-negative) ATL*-formula yields an $\mathcal{I}$-positive ($\mathcal{I}$-negative) AMC-formula; the same is of course true for (fair) ATL.

For example, while $\langle\langle\mathcal{I}\rangle\rangle\Diamond(p \wedge \neg\langle\langle B\rangle\rangle\Box q)$ is an $\mathcal{I}$-positive ATL-formula for a player $B \neq \mathcal{I}$, the formula $\langle\langle\mathcal{I}\rangle\rangle\Diamond(p \wedge \neg\langle\langle\mathcal{I}\rangle\rangle\Box q)$ is not.

The class of $\mathcal{I}$-monotone AMC-formulas is a proper fragment of the set of all AMC-formulas in terms of expressibility. However, all formulas that we encountered in the literature for specifying game-theoretic security properties of cryptographic protocols, typically formulas in (fair) ATL, are $\mathcal{I}$-monotone. Hence, the restriction to $\mathcal{I}$-monotone formulas does not seem to be a severe restriction from a practical point of view (see Section 4.4 for more details).

In what follows, we denote by PAMC(greedy/non-greedy, dssc-containing/-free, $\mathcal{I}$-positive/-negative/-monotone) the protocol induced AMC-model checking problem where the class of protocols is restricted to those that are (i) greedy/non-greedy and (ii) dssc-containing/-free, and the AMC-formulas considered are $\mathcal{I}$-positive/-negative/-monotone, respectively.

4.2 Undecidability Results

The first theorem identifies non-greedy protocols as a source of undecidability. The theorem holds even for dssc-free protocols and a very simple, fixed $\mathcal{I}$-monotone formula.

Theorem 1 PAMC*(non-greedy, dssc-free, $\mathcal{I}$-positive) is undecidable, and hence, so is* PAMC*(non-greedy, dssc-free, $\mathcal{I}$-negative) and* PAMC*(non-greedy, dssc-free, $\mathcal{I}$-monotone).*

We prove this theorem by a reduction from Post's Correspondence Problem (PCP). For a given PCP instance Π we construct a protocol $Pr = Pr_\Pi$ such that Π has a solution iff $(S_{Pr}, q^0) \models \varphi$ with the $\mathcal{I}$-positive ATL-formula $\varphi = \langle\langle\mathcal{I}\rangle\rangle\Diamond p_{\mathsf{ok}}$ where ok is a constant and p_{ok} the corresponding propositional variable in S_{Pr}. The formula φ requires that $\mathcal{I}$ has a strategy against all other players in S_{Pr} to eventually obtain ok. (Because of Fact 1, stating φ as an ATL-formula is w.l.o.g.) The protocol Pr that we construct consists of two principals: one honest principal test and one dishonest principal pcp. The honest principal test checks whether the sequence of messages sent by pcp (i.e., the intruder) encodes a solution of Π. Since test does not have to be greedy, test can non-deterministically choose (pairs of) messages in the sequence and test whether they satisfy certain conditions on solutions of Π. Principal test outputs ok to the intruder iff such a test is positive. Hence, the intruder has a strategy to always obtain ok iff the sequence of messages that he delivers passes all tests, and hence, encodes a solution.

The following theorem exhibits another source of undecidability, namely dssc-containing protocols. The theorem holds even for greedy protocols and again a very simple, fixed $\mathcal{I}$-monotone formula. The main proof idea is the same as the one for the proof of Theorem 1.

Theorem 2 PAMC*(greedy, dssc-containing, $\mathcal{I}$-positive) is undecidable, and hence, so is* PAMC*(greedy, dssc-containing, $\mathcal{I}$-negative) and* PAMC*(greedy, dssc-containing, $\mathcal{I}$-monotone).*

4.3 Decidability Results

The undecidability results above show that to obtain decidability, one has to restrict protocols to be greedy and dssc-free, unless other restrictions are imposed on protocols. As explained above, these restrictions appear to be reasonable anyway. The following theorem states that for this class of protocols and $\mathcal{I}$-monotone formulas we obtain decidability of the protocol induced AMC-model checking problem.

Theorem 3 *The problem* PAMC(*greedy, dssc-free, $\mathcal{I}$-monotone*) *is decidable. More precisely, the problem* PAMC(*greedy, dssc-free, $\mathcal{I}$-positive*) *is NEXPTIME-complete, and hence,* PAMC(*greedy, dssc-free, $\mathcal{I}$-negative*) *is coNEXPTIME-complete.*

The only question that this theorem leaves open is whether PAMC is decidable also beyond the fragment of $\mathcal{I}$-monotone formulas. As explained before, from a practical point of view the above theorem seems to suffice: all formulas that we encountered in the literature are $\mathcal{I}$-monotone (see Section 4.4).

The idea of the proof of the complexity upper bound stated in Theorem 3 is as follows: The proof consists of four main steps.

In the first step, similar to the case of the modal μ-calculus [9], we associate to a CGS S, a state q of S, and

an AMC-sentence φ (in negation normal form) a two-party parity game $G^{\varphi}_{(S,q)}$ such that $(S, q) \models \varphi$ iff player 0 has a winning strategy in $G^{\varphi}_{(S,q)}$ (see also [18]). A vertex in $G^{\varphi}_{(S,q)}$ is of the form (q, ψ) where q is a state of S and ψ is an (extended) subformula of φ.

In the second step of the proof, we define an equivalence relation $\sim$ on states of the CGS $S = S_{Pr}$ induced by the protocol Pr. We write $q \sim q'$ for states q, q' of S if q and q' are equal up to the messages on input channels of honest principals. We write $(q, \psi) \sim (q', \psi')$ for two vertices (q, ψ) and (q', ψ') of $G^{\varphi}_{(S,q^0)}$ if $q \sim q'$. We call a state q of S *consuming* if for some honest principal a the messages on its input channels can be read by some consuming rule in the next step of a. A vertex (q, ψ) in $G^{\varphi}_{(S,q^0)}$ is *consuming* if q is. Now, we say that a strategy of player 0 is $\sim$*-uniform* if it is memoryless and moreover, for all pairs of $\sim$-equivalent, non-consuming vertices of a certain form player 0 makes the same choices w.r.t. the Dolev-Yao intruder, i.e., the same messages are chosen by player 0 to be sent by the intruder. We prove that if there exists a winning strategy of player 0, then there exists a $\sim$-uniform winning strategy for this player.

In the third step, we prove that the number of vertices in the strategy graph (which is a subgraph of $G^{\varphi}_{(S,q^0)}$) associated to a $\sim$-uniform strategy of player 0 is bounded exponentially in the size of Pr and φ.

This does not imply that the *overall* size of the representation of the strategy graph is exponentially bounded, since the graph might contain large messages. In the fourth step of the proof, we therefore show that the size of the messages in the strategy graph can be bounded exponentially, which also yields an exponential bound for the overall size of the strategy graph itself. The techniques used in this step are similar to those for reachability properties (see, e.g., [17, 12]). However, these techniques have to be adapted to be applicable to strategy graphs induced by parity games. Now, to decide $(S, q^0) \models \varphi$ one guesses a potential, exponentially bounded strategy graph for player 0 and checks whether it is in fact a winning strategy graph for this player.

The complexity lower bound in Theorem 3 is by reduction from a NEXPTIME-complete variant of the tiling problem; we use the one presented in [6].

4.4 Example Properties

We now illustrate the kind of properties that can be expressed in the $\mathcal{I}$-monotone fragment of AMC. Kremer and Raskin [15, 14] were the first to formulate properties of fair exchange protocols, including contract-signing and non-repudiation protocols, in terms of fair ATL, a fragment of ATL* [2], and hence, of AMC [2]. It turns out that all properties that Kremer and Raskin have formulated fall into the $\mathcal{I}$-monotone fragment of AMC, suggesting that the $\mathcal{I}$-monotone fragment of AMC suffices for most properties of interest. In what follows, we formulate some of these properties. For convenience, we use ATL* rather than AMC. By Fact 1, this is w.l.o.g.

The precise formulations of the properties stated by Kremer and Raskin typically depend on the specific protocol analyzed, in particular the kind of propositional variables used. For concreteness, we will therefore consider the specification of the ASW protocol Pr_{ASW}, with honest A and T and dishonest B, as presented in Section 3.

Fairness. According to Kremer and Raskin [15], one version of fairness is defined as follows: A protocol is *unfair* for honest A if dishonest B together with all scheduled secure channels has a strategy to obtain a signed contract from A such that A does not have a strategy to receive a signed contract from B, given that the scheduled secure channels between A and T are fair, i.e., messages on these channels are eventually delivered. To express this property for the ASW protocol, we add to the specification of this protocol (Section 3) another honest principal, the "watch dog", which is used to check whether B obtains a signed contract from A (see [11] for details). If such a signed contract is sent to the watch dog, it goes into the vertex $\mathsf{B}^{\mathsf{has}}_{\mathsf{contract}}$. Now, the following $\mathcal{I}$-positive ATL*-formula expresses unfairness for A in the ASW protocol where $\mathcal{SC} = \mathsf{sch}(\{A, B, T\}, \{A, B, T\}) \cap \mathsf{ch}(Pr)$ is the set of all scheduled secure channels used in Pr:

$$\langle\!\langle \mathcal{I}, \mathcal{SC} \rangle\!\rangle \Diamond (p_{\mathsf{B}^{\mathsf{has}}_{\mathsf{contract}}} \wedge \neg \langle\!\langle A \rangle\!\rangle (\varphi_{fairSch} \rightarrow \Diamond \varphi_{\mathsf{A}^{\mathsf{has}}_{\mathsf{contract}}}))$$

where

$$\varphi_{fairSch} = \bigwedge_{c \in \mathcal{SC}'} (\Box\Diamond\neg\mathsf{empty}_c) \rightarrow (\Box\Diamond\mathsf{delivered}_c)$$

for $\mathcal{SC}' = \mathsf{sch}(\{A, T\}, \{A, T\})$ says that the scheduled secure channels between A and T are fair [2] (if c is infinitely often non-empty, then it infinitely often delivers a message) and $\varphi_{\mathsf{A}^{\mathsf{has}}_{\mathsf{contract}}} = p_{\mathsf{contract}} \vee p_{\mathsf{resolved}_1} \vee p_{\mathsf{resolved}_2}$ says that A is in vertex $\mathsf{contract}$, $\mathsf{resolved}_1$, or $\mathsf{resolved}_2$ (Figure 1), and hence, has a standard or replacement contract. See [11] for a formulation of a stronger version of fairness in ($\mathcal{I}$-monotone) ATL*.

Timeliness. According to Kremer and Raskin [15], a protocol is timely for honest A if A has a strategy to finish the protocol while preserving fairness. Again, the scheduled secure channels (at least those between A and T) are required to be fair. Formally, timeliness for A is expressed by the following $\mathcal{I}$-negative ATL*-formula where $\varphi_{\mathsf{A}^{\mathsf{has}}_{\mathsf{contract}}} = p_{\mathsf{contract}} \vee p_{\mathsf{resolved}_1} \vee p_{\mathsf{resolved}_2} \vee p_{\mathsf{aborted}}$ says that A finished her protocol run (Figure 1):

$$\langle\!\langle A \rangle\!\rangle (\varphi_{fairSch} \rightarrow \Diamond (\varphi_{\mathsf{Afinished}} \wedge (\neg \varphi_{\mathsf{A}^{\mathsf{has}}_{\mathsf{contract}}} \rightarrow \neg \langle\!\langle \mathcal{I}, \mathcal{SC} \rangle\!\rangle (\varphi_{fairSch} \wedge \Diamond p_{\mathsf{B}^{\mathsf{has}}_{\mathsf{contract}}})))).$$

Balance. According to Kremer and Raskin [15] (see also [7]), a protocol is *unbalanced* for honest A if at some stage of the protocol run dishonest B has both a strategy to obtain a signature from A and a strategy to prevent A from obtaining a signature from B. Again, the scheduled secure channels between A and T are required to be fair. Unbalanced for A can be formulated as an $\mathcal{I}$-positive ATL*-formula as follows:

$$\langle\!\langle \mathcal{I}, \mathcal{SC}, A, T \rangle\!\rangle \varphi_{fairSch} \wedge \Diamond(\varphi_{getContract} \wedge \varphi_{prevent})$$

where

$$\varphi_{getContract} = \langle\!\langle \mathcal{I}, \mathcal{SC}'' \rangle\!\rangle \varphi_{fairSch} \rightarrow \Diamond p_{\mathsf{B}^{\mathsf{has}}_{\mathsf{contract}}},$$

$$\varphi_{prevent} = \langle\!\langle \mathcal{I}, \mathcal{SC}'' \rangle\!\rangle \varphi_{fairSch} \rightarrow \Diamond(\neg\langle\!\langle A \rangle\!\rangle(\varphi_{fairSch} \rightarrow \Diamond \varphi_{\mathsf{A}^{\mathsf{has}}_{\mathsf{contract}}}))$$

for $\mathcal{SC}'' = \mathsf{sch}(\{B, T\}, \{B, T\}) \cap \mathsf{ch}(Pr)$. Similarly, abuse-freeness as defined by Kremer and Raskin, other variants of the properties specified above, and properties of non-repudiation protocols [14] can be formulated in the $\mathcal{I}$-monotone fragment.

5 Conclusion

In this paper, we studied the AMC-model checking problem over infinite-state concurrent games structures induced by protocols and the Dolev-Yao intruder. We proved that to obtain decidability it is necessary to restrict to greedy and dssc-free protocols, unless other restrictions are imposed on protocols. From a practical point of view, this seems to be a reasonable class of protocols. For this class of protocols and the $\mathcal{I}$-monotone fragment of AMC, which contains all game-theoretic properties formulated, for example, by Kremer and Raskin, we obtained decidability of the model checking problem with tight complexity bounds. The complexity upper bounds were obtained by combining techniques from the theory of infinite games and cryptographic protocol analysis in a novel and quite modular way, and hence, it is quite likely that results for reachability properties, e.g., taking algebraic properties into account, also carry over to our setting. The main technical question left open by our result is whether the model checking problem is decidable also for full AMC.

References

[1] R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *FOCS'97*, pages 100–109. IEEE Computer Society Press, 1997.

[2] R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. To appear in Journal of the ACM. Available from http://www.cis.upenn.edu/ alur/Jacm02.pdf, 2002.

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

[4] N. Asokan, V. Shoup, and M. Waidner. Asynchronous protocols for optimistic fair exchange. In *S&P 1998*, pages 86–99. IEEE Computer Society, 1998.

[5] M. Boreale. Symbolic trace analysis of cryptographic protocols. In *ICALP 2001*, volume 2076 of *LNCS*, pages 667–681. Springer, 2001.

[6] D. Calvanese, G .De Giacomo, M. Lenzerini, M.Y. Vardi. View-based query containment. In *PODS 2003*, pages 56–67. ACM Press, 2003.

[7] R. Chadha, M. Kanovich, and A.Scedrov. Inductive methods and contract-signing protocols. In *CCS 2001*, pages 176–185. ACM Press, 2001.

[8] R. Corin, S. Etalle, and A. Saptawijaya. A Logic for Constraint-based Security Protocol Analysis. In *S&P 2006*, pages 155–168. IEEE Computer Society, 2006.

[9] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002.

[10] D. Kähler and R. Küsters. Constraint Solving for Contract-Signing Protocols. In *CONCUR 2005*, volume 3653 of *LNCS*, pages 233–247. Springer, 2005.

[11] D. Kähler, R. Küsters, and T. Truderung. Infinite State AMC-Model Checking for Cryptographic Protocols. Technical report IFI-0702, CAU Kiel, Germany, 2007. Available from http://www.informatik.uni-kiel.de/uploads/tx_publication/KKT-CAU-TR-0702.pdf

[12] D. Kähler, R. Küsters, and T. Wilke. Deciding Properties of Contract-Signing Protocols. In *STACS 2005*, volume 3404 in LNCS, pages 158–169. Springer, 2005.

[13] D. Kähler, R. Küsters, and T. Wilke. A Dolev-Yao-based Definition of Abuse-free Protocols. In *ICALP 2006*, volume 4052 of *LNCS*, pages 95–106. Springer, 2006.

[14] S. Kremer and J.-F. Raskin. A game-based verification of non-repudiation and fair exchange protocols. In *CONCUR 2001*, volume 2154 of *LNCS*, pages 551–565. Springer, 2001.

[15] S. Kremer and J.-F. Raskin. Game analysis of abuse-free contract signing. In *CSFW 2002*, pages 206–220. IEEE Computer Society, 2002.

[16] J. K. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *CCS 2001*, pages 166–175. ACM Press, 2001.

[17] M. Rusinowitch and M. Turuani. Protocol Insecurity with Finite Number of Sessions is NP-complete. In *CSFW-14*, pages 174–190. IEEE Computer Society, 2001.

[18] S. Schewe, B. Finkbeiner. Satisfiability and Finite Model Property for the Alternating-Time mu-Calculus. In *CSL 2006*, volume 4207 of *LNCS*, pages 591-605. Springer 2006.

[19] V. Shmatikov and J. Mitchell. Finite-state analysis of two contract signing protocols. In *TCS*, 283(2):419–450, 2002.