Deciding Epistemic and Strategic Properties of Cryptographic Protocols*

Henning Schnoor

Institut für Informatik, Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany
schnoor@ti.informatik.uni-kiel.de

Abstract. We propose a new, widely applicable model for analyzing knowledge-based (epistemic) and strategic properties of cryptographic protocols. We prove that the corresponding model checking problem with respect to an expressive epistemic strategic logic is decidable. As corollaries, we obtain decidability of complex security properties including coercion-resistance of voting protocols, accountability of protocols using a trusted third party, and abuse-freeness of contract signing protocols.

Introduction

In design and verification of cryptographic protocols, symbolic techniques [DY83] have proven very successful. A breakthrough result in this area is that secrecy properties of protocols can be decided in coNP, even if the adversary is allowed to send arbitrarily complex terms [RT03]. Recently, game-based properties of cryptographic protocols have been studied [KR02]. Such properties are relevant e.g., for contract signing [BOGMR90,ASW98,GJM99] and non-repudiation [KR03] protocols, and can naturally be expressed in Alternating-Time Temporal Logic (ATL, [AHK02]), a logic explicitly designed to reason about strategies. Decidability results for such properties have been obtained in [KKT07,KKW09]. However, existing symbolic approaches for strategic analysis have the following limitations:

(i) The models and logics that have been applied cannot express *epistemic* properties, i.e., properties concerned with knowledge of principals as, e.g., abuse-freeness of contract-signing [KKW06] or anonymous broadcast [Cha88]. Similarly, they only consider *complete-information strategies*: Honest principals and the adversary base their decisions on complete knowledge about the current state, including private messages between other principals and cryptographically hidden secrets. Thus, capabilities of all parties are over-approximated, potentially leading to both "false positives" and "false negatives" in the security analysis.

(ii) They do not handle *probabilistic* protocols that allow random *decisions*. These are essential for some security goals [ASW09] and can be used to model random routing in anonymity protocols.

* This paper is the full version of [Sch12].

We propose an approach overcoming these shortcomings by a thorough treatment of knowledge and probabilism. To express security properties, we use QAPI [Sch10a], a very expressive extension of ATL*. In addition to epistemic and probabilistic aspects, QAPI allows explicit reasoning and quantification of strategies similarly to strategy logic [CHP07]. This allows to express dependencies between strategies of different coalitions, as for example knowledge that one coalition has about the behavior of others. Our contributions are as follows:

1. We define a symbolic model for protocol analysis treating explicit knowledge, incomplete information, and probabilistic protocols.
2. We show that the question whether a protocol satisfies a security property (specified by a QAPI-formula) is decidable for active and passive adversaries.

Our decidability result holds for finitely many parallel sessions, it is well-known that even very simple security properties are undecidable for the unbounded session case [EG83]. Our proof implies that relevant strategies can always be finitely represented, hence can be implemented in software.

As a toy example, we consider a coin-flipping protocol: Bob randomly chooses a bit $b_1 \in \{0,1\}$ and a random string N, and sends $hash(\langle b_1, N\rangle)$ to Alice. Alice randomly chooses $b_2 \in \{0,1\}$ and sends b_2 to Bob. He then sends N and b_1 to Alice, who verifies that these match the hash. The security property is that neither Alice nor Bob can dictate the outcome of the protocol, which is the bit $b_1 \oplus b_2$. This is only true since Alice's b_2 may not depend on the secret value b_2, hence security of the protocol can only be shown with an epistemic approach. In addition to this toy example, we give the following applications:

1. We show how accountability and verifiability of protocols that involve a trusted third party and coercion-resistance of voting protocols can be expressed in our model, implying decidability of these properties.
2. We prove that abuse-freeness of contract-signing protocols can be formalized in our model, and obtain decidability as a corollary. This resolves an open question from [KKW06].
3. We show how coercion-resistance of voting protocols can be expressed in our model. In addition to the epistemic and strategic properties, this property has a probabilistic aspect. Again, we obtain decidability as a corollary.

Related Work. In the above-mentioned [KKT07], a decision algorithm for (non-epistemic, complete-information, non-probabilistic) strategic properties of protocols is given. In [KKW09] a decidability result for a strategic property (balance) of contract-signing protocols was established. This result follows from our decidability result. In the very influential paper [BAN90], a logic for authentication protocols was introduced, which models knowledge gained during the run of an authentication protocol. Among the many follow-ups are [AT91,BM93,JYH].

[ASW09] defines a model for probabilistic protocols, however no decidability result is proven. We significantly generalize that model: First, we treat security goals that involve epistemic aspects. Second, we treat arbitrary term signatures

with equational theories instead of only nonces and signatures as in [ASW09]. Further, we allow arbitrarily complex terms.

Organization. In Section 1 and 2, we define syntax and semantics of our protocol model. In Section 3, we briefly recall the semantics of the logic QAPI. Section 4 contains our main result: The question whether a given protocol satisfies a given security property (i.e., a formula) is decidable. Section 5 contains the above-mentioned applications. In Section 6, we give the proof of our main result. We conclude in Section 7.

This paper is the full version of [Sch12].

1 Syntax: Specifying a Protocol

1.1 Two Examples

The Coin-Flipping Protocol In the coin-flipping protocol (cp. Introduction), Bob chooses his bit first and thus cannot dictate the outcome of the protocol (as Alice verifies consistency with the hash value). We therefore consider the more interesting case of dishonest Alice: Only the hash function prevents her from dictating the result unilaterally. Hence we let Alice be the adversary, and assume that only Bob follows the protocol, his specification is presented in the left-hand side of Figure 1. Dashed lines represent messages received by Bob, solid ones are messages sent by him. The message $\langle \alpha, N \rangle$ is a pair containing the bit α and the random string N.

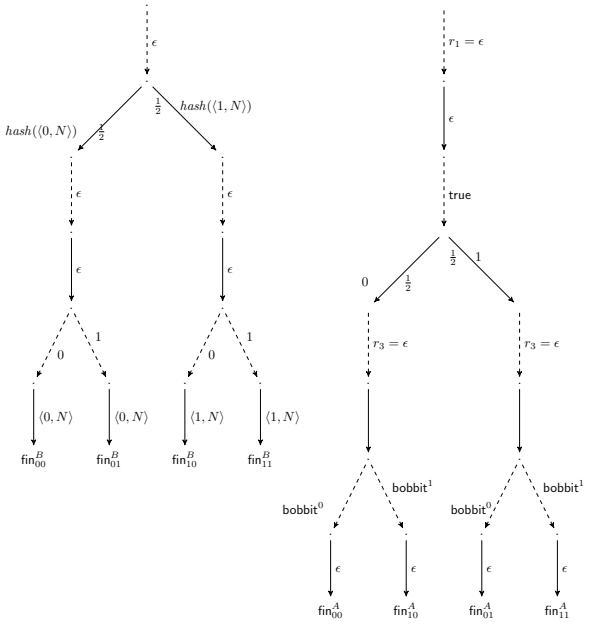

Fig. 1. Coin-Flipping Protocol: Specification

The probabilities $\frac{1}{2}$ express that Bob chooses the bits 0 and 1 with probability $\frac{1}{2}$ each. Omitted probabilities are 1. Different messages from Alice (0 or 1) lead to different follow-up states for Bob. We omit error states for syntactically incorrect incoming messages, etc. Since our model is concurrent, we add a dummy sequence for the step when Alice is active.

In our formalism, this property is expressed as $\forall_3 \mathrm{S} \neg \langle\langle \mathcal{A} : \mathrm{S} \rangle\rangle^{>0.5} \Diamond \left(\mathsf{fin}^B_{00} \vee \mathsf{fin}^B_{11} \right)$. The formula expresses that for every strategy S (that cannot break the hash function, this is specified by the index 3), if the adversary Alice follows S, she only

has a probability of $\frac{1}{2}$ to reach a state in which both random bits are the same and hence the result bit is 0; the 1-case is symmetric.

We now show how Alice's role can be specified in our model. For simplicity, the graphical representation in the right-hand side of Figure 1 uses the terms r_i for the message Alice received in the ith protocol step (our general notation will be introduced below). The final receive step made by Alice is the most important one: Here she receives the pair $\langle b_1, N\rangle$ from Bob. Alice now checks that Bob did not cheat (i.e., that this pair is indeed consistent with the hash value received earlier in the protocol run), and computes the result of the coinflip. For this, she uses the following test: For $\alpha \in \{0, 1\}$, the "test" bobbit^α is the conjunction $(r_2 = hash(r_4)) \bigwedge (\Pi_1(r_4) = \alpha)$, this test is true iff the pair sent by Bob in step 4 matches the earlier sent hash value and the bit contained in Bob's commitment is α. Here the operator Π_1 denotes extraction of the first element of a pair. Depending on this test and on her own previously chosen bit, Alice then moves into one the states fin^A_{00}, fin^A_{01}, fin^A_{10}, fin^A_{11}, where the bit combinations $\alpha\beta$ denote the 4 possible choices of bits by Alice and Bob (the first bit is Bob's random choice, the second one Alice's).

The test true used in the first receive step when the hash value of Bob's pair $\langle b_1, N\rangle$ is received always returns true: At this point of the protocol run, no tests are performed, the value is merely stored for later reference.

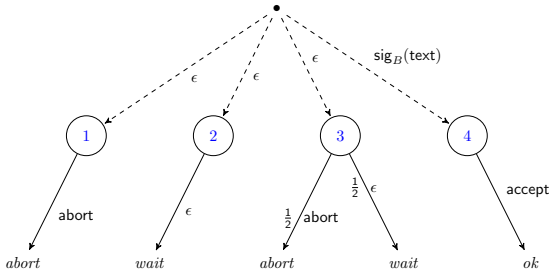

Fig. 2. Protocol State Example

Wait State in a Contract Signing Protocol Consider the protocol excerpt in Figure 2. There are two possible incoming messages: The empty term ϵ and a cryptographic signature of some text. If ϵ is received, there are three possible reactions: 1. send an abort-message, and move to an "aborted" state, 2. move into a waiting state, 3. randomly choose between the first two alternatives. If the signature is received, an *ok*-state is reached and an accept message sent. The random choice in the example is contrived, however there are protocols where randomized decisions are essential, e.g., the contract signing protocol introduced in [ASW09], the coinflipping protocol discussed above, and random routing.

1.2 Formalizing Protocol States

Our formal protocol definition is the natural one. The most important aspect is how principals react to incoming messages. These reactions depend on observable properties of the message. Such properties are modeled as *tests*. Let IDs be a set of identities in a PKI. Let $\mathcal{N}$ be the disjoint union of the infinite sets

$\mathcal{N}_\mathcal{A}$ and $\mathcal{N}_i$ for each $i \in$ IDs (nonces generated by the adversary and honest participants). Let $X = \{x_1, x_2, \dots\}$ be an infinite set of variables. Let Σ^{t} be a term signature containing function symbols with assigned arities representing cryptographic primitives. The set of terms $\mathcal{T}_{\Sigma^{\mathsf{t}}}$ is defined as usual inductively on $\mathcal{N}$, X, and symbols from Σ^{t}. We assume that for each $i \in$ IDs, there are terms i, pk_i and sk_i, denoting the name, public and private key of i, and that Σ^{t} contains operations $\langle .,. \rangle$ to construct tuples and Π_i to access their components. For $C \subseteq$ IDs, the set T_C is the set of terms constructable from Σ^{t} and $X \cup \bigcup_{i \in C} \mathcal{N}_i \cup \mathcal{N}_\mathcal{A}$ where no sk_i for $i \notin C$ appears. We call these terms *C-terms*. These can be constructed with access to the secret keys and nonces of members of C. We write $T_\mathcal{A}$ instead of T_C if C is clear from the context, to highlight that these terms can be constructed by the adversary when the identities in C are corrupted.

We write $t[t'_1/x_1, \dots, t'_n/x_n]$ for the term obtained from t by simultaneously replacing every occurrence of the variable x_i with the term t'_i.

$$\mathsf{dec}_{\mathsf{sk}_{x_i}}\left(\mathsf{enc}_{\mathsf{pk}_{x_i}}(x_t)^{x_r}\right) = x_t$$
$$\mathsf{verify}\left(\mathsf{sig}_{\mathsf{sk}_{x_i}}(x_t)^{x_r}, x_t, \mathsf{pk}_{x_i}\right) = \mathsf{ok}$$
$$\text{for } i \in \{1,2\}, \Pi_i\langle t_1, t_2\rangle = t_i$$

Fig. 3. Example equational theory

We assume a convergent equational theory E. See Figure 3 for an example theory with public-key encryption, signatures, and pairing; in the equations x_i refers to an identity, x_t is a term, and x_r represents randomization nonces. The (uniquely determined) *normal form* of a term t, denoted with $[[t]]$, is obtained by exhaustive application of equations from E. In the example, if $t = \mathsf{dec}_{\mathsf{sk}_A}\left(\mathsf{enc}_{\mathsf{pk}_A}(\mathsf{abort})^r\right)$, then $[[t]] = \mathsf{abort}$.

Formally, an *equation* over Σ^{t} is a pair of Σ^{t}-terms (l, r), written as $l = r$ (our equations are "oriented," where intuitively, we write the "more complicated" term on the left-hand side). An *equational theory* E over Σ^{t} is a set of equations over Σ^{t}. For example, the equation $\mathsf{dec}_{\mathsf{sk}_{x_i}}\left(\mathsf{enc}_{\mathsf{pk}_{x_i}}(x_t)^{x_r}\right) = x_t$ in the theory from Figure 3 models that when encrypting a term x_t with the public key of an identity x_i with randomness x_r, and decrypting the term with the private key of the same identity, then the result is x_t again. This equation is a "simplification rule," transforming a complex term (the ciphertext) into a simpler term (the plaintext). E induces a *rewrite relation* $\twoheadrightarrow_{\mathrm{E}}$ on terms, where $t_1 \twoheadrightarrow_{\mathrm{E}} t_2$ if t_2 can be obtained from t_1 by applying a rule in E in the natural way.

With $\twoheadrightarrow_{\mathrm{E}}^*$, we denote the reflexive and transitive closure of $\twoheadrightarrow_{\mathrm{E}}$, and $\equiv_{\mathrm{E}}$ is the closure of $\twoheadrightarrow_{\mathrm{E}}^*$ under transitivity, symmetry, and application of function symbols (i.e., rules can be applied in subterms). Terms t_1 and t_2 are called E-*equivalent*, if $t_1 \equiv_{\mathrm{E}} t_2$. The relation $\twoheadrightarrow_{\mathrm{E}}$ is *confluent*, if for all t, t_1, t_2 with $t \twoheadrightarrow_{\mathrm{E}}^* t_1$ and $t \twoheadrightarrow_{\mathrm{E}} t_2$, there is some t' with $t_1 \twoheadrightarrow_{\mathrm{E}} t'$ and $t_2 \twoheadrightarrow_{\mathrm{E}} t'$. The relation $\twoheadrightarrow_{\mathrm{E}}$ is *terminating* if there is no infinite sequence of terms t_1, t_2, ... such that for all i we have $t_i \neq t_{i+1}$ and $t_i \twoheadrightarrow_{\mathrm{E}} t_{i+1}$. The theory E is *convergent* if $\twoheadrightarrow_{\mathrm{E}}$ is both confluent and terminating.

E is a *convergent subterm theory* [AC06] if for each $(l, r) \in$ E, r is a subterm of l or a constant, and E is convergent. Convergent subterm theories cover many

interesting applications including the behavior of usual cryptographic primitives. Many decision problems for such theories are decidable [AC06].

A term $t \in \mathcal{T}_{\Sigma^t}$ is in *normal form* or a *message* if $t \twoheadrightarrow_{\mathrm{E}}^* t'$ implies $t = t'$. If $\twoheadrightarrow_{\mathrm{E}}$ is convergent, then for each term t there is a unique term t' in normal form such that $t \twoheadrightarrow_{\mathrm{E}}^* t'$, we denote this term with $[[t]]$. If $\twoheadrightarrow_{\mathrm{E}}$ is convergent, then terms are equivalent if and only if they have the same normal form.

Definition.[KKW06] For a set C of identifies, an *atomic C-test* is a pair (M, M') of C-terms where exactly one variable x appears in M and M'. A message m *satisfies* (M, M'), if $M[m/x] \equiv_{\mathrm{E}} M'[m/x]$. A *$C$-test* is a Boolean combination of atomic C-tests, with the obvious semantics. Messages m and m' are *C-indistinguishable* if there is no C-test that exactly one of them satisfies.

The definition extends to sequences of messages. Indistinguishibility is also known as *static equivalence* [AF01]. We now define protocol states. These specify how an incoming message is handled in a protocol: Depending on properties of the message (modeled with tests), there are different possible choices how a principal can react. In randomized protocols, these choices are probability distributions over actions, where an action consists of a reply message and a state change. In the definition below, the parsing sequence corresponds to the dashed lines in the example above; the send sequence formalizes the solid lines. A state hence consists of the dashed lines originating at the same point plus their solid successors. The dashed lines are labeled with tests (the example also uses ϵ as the test satisfied by the empty message only), the solid lines are labeled with terms sent as replies and the probabilities with which they are chosen.

Definition. A *protocol state* w is a special symbol Finished or consists of

- a *parsing sequence* $t_1, \ldots, t_k$, where each t_i is a test,
- a *send sequence* $(s_{1,1}, \alpha_{1,1}), \ldots, (s_{1,l}, \alpha_{1,l}), (s_{2,1}, \alpha_{2,1}), \ldots, (s_{k,l}, \alpha_{k,l})$, where each $s_{i,j}$ is a term, and $\alpha_{i,j} \geq 0$ is a rational number with $\sum_{j=1}^{l} \alpha_{i,j} = 1$ for all $i \in \{1, \ldots, k\}$.

If w is not Finished, then a number $i \in \{1, \ldots, k\}$ is a *choice in* w, and l is the *randomization degree* of w. We also call such states *regular protocol states*.

A protocol role is a program for a principal (see Figure 1). It combines states into a tree, with different possible actions in each state. We assume sufficiently many copies of Finished, so that a protocol role may have different final states. We model a single protocol session, a finite number of concurrent sessions can be implemented by expressing the resulting interleaving protocol in our model.

Definition. A *protocol role* $\mathcal{R}$ consists of a finite directed tree (V, E), where V is a set of protocol states and E is a set of labeled edges such that:

- If $w \in V$ has k choices and randomization degree l, then w has $k \cdot l$ successors with edges labeled with (i, j) for $i \in \{1, \ldots, k\}$ and $j \in \{1, \ldots, l\}$.
- If $w \in V$ is a copy of Finished, then w does not have any successor.
- There is an identity $i \in$ IDs such that every subterm appearing in $\mathcal{R}$ is an i-term, i is also called the *identity* of $\mathcal{R}$.

Requiring an identify for each role ensures it uses a single private key only. A *k-roles protocol* is a tuple $Pr = (\mathcal{R}_1, \ldots, \mathcal{R}_k)$, where each $\mathcal{R}_i$ is a protocol role.

2 Semantics: Executing a Protocol

We first informally describe the execution of protocols. Again, k is the number of honest protocol participants. Principals send and receive messages consisting of $(k+1)$-ary tuples. An incoming message contains in component i the message from principal $i \in \{1, \ldots, k\}$ or the adversary if $i = k + 1$. Analogously, the message sent in each round is a tuple with $(k + 1)$ entries, where the i-th entry is intended to be sent to principal i, or to the adversary if $i = k + 1$.

An honest principal $h \in \{1, \ldots, k\}$ operates as follows: In each state, h analyzes the incoming message tuple, and checks for each test from the parsing sequence whether the message satisfies it. The test takes the history of the protocol run into account, i.e., is applied to the sequence of messages received so far by h. If test t_c is satisfied, a number $d \in \{1, \ldots, l\}$ is chosen randomly using the distribution specified by $\alpha_{c,1}$, ..., $\alpha_{c,l}$, and the term $s_{c,d}$ is the reply sent by h. Using a variable referring to the sequence of previously received messages, the reply may depend on previously received messages. The local successor state is determined by the outgoing edge (c, d) of the current one. If the incoming message satisfies more than one of the tests, the principal makes a *strategic choice* by choosing the one to apply. This occurs in the above contract-signing example if the incoming message is the empty term. To avoid cumbersome case distinctions, we require that for every message, there must be a test that it satisfies.

The adversary may send arbitrary terms that he can construct using the secret keys from *corrupted* identities.

2.1 Formal Protocol Model as a Concurrent Game Structure

The formal model combines a set of global states of a protocol run (containing the protocol state of every participant) with the possible actions (*"moves"*) and consequences thereof for every party. A usual way to specify strategic situations as this one are concurrent game structures (CGS). We use the definition from [Sch10b], which models probabilistic games and incomplete information:

Definition. A *concurrent game structure* is a tuple $\mathcal{C} = (\Sigma, Q, \mathbb{P}, \pi, \Delta, \delta, \mathsf{eq})$:

- Σ and $\mathbb{P}$ are non-empty, finite sets of *players* and *propositional variables*, Q is a non-empty set of *states*,
- $\pi \colon \mathbb{P} \to 2^Q$ is a *propositional assignment* (p is true in all states from $\pi(p)$),
- Δ is a *move function* assigning to each state q and player a a nonempty 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 mapping each $a \in A$ to a move $c(a) \in \Delta(q, a)$.
- δ is a probabilistic *transition function* which for each state q and (Σ, q)-move c specifies a discrete probability distribution $\delta(q, c)$ on Q (the distribution of the follow-up state of q if all players perform their move as specified by c),

– eq is an *information function* $\mathsf{eq}\colon \{1,\dots,n\} \times \Sigma \to \mathcal{P}(Q \times Q)$, where n is a natural number, and for each $i \in \{1,\dots,n\}$ and $a \in \Sigma$, $\mathsf{eq}(i,a)$ is an equivalence relation on Q. Each $i \in \{1,\dots,n\}$ is called a *degree of information*.

A subset $A \subseteq \Sigma$ is a *coalition of* $\mathcal{C}$. We write $q_1 \sim_{\mathsf{eq}_i(A)} q_2$ for $(q_1,q_2) \in \cap_{a\in A}\mathsf{eq}(i,a)$. If $q_1 \sim_{\mathsf{eq}_i(a)} q_2$, then player a cannot distinguish states q_1 and q_2 (if i denotes the degree of information available to him). Multiple degrees of information allow to dynamically specify the information available to principals, e.g., whether they are regarded as being able to break cryptography, etc.

We define the protocol execution as CGS, which formalizes the mechanisms described earlier. In the state description below, C is the set of corrupted identities, each honest principal $h \in \{1,\dots,k\}$ is in protocol state w_h. For each principal $i \in \{1,\dots,k,\mathcal{A}\}$, the sequence $\mathcal{M}_i$ contains the messages received so far. The sequence $\mathrm{moves}_{\mathcal{A}}$ records the moves performed by the adversary. The numbers c_h and d_h are the strategic and random choices made by h. The propositional variables allow to reason about the local state of honest principals.

Definition. Let $Pr = (\mathcal{R}_1,\dots,\mathcal{R}_k)$ be a protocol. The *CGS induced by* Pr is $\mathcal{C}_{Pr} = (\Sigma, Q, \mathbb{P}, \pi, \Delta, \delta, \mathsf{eq})$, where

– $\Sigma = \{1,\dots,k,\mathcal{A}\}$,
– Q consists of tuples of the form $q = (C, w_1, \mathcal{M}_1, \dots, w_k, \mathcal{M}_k, \mathcal{M}_{\mathcal{A}}, \mathrm{moves}_{\mathcal{A}})$, where $C \subseteq \mathrm{IDs}$, for each $i \in \{1,\dots,k\}$, w_i is a protocol state of $\mathcal{R}_i$, $\mathcal{M}_i$ and $\mathcal{M}_{\mathcal{A}}$ are sequences of messages, and $\mathrm{moves}_{\mathcal{A}}$ is a sequence of terms.
– for each protocol state w occurring in Pr and each $h \in \{1,\dots,k\}$ there is a propositional variable st^h_w which is true in a state q as above iff $w_h = w$,
– for a state q as above where for all $h \in \{1,\dots,k\}$, w_h has k_h choices, randomization degree l_h, parsing sequence $t^h_1, \dots, t^h_{k_h}$ and send sequence $(s^h_{1,1}, \alpha^h_{1,1}), \dots, (s^h_{k_h,l_h}, \alpha^h_{k_h,l_h})$, the available moves are as follows: For $\mathcal{A}$, every term $m_{\mathcal{A}} \in T_{\mathcal{A}}$ is a move, for an honest principal $h \in \{1,\dots,k\}$, the number $c_h \in \{1,\dots,k_h\}$ is a move if and only if $\mathcal{M}_h$ satisfies the test $t^h_{c_h}$. The transition function δ is defined as follows: For the move determined by the adversary move $m_{\mathcal{A}}$ and the principal moves $c_1,\dots,c_k$ and numbers $d_1, \dots, d_k$, where $1 \le d_h \le l_h$, there is a successor state $q' = (C, w'_1, \mathcal{M}'_1, \dots, w'_k, \mathcal{M}'_k, \mathcal{M}'_{\mathcal{A}}, \mathrm{moves}_{\mathcal{A}} \circ m_{\mathcal{A}})$, where
  - w'_h is the successor of w_h in $\mathcal{R}_h$ connected with the edge labeled (c_h, d_h),
  - to define $\mathcal{M}'_j$, we denote with M_i for $i \in \{1,\dots,k,\mathcal{A}\}$ the *message sent by* i, which is $[[s^i_{c_i,d_i}[\mathcal{M}_i/x]]]$ if $i \le k$, or $[[m_{\mathcal{A}}[\mathcal{M}_{\mathcal{A}}/x]]]$ if $i = \mathcal{A}$,
  - for all $i \in \{1,\dots,k,\mathcal{A}\}$, the new sequence $\mathcal{M}'_i$ is obtained by adding to $\mathcal{M}_i$ a $(k+1)$-ary tuple containing in its j-th component the i-th component of M_j (i.e., the term that j sends to i),
  - the probability of this successor state is $\prod_{h=1}^{k} \alpha^h_{c_h,d_h}$.

  If a principal is in a copy of $\mathsf{Finished}$, he only has dummy moves.
– We define three information degrees: For a player $a \in \Sigma$,

  1. $\mathsf{eq}(1,a)$ is the equality relation—this models complete information modulo $\equiv_{\mathrm{E}}$ (since the states only contain normal forms of terms),

2. in $\mathsf{eq}(2, a)$, two states are equivalent if and only if principal a is in the same local state[1], and the component $\mathcal{M}_a$ is the same in both states (this models local information with ability to break cryptography)
3. $\mathsf{eq}(3, a)$ is the equivalence relation where states are equivalent if and only if the principal is in the same local state[1], and components $\mathcal{M}_a$ are a-indistinguishable (C-indistinguishable if $a = \mathcal{A}$).

The message received by a principal in each step is a tuple containing messages from every protocol principal, allowing simulations processing of messages. Messages are immediately delivered to the intended recipients using secure channels. Realistically, use of such channels can be restricted by using *buffer principals* which the adversary may instruct to delay/drop messages.[2] These are modeled as ordinary protocol roles relaying messages, allowing flexible "implementations" of channels and various levels of "adversary activeness:" If a protocol does not use buffers at all, but principals only communicate via the adversary, the adversary is active without restriction. If all communication uses secure channels (with copies sent to the adversary), the adversary is passive. Intermediate degrees can express secure channels to trusted third parties, etc.

For each $C \subseteq \text{IDs}$, there is an *initial state* $q^C_{init} = (C, r_1, \epsilon, r_2, \epsilon, \ldots, r_k, \epsilon, \epsilon, \epsilon)$, where r_i is the root of $\mathcal{R}_i$. In this state, no message has been sent, every principal is in its initial state, and the adversary knows the keys of all identities in C. This models *static corruption*, where a set of identities (fixed before the protocol run) as adversarial. See Section 5.3 for an example of dynamic corruption. We remove all states from $\mathcal{C}_{Pr}$ that cannot be reached from one of the initial states.

We note that there are two ways in which probabilism is relevant in our model: First, protocol specifications may use random decisions as in the coin-flipping protocol. Second, some security properties contain success probabilities. In the coin-flipping protocol, the adversary has a success probability of $\frac{1}{2}$ but not higher, we will sketch a less trivial application in Section 5.4.

3 Probabilistic, epistemic ATL with strategy quantification

To express security goals, we use the ATL*-variant QAPI [Sch10a,Sch10b]. QAPI is not security-specific, but a logic for reasoning about strategic and epistemic properties of general multi-agent systems. QAPI is very expressive and contains several related logics. We only discuss the subset of QAPI relevant for this paper, however our results hold for the complete logic. [Sch10a] contains detailed discussions and comparisons to as well as references to many related logics.

[1] The local state of $\mathcal{A}$ consists of the set C and the sequence $\text{moves}_{\mathcal{A}}$.

[2] In order to avoid infinite protocol runs, we forbid rounds in which the adversary delays every available channel in the obvious way.

3.1 Formulas

QAPI extends ATL* with epistemic features, probabilities, and explicit strategies. Formulas may contain variables $\mathrm{S}_1, \ldots, \mathrm{S}_n$ referring to strategies, these will be bound by quantifiers. This allows explicit reasoning about strategies.

Definition. The set of *QAPI-formulas* for a CGS $\mathcal{C}$ is defined as follows:

- A propositional variable of $\mathcal{C}$ is a state formula, conjunctions and negations of state (path) formulas for $\mathcal{C}$ are state (path) formulas for $\mathcal{C}$,
- every state formula is a path formula,
- if $A_1, \ldots, A_n$ are coalitions, $\blacktriangleleft$ is one of $\leq, <, \geq, >$, ψ is a path formula, and $\mathrm{S}_1, \ldots, \mathrm{S}_n$ are variables for strategies, then $\langle\langle A_1 : \mathrm{S}_1, \ldots, A_n : \mathrm{S}_n \rangle\rangle^{\blacktriangleleft \alpha} \psi$ is a state formula,
- if A is a coalition, i is a degree of information, and ψ is a state formula, then $\mathcal{K}_i^A \psi$ is a state formula,
- If φ_1 and φ_2 are path formulas, then so are $\mathsf{X}\varphi_1$, $\mathsf{P}\varphi_1$, $\mathsf{X}^{-1}\varphi_1$, and $\varphi_1 \mathsf{U} \varphi_2$.

Intuitively, $\langle\langle A_1 : \mathrm{S}_1, \ldots, A_n : \mathrm{S}_n \rangle\rangle^{\blacktriangleleft \alpha} \psi$ expresses that if the coalitions A_1, $\ldots$, A_n play the strategies referred to by S_1, $\ldots$, S_n, then for every possible behavior of the remaining players, the probability that the resulting sequence of states satisfies the formula ψ is $\blacktriangleleft\ \alpha$. The formula $\mathcal{K}_i^A \psi$ expresses "coalition A *knows* that ψ is true (with information degree i)." We use standard abbreviations like $\varphi \vee \psi = \neg(\neg\varphi \wedge \neg\psi)$, $\Diamond\varphi = \mathsf{true}\,\mathsf{U}\varphi$, and $\Box\varphi = \neg\Diamond\neg\varphi$.

3.2 Strategies and Semantics

An *a-strategy* for a player a is a function s assigning a move from $\Delta(q, a)$ to each state q. It is *i-uniform*, if $q_1 \sim_{\mathsf{eq}_i(a)} q_2$ implies $s(q_1) = s(q_2)$: In states that a player cannot tell apart with information degree i, he performs the same move. For a coalition A, an *A-strategy* is a family $(s_a)_{a \in A}$, where each s_a is an a-strategy, it is i-uniform if every s_a is. We only consider *memoryless* strategies, since each state contains complete information about the preceding protocol run. Formulas are evaluated on states or on paths, where a *path* is a sequence λ of states in a CGS $\mathcal{C}$. With $\lambda[i]$ we denote the ith state in λ.

Definition. Let $\mathcal{C} = (\Sigma, Q, \mathbb{P}, \pi, \Delta, \delta, \mathsf{eq})$ be a CGS, let φ be a state formula, let ψ_1 and ψ_2 be path formulas, let $\mathsf{S}_1, \ldots, \mathsf{S}_n$ be strategies instantiating the variables $\mathrm{S}_1, \ldots, \mathrm{S}_n$, let λ be a path, let $t \in \mathbb{N}$, let $q \in Q$ be a state, let $\overrightarrow{\mathsf{S}}$ be an abbreviation for $(\mathsf{S}_1, \ldots, \mathsf{S}_n)$. Then

- $\mathcal{C}, \overrightarrow{\mathsf{S}}, q \models p$ iff $q \in \pi(p)$ for $p \in \mathbb{P}$,
- negation and conjunction are treated as usual,
- $(\lambda, t), \overrightarrow{\mathsf{S}} \models \varphi$ iff $\mathcal{C}, \overrightarrow{\mathsf{S}}, \lambda[t] \models \varphi$,
- $(\lambda, t), \overrightarrow{\mathsf{S}} \models \mathsf{X}\psi_1$ iff $(\lambda, t+1), \overrightarrow{\mathsf{S}} \models \psi_1$,
- $(\lambda, t), \overrightarrow{\mathsf{S}} \models \mathsf{P}\psi_1$ iff $(\lambda, t'), \overrightarrow{\mathsf{S}} \models \psi_1$, for some $t' \leq t$,
- $(\lambda, t), \overrightarrow{\mathsf{S}} \models \mathsf{X}^{-1}\psi_1$ iff $t \geq 1$ and $(\lambda, t-1), \overrightarrow{\mathsf{S}} \models \psi_1$,

– $(\lambda, t), \overrightarrow{\mathsf{S}} \models \psi_1 \mathsf{U} \psi_2$ iff there is some $i \geq t$ such that $(\lambda, i), \overrightarrow{\mathsf{S}} \models \psi_2$ and $(\lambda, j), \overrightarrow{\mathsf{S}} \models \psi_1$ for all $t \leq j < i$,
– $\mathcal{C}, \overrightarrow{\mathsf{S}}, q \models \mathcal{K}_i^A \varphi_1$ iff $\mathcal{C}, \overrightarrow{\mathsf{S}}, q' \models \varphi_1$ for all $q' \in Q$ with $q' \sim_{\mathsf{eq}_i(A)} q$,
– $\mathcal{C}, \overrightarrow{\mathsf{S}}, q \models \langle\langle A_{i_1} : \mathrm{S}_{i_1}, \ldots, A_{i_k} : \mathrm{S}_{i_k} \rangle\rangle^{\blacktriangleleft \alpha} \psi$ iff when coalition A_{i_j} plays[3] the A_{i_j}-strategy S_{i_j} for all j, then the resulting path satisfies ψ with probability $\blacktriangleleft \alpha$, for every possible behavior of the players in $\Sigma \setminus (A_{i_1} \cup \cdots \cup A_{i_k})$.

This definition treats formulas where strategies instantiating the variables S_i are given. A *quantified strategy formula* is a state formula prefixed by a quantifier block where each strategy variable S_i is quantified with $\exists_i$ or $\forall_i$ for an information degree i. This expresses "there is (for all) i-uniform strategies," with the obvious semantics: $\exists_{i_1} \mathrm{S}_1 \forall_{i_2} \mathrm{S}_2 \ldots \exists_{i_n} \mathrm{S}_n \varphi$ is true in state q if there is a i_1-uniform strategy S_1 such that for all i_2-uniform strategies S_2, ..., there is an i_n-uniform strategy S_n such that this choice of strategies satisfies φ according to the definition above.

3.3 Modeling of Knowledge

The knowledge operator used in QAPI (see above) has the usual semantics from epistemic logics. For security settings, this is often unsuitable: If a party "knows" a fact to be true with probability significantly larger than $\frac{1}{2}$, is often enough for a protocol to be insecure. This, however, is not captured in the standard definition. Also, a party's knowledge may sometimes take other principals' strategies into account, which also cannot be expressed with the standard epistemic knowledge operator. However, QAPI's quantified strategies can be used to address these issues. As an example, "there is a strategy s_A such that B knows (with information degree i) whether φ holds with probability at least $\frac{4}{5}$, if B knows that A follows s_A," can be expressed as follows: We modify the protocol for B to allow an "announcement" proclaiming that B believes φ to be true.[4] Let bel_φ be a formula true in all states in which B has made this announcement (see also [HT93]). Then the above can be expressed as

$$\exists \mathrm{S}_A \exists_i \mathrm{S}_B \, \langle\langle A : \mathrm{S}_A \rangle\rangle^{\geq 1} \, \langle\langle B : \mathrm{S}_B \rangle\rangle^{\geq \frac{4}{5}} \, (\mathsf{bel}_\varphi \iff \varphi) \,.$$

Our discussion of coercion-resistance (Section 5.4) contains an example of a security analysis where such considerations are relevant. The above discussion shows that explicit *uniform* strategies are strong enough to express the knowledge operator, although at the cost of modifying the game structure (in our case, the protocol). Hence the basic knowledge operator can be seen as "syntactic sugar," which we however keep in the language as it can increase readability. We are grateful to anonymous reviewers pointing out these issues.

[3] If a appears in more than one A_{i_j}, he follows strategy S_{i_j} with $j = \min \{ j \mid a \in A_{i_j} \}$.
[4] This can be done by e.g., introducing a dedicated party who receives messages saying "I believe φ is true/false," or with several other mechanisms

4 Main Result

Security of protocols in our model is decidable for convergent subterm theories:

Theorem 4.1. *Assume that* E *is a convergent subterm theory. There is an algorithm which, given a protocol Pr, a set C of corrupted identities, and a quantified strategy formula φ, decides whether $\mathcal{C}_{Pr}, q_{init}^{C} \models \varphi$.*

The challenge in the proof is that active adversaries can send arbitrarily complex messages, leading to an infinite structure $\mathcal{C}_{Pr}$. We show that it suffices to consider "bounded strategies:" Protocols only parse terms up to a bounded depth; rewriting rules resulting from convergent subterm theories also have "bounded" effects. It follows that one can restrict the adversary to send terms of bounded depth. This [RT03]-style argument only directly covers reachability properties; more involved arguments apply to strategic and epistemic properties.

5 Applications

We now show several examples of application of our result. In addition to our running coin-flipping example, we also treat abuse-freeness of contract signing protocols. We briefly mention that standard anonymous broadcast protocols as the dining cryptographers can be expressed in our model in the straight-forward way. We also treat two applications that use our framework in a less obvious way, namely 1. accountability and verifiability, and 2. coercion-resistance of voting protocols. An in-depth discussion of these properties is out of the scope of this paper, we treat these properties in as much detail as required to highlight the features of our approach. In particular, our treatment uses direct reasoning about strategies, epistemic and probabilistic aspects in an essential way.

5.1 The Coin-Flipping Protocol

The coin-flipping protocol satisfies its previously-mentioned security property:

Proposition 5.1. *The state $q_{init}^{\{\mathsf{Alice}\}}$ of the CGS induced by the coin-flipping protocol satisfies the formula* $\forall_3 \mathrm{S} \neg \langle\langle \mathcal{A} : \mathrm{S} \rangle\rangle^{>0.5} \Diamond \left(\mathsf{fin}_{00}^{B} \vee \mathsf{fin}_{11}^{B} \right)$.

The formula is satisfied because the messages $hash(\langle 0, N \rangle)$ and $hash(\langle 1, N \rangle)$ are indistinguishable for Alice, since she does not know N. Therefore, a 3-uniform strategy has to choose the same action for both of Bob's possible messages.

5.2 Abuse-Freeness of Contract Signing Protocols

If Alice and Bob want to exchange a contract, abuse-freeness requires that there is no situation where Bob can prove to an outsider Charly that the current state is *unbalanced*, i.e., Bob can unilaterally decide whether the contracts are

successfully exchanged or not. A straight-forward definition of abuse-freeness is "there is no point in the execution of the protocol where Bob has a strategy to ensure that Charly knows that the protocol is in an unbalanced state." This can be expressed in our model in the obvious way. We now show how the more complex definition of abuse-freeness given in [KKW06] can be expressed in our framework. As a consequence, we obtain decidability of abuse-freeness.

The definition from [KKW06] can be strengthened in various ways (see, e.g., [KST10]), we use their definition as decidability was mentioned as an open question in [KKW06]. Let φ_1 and φ_2 be formulas describing protocol outcomes (e.g., "Bob obtains a contract"). A state q of a protocol run is (φ_1, φ_2)-*unbalanced*, if Bob has a strategy ensuring that the remaining run satisfies φ_1, and a strategy to ensure φ_2. Informally, a protocol is (φ_1, φ_2)-*abusive*, if there is a situation where Bob can prove to an outsider Charly that the current state is (φ_1, φ_2)-unbalanced. We often write "abusive" and "unbalanced" for (φ_1, φ_2)-abusive and (φ_1, φ_2)-unbalanced, since φ_1 and φ_2 will always be clear from the context.

In the [KKW06] definition, Charly receives a single message m from Bob, and plays no active part. Based on m, Charly decides whether to believe Bob. Since Bob can always delay sending a proof, Bob can only convince Charly that the current state is or an earlier state was unbalanced. For the epistemic aspects, the definition from [KKW06] uses tests (see Section 1) that Charly performs on proofs presented by Bob. A test θ is *convincing*, if for every state q where Bob can produce a message m satisfying θ, there is an ancestor state q' of q such that 1. q' is (φ_1, φ_2)-unbalanced, 2. in q', Bob can produce a message satisfying θ.

The second condition precludes protocols in which the adversary can produce proofs of unbalance of a previous state only after the protocol run is over. In particular, a proof can always be generated in the state that actually *is* unbalanced. A protocol is *abusive* if there is a convincing test θ and a reachable state where the adversary can produce a message satisfying θ, such a state is called θ-*possible*. This is a state in which the adversary can convince Charly. A protocol is abuse-free if it is not abusive.

Roughly, a protocol is abuse-free if there is no state where Charly *knows* that Bob has strategies satisfying the above. Since our model is concurrent, Charly obtains, in addition to the message from Bob, timing information about the protocol run. If Charly is an "outsider," this information should be unavailable to him. Therefore, we must ensure that Charly does not take this additional timing-information into account when deciding on whether to accept Bob's proof.

Since accepting/rejecting a proof is Charly's choice, the set of accepted proofs is a strategy for Charly, which must be constant throughout the protocol run: Charly may not change his strategy, even if other principals do. This cannot be expressed in ATL*, but is easy to express in QAPI, since QAPI can directly assign strategies to players. Therefore, we can express the "consistency" of Charly and similar aspects.

We assume that Alice is honest (as are other parties like a trusted third party, buffers, etc.), and Bob is the adversary. Since the definition from [KKW06] uses a deterministic model, we assume that the protocol does not use probabilism.

Theorem 5.2. *There is an algorithm which, when given a protocol Pr and path formulas φ_1 and φ_2, produces a protocol Pr' and a formula ψ^{abuse} such that Pr is (φ_1, φ_2)-abusive if and only if $\mathcal{C}_{Pr'}, q_{init}^{\{B\}} \models \psi^{\text{abuse}}$.*

We first give the construction of Pr', and then prove that it indeed satisfies the required properties. Pr' is obtained from Pr as follows:

- we introduce a principal C (Charly) who receives a message from Bob at some point during the protocol run. In the second-to-last step of the protocol, Charly forwards the first message received from the adversary to the verifier (see below), and ignores all other messages. In the final step, Charly chooses to move into either an "accept" or a "reject" state.
- we introduce a principal, V (Verifier), who ensures that Charly only bases his decision whether to accept Bob's proof on the actual proof. To this end, V receives a message from Charly at the last step of the protocol, and has access to the same private keys as Charly. V does not send any messages.

Charly makes his decision *after* the verifier receives Charly's message. This allows us to easily express the verifier's knowledge about Charly's decision.

The idea of the construction is the following: At some point during the protocol run, the adversary sends a message to Charly, which provides Charly with timing information that he should not have. We ensure that Charly's decision is independent of the time when the message is received. This is realized by requiring that V *knows* Charly's decision, and the only information that V has about Charly's state is the message which Charly received from Bob.

We now construct the formula expressing abusiveness.

- let φ^{unbal} be the formula $\left\langle\left\langle \mathcal{A} : \mathrm{S}_1^{\mathcal{A}} \right\rangle\right\rangle \varphi_1 \wedge \left\langle\left\langle \mathcal{A} : \mathrm{S}_2^{\mathcal{A}} \right\rangle\right\rangle \varphi_2$ which expresses that the current state is unbalanced, here $\mathrm{S}_1^{\mathcal{A}}$ and $\mathrm{S}_2^{\mathcal{A}}$ will be existentially quantified in the quantifier block preceding the entire formula,
- let φ^{ver} be the formula $\mathcal{K}_3^{\mathsf{V}} \langle\langle \mathsf{C} : \mathrm{S}_{\mathsf{C}} \rangle\rangle \mathsf{X}\mathsf{acc} \vee \mathcal{K}_3^{\mathsf{V}} \langle\langle \mathsf{C} : \mathrm{S}_{\mathsf{C}} \rangle\rangle \mathsf{X}\mathsf{rej}$, where acc and rej are true if Charly is in a local state where he accepts, respectively where he rejects. φ^{ver} expresses that the verifier knows whether Charly will accept or reject Bob's proof, given that Charly plays the strategy S_{C}.
- let rec be a variable true in the states where Charly has previously received a message from the adversary, and let $\varphi^{\mathcal{A}-\text{greedy}}$ be the formula $\neg \mathsf{P}\left(\left(\langle\langle \mathcal{A} : \mathrm{S}_{\mathcal{A}}, \mathsf{C} : \mathrm{S}_{\mathsf{C}} \rangle\rangle \left((\mathsf{X}\mathsf{rec}) \wedge (\Diamond \mathsf{acc})\right)\right) \wedge \neg \mathsf{X}\mathsf{rec}\right)$. This expresses that (if strategy $\mathrm{S}_{\mathcal{A}}$ is all-quantified over complete-information strategies) the strategy played by the adversary up to now in the protocol run is "greedy:" If at some point in the protocol run the adversary could send an "accepted" proof to Charly (according to Charly's strategy S_{C}), then the adversary in fact did send a message to Charly in the next step (or previously).

Let end be true at the end of the original protocol run. The formula ψ^{abuse} expressing abusiveness is the following:

$$\begin{aligned}\exists_2 \mathrm{S}_1^{\mathcal{A}} \exists_2 \mathrm{S}_2^{\mathcal{A}} \exists_3 \mathrm{S}_{\mathsf{C}} \exists_1 \mathrm{S}_{\Sigma} \forall_3 \mathrm{S}_{\mathcal{A}} \quad & \langle\langle \mathsf{C} : \mathrm{S}_{\mathsf{C}} \rangle\rangle \, (\Box(\mathsf{end} \to \varphi^{\text{ver}}) \\ & \wedge \Box(\mathsf{acc} \to (\varphi^{\mathcal{A}-\text{greedy}} \to (\mathsf{P}_{\Box}(\mathsf{rec}' \to \mathsf{X}^{-1}\varphi^{\text{unbal}})))) \\ & \wedge \ \langle\langle \mathsf{C} : \mathrm{S}_{\mathsf{C}}, \Sigma : \mathrm{S}_{\Sigma} \rangle\rangle \Diamond \mathsf{acc})\end{aligned}$$

Here $\mathsf{P}_{\Box}\varphi$ abbreviates $\neg\mathsf{P}\neg\varphi$, ("$\varphi$ is true in the past") and with slight abuse of notation Σ denotes the principals in the original protocol (everyone except Charly and the verifier), and rec' is true in the states where Charly just received the first non-empty message from the adversary. [5] Quantifying S_{Σ} over complete-information strategies quantifies over all reachable states. This expresses

1. After V received the forwarded message from C, V knows whether Charly accepts the proof. This ensures that Charly bases his decision only on the information chat Charly can obtain from the first message received from the adversary, and thus Charly's decision is determined by a test.
2. whenever Charly moves into an accepting state, the state of the protocol run in fact was unbalanced in the state where the adversary sent the proof to Charly—*if* the adversary played a "greedy" strategy (see above). This ensures that Charly only accepts proofs that could be generated in an unbalanced state, and implies that minimal θ-possible states are exactly those in which Bob sends a convincing proof to Charly.
3. there is a reachable state in which Charly's strategy accepts.

We now show that the above construction is indeed correct, i.e., we give the proof of Theorem 5.2:

Proof. First assume that the protocol is not abuse-free. Let θ be a corresponding test, and let q be a θ-possible and unbalanced state, without loss of generality assume that no proper ancestor of q is θ-possible (by definition of abusiveness in the sense of [KKW06], the first θ-possible state in a protocol run must be unbalanced). We define the strategies for Σ and Charly, instantiating S_{Σ} and S_{C}, as follows:

- The adversary and the honest principals of the original protocol run perform all necessary actions to reach the state q. Note that this is a state of the original protocol, hence they do not need Charly's help to achieve this.[6] After this, the adversary sends a message satisfying the test θ to Charly.
- Charly's only decision is whether to accept or reject at the end of the protocol run (the forwarding of the message to the verifier is hard-coded into Charly's definition), he moves into the accepting state if the first message he received from the adversary satisfies θ and in the rejecting state otherwise.

[5] The quantification for $\mathrm{S}_1^{\mathcal{A}}$ and $\mathrm{S}_2^{\mathcal{A}}$ only requires strategies to be uniform for information degree 2 to remain compatible with [KKW06].

[6] formally, they reach a state q' of the new protocol which corresponds to q in a natural way, it is straight-forward to define this relationship—recall that the additional principals have no influence on the behavior of the principals present in the original protocol.

– The strategies instantiating $S_1^{\mathcal{A}}$ and $S_1^{\mathcal{A}}$ perform appropriate actions to reach φ_1 and φ_2 whenever possible, i.e., contain hard-coded strategies to reach φ_1 and φ_2 from every state where this is possible.[7]

We claim that this choice of strategies satisfies the formula.

1. the first conjunct is satisfied because by definition, the question whether Charly moves into an accepting or a rejecting state at the end of the protocol run only depends on the message he received from the adversary, which is a message that, by construction, the verifier has access to, and by construction, Charly and the verifier have access to the same secret keys, hence they derive the same knowledge from the message (recall that Charly's nonces do not appear in the protocol run, since Charly does not construct messages on his own, but only forwards a message received from the adversary).
2. the second conjunct requires that if Charly accepts at the end of the protocol run, and the strategy played by the adversary is greedy, then the state directly before Charly received the first message from the adversary was unbalanced. This is satisfied because the state q is unbalanced.
3. the final conjunct is satisfied since by construction, the adversary sends a message satisfying the test θ, and thus by definition of Charly's strategy, he moves into an accepting state.

Hence if the protocol Pr is abusive, then the formula is indeed satisfied in the initial state of the protocol Pr' in which Bob is corrupted.

For the converse, suppose that the formula is satisfied in the initial state of the protocol. Note that we can, without loss of generality, assume that the adversary does not send any messages to the verifier. We construct a test θ satisfying the requirements. The strategy used to instantiate $\mathsf{S_C}$ has, by the first conjunct of the formula, the property that the choice depends only on the verifier's knowledge, which (by definition of knowledge in our model) means that Charly's decision only depends on the outcome of a test which the verifier can perform on the message received by Charly, which is (by construction of Charly) the first message that Charly receives from the adversary. Let θ denote this test.

Obviously, there is a θ-possible state, since a state in which the verifier accepts is reachable in the protocol due to the final conjunct of the formula.

Hence it remains to show that every reachable θ-possible state is the (not necessarily direct) successor of an unbalanced state. Thus let q be a θ-possible, reachable state, without loss of generality assume that no proper predecessor of q is θ-possible. Now consider a strategy for all principals in Σ that first reaches the state q without the adversary sending any messages to Charly, and then letting the adversary deliver a message satisfying θ to Charly. Since the message satisfies θ, Charly will move to the accepting state at the end of the protocol run. By construction, since no predecessor of q is θ-possible, the adversary's strategy is

[7] See below for a note on the uniformity issues appearing here; in the current situation, our definition of these strategies implies that information degree 2 is sufficient to identify the strategies to apply here.

greedy, i.e., the protocol run satisfies $\varphi^{\mathcal{A}-\text{greedy}}$. Therefore, the second conjunct requires that the state directly preceding the one in which Charly receives the first message from the adversary. i.e., the state q, is unbalanced. This concludes the proof.

From the above and our main result, Theorem 4.1, we obtain the following corollary:

Corollary 5.3. *Abuse freeness as defined in [KKW06] is decidable.*

On variations of abuse-freeness We note that different notions of abuse-freeness can also be captured in our model. As mentioned above, the definition of abuse-freeness in [KKW06] grants the adversary additional knowledge to identify a strategy, in this section we show that this notion of abuse-freeness can be defined in our model as well (and thus is decidable). We also comment on natural variations of the definition of abuse-freeness.

As mentioned before there is a subtle point when dealing with incomplete information strategies, which is the difference between requiring a strategy to *exist*, or to be *known*. As an illustration, consider the following (contrived) example: Assume we have a cryptographic protocol where two outcomes, described by the formulas φ_1 and φ_2, are of interest. Assume that there is a single honest principal, Alice, and her first move is to choose a successor state out of q_1 and q_2, these states are indistinguishable for the adversary. In both cases, the message `unbalanced` is sent to the adversary. Now, Alice awaits a message consisting of a single bit, and behaves as follows:

- In state q_1, if the bit is 0, she proceeds in a fashion satisfying φ_1, if the bit if 1, she chooses actions satisfying φ_2.
- In the state q_2, she behaves exactly the opposite way, i.e., when receiving the bit 1 the satisfies φ_1, and on bit 0, she ensures φ_2.

(Of course here we assume that it is in Alice's power alone to ensure that φ_1 or φ_2 are satisfied, for example these could be formulas talking only about the internal state of Alice.)

Consider the state q_1. There *is* a strategy for the adversary to ensure φ_1 (namely, send bit 0), and a strategy to ensure φ_2 (send bit 1 instead). These strategies are constant, therefore in particular, both of them are uniform (or *view*-strategies in the terminology of [KKW06]). However, since the adversary does not have a way of knowing whether the current state is q_1 or q_2, the mere existence of such a strategy does not enable the adversary to actually control the outcome, since he cannot *identify* the correct strategy. This distinction is sometimes regarded as the difference between knowledge *de dicto* and knowledge *de re* (see, e.g., [JÅ06]). This topic is not addressed in [KKW06], our formulation of abuse-freeness does not allow (in the above example) the adversary to choose different strategies in the states q_1 and q_2. However, he is allowed to decide to always act as if the state is q_1 and then also is regarded as successful in that state,

however he is then unsuccessful in q_2—this comes with the additional price that to achieve abusiveness of a protocol, the adversary needs to be able to convince Charly that he is in state q_1, not in state q_2, which is only possible if Charly has some knowledge the adversary does not have. Hence for a passive Charly, the above situation will be regarded as not abusive in our model, since the adversary does not have a way to exploit the theoretically available strategies.

However, one might want to give the adversary these additional capabilities, essentially only demanding that strategies can be *implemented*, but not necessarily *identified* with the adversary's knowledge—formally this corresponds to existentially quantify the strategies in nested subformulas of a formula, such that existential quantifiers may return a different strategy in each state. This can easily be achieved by using strategy choices instead of strategies which can be used to allow strategies to depend on the state[8], details of this can be found in [Sch10a]—we did not introduce strategy choices in our introduction of QAPI as the simpler situation where we just consider strategies is sufficient to express most properties. Using strategy choices in this way essentially simulates the above-mentioned quantification of strategies in the nested subformulas. However note that, as mentioned in [Sch10a], it seems very unnatural to allow players to use knowledge which is not available to them for the identification, but not the implementation of a strategy (in particular, due to these arguments, [Sch10a] does not introduce notation for the particular form of strategy choices used above, but explicitly states that the decidability result remains true for this generalization).

Similar issues also have been discussed in various papers on epistemic strategic logics, see for example [JvdH04].

Finally, if Charly has inside information about the protocol run (including the number of steps that have been performed), then the situation is much simpler, in particular there is no need to introduce a verifier as done above. In this case, abusiveness can be characterized with a Charly similarly as in the previous section and the formula $\exists_1 \mathrm{S}_\Sigma \, \langle\langle \Sigma : \mathrm{S}_\Sigma \rangle\rangle^{\geq 1} \, \mathcal{K}_3^{\mathsf{C}} \mathsf{X}^{-1} \varphi^{\mathrm{unbal}}$, which (with additional quantification for the adversarial strategies mentioned in φ^{unbal}) simply states that there is a reachable protocol state in which Charly knows that the previous state was unbalanced (he cannot know that the *current* state is unbalanced because our model is concurrent and Charly does not know actions occurring at the same time as the adversary presenting his proof to Charly).

As a consequence of our main result Theorem 4.1, the variations of abuse-freeness discussed above remain decidable.

[8] To cover the version of abuse-freeness discussed here, one would consider strategy choices for the adversary where the *choice* of strategy can be performed with full information, i.e., information degree 1, while the strategies themselves have to be uniform for information degree 2 or 3—the definition in [KKW06] uses what is information degree 2 in our terminology, which is appropriate there since there are no relevant strategic decisions by the adversary that depend on the content of ciphertexts that the adversary cannot decrypt in their situations. In a more general setting, limiting the cryptographic abilities of the adversary, and thus requiring information degree 3 may be more appropriate.

5.3 Accountability and Verifiability

Accountability and verifiability are properties relevant for protocols involving trusted parties, e.g., voting [KRS10], auctions [PRST08], contract signing [ASW98], identity-based encryption etc. In [KTV10a], a formal definition of accountability is given that is independent of the specific application.

Accountability requires that if a protocol run "fails" (i.e., does not achieve some goal), then a party J (the "judge") can determine which one of the participants in the protocol "misbehaved," i.e., did not follow the protocol.

Up to now, we modeled principals either as honest, or as part of the adversary. Accountability is concerned with principals who have a "wanted" behavior (the protocol), but can start "misbehaving" during the protocol run (i.e., abandon the protocol and behaving adversary-like from that point on).

To express this we use our model in a different way: We modify every honest principal of the protocol except J to run an "adversary program" at any time. This is a new sub-branch of the protocol, and forwards received messages to the adversary, lets the adversary dictate messages to be sent to the other principals, and provides an oracle for operations involving the private key of the "misbehaving" identity, e.g., decrypts ciphertexts and signs messages as instructed by the adversary (the exact set of services provided depends on the involved cryptographic primitives).[9] Since the variables in $\mathcal{C}_{Pr}$ indicate the current state of honest principals, for each i we have a formula φ_i^{adv} that is true iff i runs the adversary program. Forwarding and oracle access causes delay in the protocol execution, to account for this we introduce "wait cycles" into the protocol. The adversary program essentially models dynamic corruption.

In [KTV10a], *individual accountability* is defined as follows: At the end of every protocol run in which a goal φ is not satisfied, J announces the identity of some party that did not follow the protocol (using a distinguished state for each output). Let blame_i be a formula that is true if J announced that i "misbehaved." Let φ be a goal. Then a protocol provides individual accountability for φ if the following formula is satisfied ($\forall \mathrm{S}_\emptyset \langle\langle \emptyset : \mathrm{S}_\emptyset \rangle\rangle$ quantifies over all reachable states):

$$\forall \mathrm{S}_\emptyset \langle\langle \emptyset : \mathrm{S}_\emptyset \rangle\rangle \Box(\neg\varphi \rightarrow \Diamond(\bigvee_{i \in I} (\mathrm{blame}_i \wedge \varphi_i^{\mathsf{adv}})).$$

This expresses that if φ is not satisfied, then at the end of the run J will correctly announce one identity from I that did not follow the protocol.

The above does not use epistemic or strategic properties: We merely expressed that J works "correctly." Epistemic features come into play when the situation is less clear than above, i.e., when there is no existing judge procedure that we can use. We can ask whether a party J has enough information[10] to

[9] Usually, the adversary only accesses the oracle a finite number of times: Decryptions and signatures are only necessary for encryptions done by, or signature verifications performed by, honest principals; these only perform a finite number of operations. Hence the "oracle" can be implemented in a finite protocol role.

[10] Clearly, the protocol must specify which information J has, i.e., which messages J receives—if J has complete information, accountability trivial.

serve as a judge, and derive an implementation. The following expresses that if φ is false, then J will know, for some party i, that i did not follow the protocol:

$$\forall \mathrm{S}_\emptyset \, \langle\langle \emptyset : \mathrm{S}_\emptyset \rangle\rangle \, \Box(\neg\varphi \to \Diamond(\bigvee_{i \in I} (\mathcal{K}_3^{\mathsf{J}} \varphi_i^{\mathsf{adv}})).$$

If the formula is true, J has enough knowledge to serve as judge (the index 3 states that J's knowledge is limited by cryptography). We obtain an "implementation" of J in a straight-forward way: We allow J (in addition to other instructions that J follows in the original protocol) to perform "blame" announcements as earlier. We now ask whether there is a strategy for J to "blame correctly:"

$$\exists_3 \mathrm{S}_\mathsf{J} \, \langle\langle \mathsf{J} : \mathrm{S}_\mathsf{J} \rangle\rangle \, (\Box(\neg\varphi \to \Diamond(\bigvee_{i \in I} \mathrm{blame}_i)) \wedge \bigwedge_{i \in I} (\mathrm{blame}_i \to \varphi_i^{\mathsf{adv}})),$$

in the positive case the strategy for J then encodes a verification program. Finally, verifiability can be seen as a weaker form of accountability. In [KTV10a], it is defined as follows: A goal φ of a protocol Pr is verifiable by J if J knows whether φ holds when the protocol run is over. This can be easily expressed in our model: Let end be a propositional variable that is true at the end of the protocol run. Then the formula

$$\forall \mathrm{S}_\emptyset \, \langle\langle \emptyset : \mathrm{S}_\emptyset \rangle\rangle \, \Box(\mathsf{end} \to (\mathcal{K}_3^J \varphi \vee \mathcal{K}_3^J \neg\varphi))$$

expresses that J knows whether φ holds at the end of every possible protocol run. Theorem 4.1 now implies decidability of accountability and verifiability.

5.4 Coercion-Resistance of Voting Protocols

Coercion-resistance requires that no voter Alice can prove to a party Charly that she voted as instructed by him, precluding selling of votes. In [KTV10b], coercion-resistance was defined[11] as follows: For every "coercer strategy" of Charly, there is a "counter-strategy" for Alice such that Alice's vote is counted as she wants to vote, but Charly believes that he controlled her voting process.

Clearly, we cannot require Charly to *always* fail to "catch" Alice—if Charly's chosen candidate receives zero votes, then Charly knows that Alice did not obey him. We thus allow Charly to correctly guess that Alice voted differently than promised with some probability, possibly larger than $\frac{1}{2}$. See [KTV10b] for a discussion of suitable values for the involved probabilities. We model Charly's belief as the probability to successfully "guess" whether Alice followed his instructions. This mirrors the approach of [HT93] to consider probabilistic knowledge as strategies for a betting game, see also Section 3.3.

[11] Their definition is given in a cryptographic model, we present an analogous formulation in our symbolic model. Other definitions [KT09] are expressed in epistemic terms close to our model. However the game-based definition from [KTV10b] covers probabilistic aspects that we want to model.

We express coercion-resistance in our model. We note that our model requires that the number of communication rounds between Alice and Charly is bounded by a constant, since this has to be encoded into Alice's protocol description. A generalized model with no bounds on the protocol length can be defined, however, such a model will be undecidable (cp. [KKW09]). We stress that neither the complexity nor the structure of the messages are restricted in our model.

In coercion-resistance, two principals may deviate from the protocol: Charly uses a *coercer strategy* to influence Alice, and Alice runs a *counter-strategy* to vote as she intends[12]. Our model allows arbitrary behavior only for the adversary, hence we model *both* the coercer and the counter strategy as adversary-strategies. We introduce a test principal T whose goal it is to determine whether Alice follows Charly's instructions (the adversary plays the "coercer strategy") or uses the "counter-strategy." Since both of these strategies are played by the adversary, we need a way to distinguish them. To this end, the strategies have to "announce themselves:" We let Alice expect, in the first message from the adversary, a bit signaling the performed strategy, she changes local state accordingly. She runs a copy of the adversary program (see Section 5.3) from then on. We use formulas $\varphi^{\mathsf{A-coerc}}$ and $\varphi^{\mathsf{A-counter}}$ to express that the running strategy signaled coercion or counter, respectively. A T-strategy is *successful* if T announces "coercion" iff the running strategy signaled coercion, and "counter" iff the strategy signals "counter." Since T's epistemic capabilities should match Charly's, T has access to the same messages that Charly would see in a protocol run.

To express that the counter-strategy lets Alice vote as she wants, we introduce a principal V (vote) choosing Alice's (sincere) vote, which he sends to Alice. V's strategies then correspond to Alice's possible votes. Coercion-resistance for a probability δ is now (semi-formally) expressed as follows[13]:

> `for all` $\mathcal{A}$-strategies s_{coerc} signaling coerce
> `there is` an $\mathcal{A}$-strategy s_{counter} signaling counter s.t.
> s_{counter} lets Alice vote as chosen by V
> `AND` no T-strategy is successful with probability $\geq \delta$.

This expresses that for every coercer strategy, there is a counter-strategy letting Alice vote as she wants, and the test principal (with information as available to Charly) cannot identify the performed strategy with probability $\geq \delta$.

To express this in QAPI, let φ^{V} express that Alice voted as instructed by V (this formula depends on the voting system), let $\varphi^{\mathsf{A\text{-}coerc}}$ and $\varphi^{\mathsf{A\text{-}counter}}$ express that coercion (counter) is signaled. Let $\varphi^{\mathsf{T\text{-}suc}}$ indicate that T guesses correctly.

$$\varphi^{\mathsf{T}<\delta} = \neg\left(\left(\langle\langle \mathsf{T}:\mathrm{S}_{\mathsf{T}}, \mathcal{A}:\mathrm{S}_{\mathsf{counter}}\rangle\rangle^{\geq\delta}\,\varphi^{\mathsf{T\text{-}suc}}\right) \wedge \left(\langle\langle \mathsf{T}:\mathrm{S}_{\mathsf{T}}, \mathcal{A}:\mathrm{S}_{\mathsf{coerce}}\rangle\rangle^{\geq\delta}\,\varphi^{\mathsf{T\text{-}suc}}\right)\right)$$

expresses that T's success probability is less that δ for one of the strategies.

[12] Clearly, in many protocols there will be a fixed counter-strategy that Alice can use which we could directly "implement" into our modeling of Alice; this would simplify the modeling of coercion-resistance significantly.

[13] For readability, we omit the universal quantification over V's strategy

$$\varphi^{\mathsf{sig\text{-}coerce}} = (\langle\langle \mathcal{A} : \mathrm{S}_{\mathsf{coerc}} \rangle\rangle^{\geq 1} \Diamond \varphi^{\mathsf{A\text{-}coerc}})$$

expresses that $\mathrm{S}_{\mathsf{coerc}}$ signals coercing correctly, analogously let $\varphi^{\mathsf{sig\text{-}counter}}$ express that $\mathrm{S}_{\mathsf{counter}}$ signals counter. Finally,

$$\varphi^{\mathsf{vote}} = \langle\langle \mathcal{A} : \mathrm{S}_{\mathsf{counter}} \rangle\rangle^{\geq 1} \Diamond \varphi^{\mathsf{V}}$$

expresses that the strategy $\mathrm{S}_{\mathsf{counter}}$ lets Alice vote as she wants to. We now express coercion-resistance as follows:

$$\forall_3 \mathrm{S}_{\mathsf{coercer}} \exists_3 \mathrm{S}_{\mathsf{counter}} \forall_3 \mathrm{S}_{\mathsf{V}} \forall_3 \mathrm{S}_{\mathsf{T}} \varphi^{\mathsf{sig\text{-}coerce}} \rightarrow (\varphi^{\mathsf{sig\text{-}counter}} \wedge \varphi^{\mathsf{vote}} \wedge \varphi^{\mathsf{T}<\delta}).$$

We stress that the coercer- and counter-strategies are played by the adversary $\mathcal{A}$ and not by Alice.

We point out some subtleties of the above modeling. Note that in order to ensure that the test principal T has exactly the same information about a protocol run that Charly would have, Alice forwards messages to T as instructed by the adversary. This happens "automatically," as Alice runs the adversary program, and hence the adversary (i.e., the coercer- and counter-strategies) completely control the communication between Alice and T. Hence the coercer strategy will attempt to include "proofs" into this strategy showing that Alice indeed followed the instructions. In particular, the coercer strategy will inform T about the complete communication between Alice and the adversary. This communication will contain any kind of "receipt" that Alice might receive from the voting system, if such a receipt is issued. The counter-strategy has to simulate this exchange—including any possible "receipts"—and at the same time make sure that the actual communication between Alice and the voting system ensures that Alice's vote is counted as instructed by V.

Also note that it is not even necessary to only consider coercer-strategies that signal "coerce" correctly—since such a strategy, as argued above, will attempt to make itself distinguishable from a counter-strategy, it is in its best interest to signal correctly. However, we included the requirement in the formula above for ease of presentation.

Note also that if Alice runs the adversary program, then the adversary knows which vote V wants Alice to cast—the communication between Alice and V is visible to the adversary. This allows both the coercer strategy to deliberately vote differently than Alice wants to, and allows the counter-strategy to depend on Alice's vote (which it has to in order to ensure that the vote is counted as intended, since the counter-strategy completely controls Alice's communication with the voting system).

We stress again that *both* strategies, that of "Charly" and that of "Alice," are in fact "played" by the adversary, who in both cases completely takes over Alice's communication. It is the task of the verifier to determine—with the knowledge that Charly would have—whether the strategy that Alice "effectively plays" is one that lets Charly dictate Alice's actions, or one that achieves Alice's own goal.

Several key features of our approach are used in the above modeling: It is clearly necessary to consider only uniform strategies. We also made extensive use of quantification: Letting the strategy of T depend on the $\mathcal{A}$-strategies is crucial for the approach, as is the ability to directly talk about specific strategies in formulas. Finally, reasoning about success probabilities of strategies was required to express the probabilistic notion of coercion-resistance.

Variations of coercion-resistance can be expressed similarly: One can exchange the order of quantification of the counter-strategy and the strategy of T to only demand that for every fixed test strategy there is a counter-measure, one can require only that Alice's counter-strategy is successful with some given probability, etc. The above implies decidability of coercion-resistance.

6 Proof of the Main Result

In this section, we prove the main result, Theorem 4.1. The section is organized as follows:

- In Section 6.1, we introduce the central bisimulation-like concepts that allow us to perform model checking on a finite model $\mathcal{C}_{Pr}/_{\equiv}$ instead of the infinite model $\mathcal{C}_{Pr}$,
- In Section 6.2, we introduce additional notation and a notion of state-equivalence such that roughly, the model $\mathcal{C}_{Pr}$ is bisimilar to the (finite) set of equivalence classes,
- In Sections 6.3 and 6.4, we show how actions of honest principals and the adversary can be transferred from $\mathcal{C}_{Pr}$ to the finite structure $\mathcal{C}_{Pr}/_{\equiv}$ and vice versa,
- Building on concepts established in Sections 6.2 to 6.4, Section 6.5 then contains the formal definition of the finite model $\mathcal{C}_{Pr}/_{\equiv}$,
- In Section 6.6, we use the previously established building blocks to conclude the proof of our main theorem 4.1.
- Section 6.7 briefly explains how the proof of decidability result can be generalized to some extensions of our model.

In many of the following definitions, we omit the protocol Pr, the term signature Σ^{t}, and the equational theory E from the notation—this will always be clear from the context. For the decidability proof, it is convenient to make the following assumptions about the protocol Pr, which can be made without loss of generality:

- in every protocol role, every path from the root of the role to every occurrence of the special state $\mathsf{Finished}$ has the same length, also called the *length of the protocol rule*,
- in a protocol, every role has the same length, also called the *length of the protocol*.

Both of these conditions can easily be satisfied by introducing appropriate dummy states and transitions to the protocol roles and replacing in the QAPI-formulas, variables for original final states with disjunctions including the relevant added dummy states.

6.1 Bisimulations

Although the main idea of the reason for decidability is simple—since principals perform operations that consider incoming terms to a "bounded depth" only and hence the adversary does not gain anything from sending arbitrarily complicated terms to principals, we can consider a restricted structure with a maximal depth for adversary-constructed terms—the formalization of this idea requires some technical details. The intuitive argument is enough to prove decidability for reachability properties, however we also prove that *strategic* and *epistemic* properties are maintained under the above-mentioned simplification of the protocol structure, i.e., we show that truth of every QAPI-formula is maintained.

An established tool for showing invariance of properties expressible by a certain class of formulas is to establish *bisimulations* between structures, and this is the tool that we will apply to prove our result: We show that there is a finite structure which is bisimilar to $\mathcal{C}_{Pr}$, and that this finite structure can be algorithmically constructed. Since QAPI-model checking is decidable for finite structures, our decidability result then follows (note however that our proof does not establish that the upper complexity bounds from [Sch10a] hold for protocol analysis, since the size of the finite structure we construct is not polynomial in the size of the protocol).

We give the following definition of a bisimulation from [Sch10a] (see also [Sch10b]) In the following, when Z is a binary relation on state sets, then for a state q, we write $Z(q)$ to denote the set $\{q' \mid (q, q') \in Z\}$.

Definition. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be CGSs with state sets Q_1 and Q_2, the same set of players, the same set of propositional variables, and n degrees of information. Then a relation $Z \subseteq Q_1 \times Q_2$ is a *probabilistic uniform strong alternating simulation for a coalition A from $\mathcal{C}_1$ to $\mathcal{C}_2$* if for all $(q_1, q_2) \in Z$, all $i \in \{1, \ldots, n\}$, and all players $a \in A$, there is a function $\Delta^{1\to 2}_{(i,a,q_1,q_2)}$ such that for all $A' \subseteq A$ we have

- *propositional equivalence:* q_1 and q_2 satisfy the same propositional variables,
- for all (A', q_1)-moves c_1, the (A', q_2)-move c_2 with $c_2(a) = \Delta^{1\to 2}_{(i,a,q_1,q_2)}(c_1(a))$ has the
  1. *Forward Move Property:* for each $(\overline{A'}, q_1)$-move $c_1^{\overline{A'}}$, there is a $(\overline{A'}, q_2)$-move $c_2^{\overline{A'}}$ such that for all $q_1' \in Q_1$, we have
  $$\Pr\left(\delta(q_2, c_2 \cup c_2^{\overline{A'}}) \in Z(q_1')\right) = \Pr\left(\delta(q_1, c_1 \cup c_1^{\overline{A'}}) = q_1'\right).$$
  2. *Backward Move Property:* for each $(\overline{A'}, q_2)$-move $c_2^{\overline{A'}}$, there is a $(\overline{A'}, q_1)$-move $c_1^{\overline{A'}}$ such that for all $q_1' \in Q_1$, we have

$$\Pr\left(\delta(q_2, c_2 \cup c_{q_2}^{\overline{A'}}) \in Z(q_1')\right) = \Pr\left(\delta(q_1, c_1 \cup c_1^{\overline{A'}}) = q_1'\right).$$

- *Move Uniformity:* If $(q_1, q_2), (q_1', q_2') \in Z$ with $q_1 \sim_{\mathsf{eq}_1^i(a)} q_1'$ and $q_2 \sim_{\mathsf{eq}_1^i(a)} q_2'$, then $\Delta^{1\to 2}_{(i,a,q_1,q_2)} = \Delta^{1\to 2}_{(i,a,q_1',q_2')}$,
- *Uniformity:* for all $a \in A$, and all $(q_1', q_2') \in Z$, if $q_2 \sim_{\mathsf{eq}_2^i(a)} q_2'$, then $q_1 \sim_{\mathsf{eq}_1^i(a)} q_1'$.
- *Knowledge Transfer:* if $q_1' \sim_{\mathsf{eq}_i^1(A')} q_1$, then there is some $q_2' \in Q_2$ such that $q_2' \sim_{\mathsf{eq}_i^2(A')} q_2$ and $(q_1', q_2') \in Z$.
- *Uniqueness:* For all $q_2 \in Q_2$, there is exactly one $q_1 \in Q_1$ with $(q_1, q_2) \in Z$ (i.e., $Z^{-1} \colon Q_2 \to Q_1$ is a function).

If we have probabilistic uniform strong alternating simulations in both directions, and the two simulations agree on the related states in a certain manner, we have a bisimulation:

Definition. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be concurrent game structures. Then a *probabilistic bisimulation* for a coalition A between $\mathcal{C}_1$ and $\mathcal{C}_2$ is a pair of relations (Z_1, Z_2) such that

- Z_1 is a probabilistic strategy simulation for A from $\mathcal{C}_1$ to $\mathcal{C}_2$,
- Z_2 is a probabilistic strategy simulation for A from $\mathcal{C}_2$ to $\mathcal{C}_1$,
- $Z_1^{-1} \circ Z_2^{-1}$ and $Z_2^{-1} \circ Z_1^{-1}$ are idempotent.

Bisimulations ensure that the related structures satisfy exactly the same formulas:

Theorem 6.1 ([Sch10a]). *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be concurrent game structures, let $\mathbb{A}$ be a set of coalitions such that (Z_1, Z_2) is a probabilistic bisimulation for every $A \in \mathbb{A}$ between $\mathcal{C}_1$ and $\mathcal{C}_2$, let q_1 be a state of $\mathcal{C}_1$, let q_2 be a state of $\mathcal{C}_2$ such that $(q_1, q_2) \in Z_1$ and $(q_2, q_1) \in Z_2$. Let φ be a quantified strategy formula for $\mathcal{C}_1$ (and thus for $\mathcal{C}_2$) such that every coalition appearing in φ is an element of $\mathbb{A}$. Then $\mathcal{C}_1, q_1 \models \varphi$ if and only if $\mathcal{C}_2, q_2 \models \varphi$.*

This theorem is the key ingredient for our decidability proof: As mentioned above, we will establish that there is a finite structure $\mathcal{C}_{Pr}/\equiv$ and a probabilistic bisimulation between this one and $\mathcal{C}_{Pr}$. The construction of $\mathcal{C}_{Pr}/\equiv$ follows the above intuition: Essentially we disallow the adversary from sending terms exceeding a certain maximal depth to honest principals, and additionally restrict the adversary to using only finitely many different nonces. The latter restriction can be made without loss of generality if the size and number of the terms is finitely bounded. This results in the finite structure $\mathcal{C}_{Pr}/\equiv$.

Hence our main result, Theorem 4.1 immediately follows from the decidability result for model checking a finite structure and a QAPI-formula proven in [Sch10a] and the following Theorem:

Theorem 6.2. *There is an algorithm which, on input Pr, computes a finite concurrent game structure $\mathcal{C}_{Pr}/\equiv$ such that there is a relation Z which is a probabilistic bisimulation between $\mathcal{C}_{Pr}$ and $\mathcal{C}_{Pr}/\equiv$ for every coalition, and the initial states of $\mathcal{C}_{Pr}$ and $\mathcal{C}_{Pr}/\equiv$ are identical.*

6.2 Notation and the key equivalence notion on states

For proving the main result, we introduce some additional notation. A lot of the objects introduced here and in the remainder of Section 6 depend on the protocol Pr, however in order to increase readability we do not always make this dependence explicit in the notation—the protocol will always be clear from the context.

Definition. Let Pr be a protocol over the term signature Σ^{t} with equational theory E, let q be a state in $\mathcal{C}_{Pr}$. Then

- $\mathrm{ar}(\Sigma^{\mathsf{t}})$ is the maximal arity of a symbol in Σ^{t},
- *depth* (E) is the maximal depth of a term appearing as the left- or righthand-side of an equation in E,
- d_{Pr} is the product of (the maximal depth of a term appearing in one of the descriptions of the protocol roles plus 1) and *depth* (E),
- if q is a state which is not initial, then $\mathit{pred}(q)$ denotes the unique predecessor state of q in $\mathcal{C}_{Pr}$,
- $\mathit{prvst}(q)$ denotes the number of steps needed to reach q in a protocol run, i.e., if q is an initial state then $\mathit{prvst}(q) = 0$, and otherwise $\mathit{prvst}(q) = \mathit{prvst}(\mathit{pred}(q)) + 1$.

We introduce some notation that allows us to succinctly refer to certain elements and subterms of larger terms. In the following, we regard terms as trees in the natural way.

Definition. Let $t = (t_1, \ldots, t_n)$ be a sequence of terms over the signature Σ^{t}, let u be a term over Σ^{t}, and let *path* be a path (i.e., a sequence of natural numbers bounded by $\mathrm{ar}(\Sigma^{\mathsf{t}})$), let i be a natural number, and let $p = (i, \mathit{path})$, then

- p is a *position*,
- $u{\downarrow}\mathit{path}$ is the subterm of u whose root is the vertex reached when following the path *path* starting in the root of u. If this path uses non-existing successors in u, then $u{\downarrow}\mathit{path} = \texttt{error}$ (where `error` is a special symbol not used anywhere else),
- $u(\mathit{path})$ is the label of the root node of $u{\downarrow}\mathit{path}$ (where the label of `error` is `error`),
- $|t| = n$,
- if path_2 is a path, then $p \circ \mathit{path}_2$ is defined as the position $(i, \mathit{path}_1\mathit{path}_2)$ (in this case we say that $p \circ \mathit{path}_2$ is a *extension* of p, and p is a *prefix* of $p \circ \mathit{path}_2$),
- $t{\downarrow}p = t_i{\downarrow}\mathit{path}$ and $t(p) = t_i(\mathit{path})$ (both of these are `error` if $i > n$)
- *depth* (p) is the length of *path*.

The following defines a natural notion of equivalence of term sequences: For a natural number d, $\sim_d$-equivalence requires that two sequences "look the same" when we only consider elements and subterms appearing down to depth d: The elements in these positions must be the same, and equality between positions

must hold in one sequence if and only if it holds in the other (note however that the equality of the subterms must hold down to the leaves in the trees, notwithstanding the depth). We will later use this definition to define a similar equivalence on states of a cryptographic protocol: These are "equivalent," if the honest principals are in the same protocol states and the so-far observed terms are equivalent to a sufficient degree.

Definition. Let t^1 and t^2 be sequences of terms, and let $d \in \mathbb{N}$. Then $t^1 \sim_d t^2$ (t^1 and t^2 are *d-equivalent*), if for every pair of positions p_1, p_2 with $depth(p_1), depth(p_2) \leq d$, we have

- $t^1(p_1) = t^2(p_2)$, and
- $t^1{\downarrow}p_1 = t^1{\downarrow}p_2$ if and only if $t^2{\downarrow}p_1 = t^2{\downarrow}p_2$.

Note that it $t^1 \sim_d t^2$ for some $d \geq 0$, then $\left|t^1\right| = \left|t^2\right|$ (this follows due to the equality of elements on the first level, and the fact that such an element is `error` if and only if the referenced term does not exist, i.e., $|t| \geq n$ if and only if $t((n, \epsilon)) \neq \texttt{error}$).

Definition. Let Pr be a k-roles protocol, and let q be a state of $\mathcal{C}_{Pr}$, then $terms(q)$ is the sequence containing all terms from the sequences $\mathcal{M}_1$, $\mathcal{M}_2$, ..., $\mathcal{M}_k$, and $\mathcal{M}_{\mathcal{A}}$. For a position p, with $q{\downarrow}p$ we denote $terms(q){\downarrow}p$, and with $q(p)$ we denote $terms(q)(p)$.

Hence $terms(q)$ contains the set of all terms sent, received, and parsed by the adversary and principals. Equivalence of states is now defined in the natural way:

Definition. Let Pr be a protocol, let q_1, q_2 be states in $\mathcal{C}_{Pr}$, and let $d \in \mathbb{N}$. Then $q_1 \sim_d q_2$ (q_1 and q_2 are *d-equivalent*), if all honest principals are in the same local state in both q_1 and q_2, the same set of identities is corrupted in q_1 and q_2, and $terms(q_1) \sim_d terms(q_2)$.

Note that this definition of equivalence does *not* refer to the indistinguishability relations of the principals: If $q_1 \sim_d q_2$, then there may very well be tests that a principal can perform to distinguish these states. However, the tests that occur in the protocol description will yield the same result in states that are equivalent to a "sufficient" degree (see later), so that the available choices for the principal are the same. This is the key property of this construction: We use the above equivalence of states to show that if $q_1 \sim_d q_2$ for a sufficiently large d, then both the adversary and the honest principals have exactly the same strategic options in q_1 and q_2, even if these options take into account the (possible different) knowledge in the states q_1 and q_2. In order to make this precise, we now define the level of $\sim_d$-equivalence we require after each number of protocol steps, this is done with the function $\mathsf{eqdeg}(.)$. The purpose of this function is the following: If q_1 and q_2 are states such that all honest principals are in the same local state in q_1 and q_2 (and thus in particular $prvst(q_1) = prvst(q_2) =: s$) then $d := \mathsf{eqdeg}(s)$ has the property that if $q_1 \sim_d q_2$, then q_1 and q_2 are "strategically equivalent" (proving this is the main work required to show our result). To see that this degree depends on the state, observe that when the protocol run is

over, we are not interested in the terms at all anymore, but only need to require that principals have reached the same local protocol state. In previous states however, the question which sub-terms of incoming messages for principals are identical to previously-received messages is very relevant, as the question which tests performed by honest principals are satisfied in the state clearly depends on this.

Proposition 6.3. *Let q^1 and q^2 be states in $\mathcal{C}_{Pr}$ such that $q^1 \sim_0 q^2$. Then $prvst(q^1) = prvst(q^2)$.*

Proof. By definition of equivalence, all principals are in the same local state in q^1 and q^2. The number of steps performed in the protocol run is the same as the number of steps performed by any principal (since our model is concurrent). Hence the claim follows.

We now formally define the function $\mathsf{eqdeg}\,(.)$ as explained above:

Definition. For a k-protocol Pr with length ℓ and a natural number s, let

- $\#e(s) = k \cdot (\mathrm{ar}(\Sigma^{\mathsf{t}})^s)$,
- $mdagdpth_{\mathcal{A}}(s) = 2^{\#e} \cdot (s+1)$,
- let $\mathsf{eqdeg}\,(0) = \ell \cdot d_{Pr}$,
- for $s \geq 1$, let

$$\begin{aligned}\mathsf{eqdeg}\,(s+1) = \quad & 2\mathsf{eqdeg}\,(s) + 4 \cdot d_{Pr} \\ & + 2mdagdpth_{\mathcal{A}}(\mathsf{eqdeg}\,(s) + 2 \cdot d_{Pr}) \\ & + 2 \cdot \ell \cdot d_{Pr}.\end{aligned}$$

- for a state q, let $\mathsf{eqdeg}\,(q) = \mathsf{eqdeg}\,(\ell - prvst(q))$.
- for two states q_1 and q_2 of $\mathcal{C}_{Pr}$ with $prvst(q_1) = prvst(q_2)$, let $q_1 \equiv q_2$ if
  1. $q_1 \sim_{\mathsf{eqdeg}(q_1)} q_2$, and
  2. either q_1 and q_2 both are initial states, or $pred(q_1) \equiv pred(q_2)$

The condition that if $q_1 \equiv q_2$, then $pred(q_1) \equiv pred(q_2)$ implies that if two states are equivalent, then their histories are equivalent as well.

6.3 Move Transfer For Honest Principals

We now show that honest principals have essentially "the same options" in states q_1 and q_2 if $q_1 \equiv q_2$, i.e., essentially we show the forward move property for honest principals. The main work needed to be done here is to prove that the effects of actions performed by honest principals are limited to a certain depth in the resulting protocol state. This is intuitively clear, since the operations of honest principals only use terms with bounded depth—hence both modifications performed and analysis carried out by principals only concern parts of the message down to some bounded depth.

The following lemma makes this precise by showing that when constructing new terms from a term sequence using terms constructed from Σ^{t}, the resulting term contains only references of limited depth into the original term sequence.

Lemma 6.4. *Let $\mathcal{M}$ be a sequence of messages, let m be a term, and let $r = [[m[\mathcal{M}/x]]]$. Then there are a term s and positions $q_1, \ldots, q_n$ such that for all relevant i,*

1. $\mathit{depth}(q_i) \leq \mathit{depth}(m) \cdot \mathit{depth}(\mathrm{E}) =: d$,
2. $r = s[(\mathcal{M}{\downarrow}q_1)/x_1, \ldots, (\mathcal{M}{\downarrow}q_n)/x_n]$,
3. $\mathit{depth}(s) \leq (\mathit{depth}(m) + 1) \cdot \mathit{depth}(E)$.

In addition, the term s only depends on m and on the entries of $\mathcal{M}$ of depth at most d.

Proof. Let $r' = m[\mathcal{M}/x]$, then by definition $r = [[r']]$, i.e., r is obtained from r' by exhaustive application of equations from the theory E. Since E is a subterm theory, each application of an equation replaces a subterm t of an intermediate result r'_i with a subterm of t or a constant. In the latter case, no reference to the term $\mathcal{M}$ is obtained, hence we only consider the first case. Since $\mathcal{M}$ is a sequence of messages, we know that each application of an equation must consume a symbol from m (no rewrite rule can be applied to $\mathcal{M}$). Therefore, at most j applications of equations are required to obtain r from r', where j is the size of m (i.e., the number of symbols contained in m).

Each application of an equation from E leading from r' to r transforms the intermediate result r^i into another intermediate result $r^{i'}$. Since E is a subterm theory, we again know that some subterm t^i or r^i was replaced with a subterm $t^{i'}$ of t^i (we can again ignore the case of constants). In particular, $t^{i'}$ appears in r^i with depth at most $\mathit{depth}(\mathrm{E})$.

Inductively, r is obtained from r' by using subterms of $\mathcal{M}$ that appear at depth of at most $\mathit{depth}(\mathrm{E}) \cdot j$ and adding elements with depth of at most $\mathit{depth}(m)$. This results in the term s with the claimed properties. Each of the positions q_i in the final term has depth at most $\mathit{depth}(m) \cdot \mathit{depth}(\mathrm{E})$, since each position is only affected by the operations appearing in one branch of m. The depth of the term s is at most the depth of m increased by $\mathit{depth}(\mathrm{E})$ in every step, where again each position is only affected by the operations in a single branch of m. Hence, the depth of s is at most $\mathit{depth}(m) + \mathit{depth}(m) \cdot \mathit{depth}(\mathrm{E})$.

We now show that $\equiv$-equivalence is maintained under adding specific terms—the following lemma describes the situation where principals perform their protocol rules and send out the corresponding terms. In the later application of the lemma, $\mathcal{M}_1$ and $\mathcal{M}_2$ will be the sequences of messages received by principals (including the adversary) in states q_1 and q_2 with $q_1 \equiv q_2$, and $\mathcal{M}'_1$ and $\mathcal{M}'_2$ will be the messages received in the states q'_1 and q'_2 obtained from q_1 and q_2 by letting the honest principals perform the same move in both steps. The terms m_i are the ones from the send sequence of our protocols, and are used to construct messages sent by honest principals. The second application of the lemma is when principals build new messages not to send to other principals, but to perform the tests as part of their parsing sequence.

Lemma 6.5. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be term sequences, let $m_1, \ldots, m_k$ be terms, and let $d_1, d_2 \in \mathbb{N}$ such that $\mathcal{M}_1 \sim_{d_1} \mathcal{M}_2$, and $(\mathit{depth}(m_i) + 1) \cdot \mathit{depth}(E) \leq d_2$ for all relevant i.*

For $a \in \{1, 2\}$, let $\mathcal{M}'_a$ be obtained from $\mathcal{M}_a$ by adding the terms $[[m_1[\mathcal{M}_a/x]]]$, $\ldots$, $[[m_k[\mathcal{M}_a/x]]]$. Then $\mathcal{M}'_1 \sim_{d-2d_2} \mathcal{M}'_2$.

In a protocol run, Lemma 6.5 covers the result of actions performed by honest principals: The messages sent by honest principals are obtained by constructing new terms, which may reference elements of the sequence of previously received messages. Due to Lemma 6.4, we know that the depth of reference into previously-received terms is limited by a constant that only depends on the protocol and the equational theory E. This lemma essentially shows that if two states are "sufficiently" equivalent, and the principals then perform the same moves, then the resulting states are equivalent (to a slightly lesser degree). This fact will be an ingredient in the proof of the forward- and backward move properties required by the bisimulation. We now prove the lemma.

Proof. Due to Lemma 6.4, there are terms $s_1, \ldots, s_m$ and positions $q_1, \ldots, q_n$ such that the depth of each q_i is at most $\mathit{depth}(E) \cdot \max\{\mathit{depth}(m_j) \mid 1 \leq j \leq m\} \leq d_2$, the depth of each s_i is at most $(\mathit{depth}(m_i) + 1) \cdot \mathit{depth}(E) \leq d_2$, and $\mathcal{M}'_a$ is obtained from $\mathcal{M}_a$ by adding the terms $s_1[(\mathcal{M}{\downarrow}q_1)/x_1, \ldots, (\mathcal{M}{\downarrow}q_n)/x_n]$, $\ldots$, $s_m[(\mathcal{M}{\downarrow}q_1)/x_1, \ldots, (\mathcal{M}{\downarrow}q_n)/x_n]$. Note that not all of the variables necessarily appear in all of the s_i.

For $a \in \{1, 2\}$, let $\mathcal{M}_a = (t^a_1, \ldots, t^a_{|\mathcal{M}_a|})$. We denote $m_i[\mathcal{M}_a/x]$ with s^a_i for $a \in \{1, 2\}$ and $i \in \{1, \ldots, k\}$.

We can without loss of generality assume that among the positions q_i, for each $j \leq |\mathcal{M}_1|$ (which must be identical to $|\mathcal{M}_2|$ since $\mathcal{M}_1 \sim_{d_1} \mathcal{M}_2$), there is a position of the form (j, ϵ). If these are not present, we add these positions and prove the claim for this extended set of positions (note that these positions have depth 0). Similarly, we can assume that for each $i \in \{1, \ldots, n\}$, there is some term s_{j_i} which is the variable x_i. Again, if these are not present we add them (noting again that all these terms have depth 0). Since we now have terms s_{j_i} such that $s^a_{j_i}$ is the i-th message in the sequence $\mathcal{M}_a$, it suffices to prove that $u^1 = (s^1_1, \ldots, s^1_k)$ and $u^2 = (s^2_1, \ldots, s^2_k)$ are $(d_1 - 2d_2)$-equivalent.

Hence let p_1 and p_2 be positions, where for $b \in \{1, 2\}$, we have $p_b = (i_b, \mathit{path}_b)$, and $|\mathit{path}_b| \leq d_1 - 2d_2$. Without loss of generality, we assume $i_b = b$.

We first show $u^1(p_1) = u^2(p_1)$ (note that in this proof, we only use the fact that $\mathit{depth}(p_1) \leq d_1 - d_2$—we will refer to this slightly stronger result in the second part of the proof). By construction, (since $p_1 = (1, \mathit{path}_1)$), we have $u^a(p_1) = s^a_1(\mathit{path}_1)$. If path_1 does not visit a position in s_1 which is a variable, then obviously $s^a_1(\mathit{path}_1) = s_1(\mathit{path}_1)$, and it follows that $u^1(p_1) = s^1_1(\mathit{path}_1) = s_1(\mathit{path}_1) = s^2_1(\mathit{path}_1) = u^2(p_1)$ as required. Hence assume that when following path_1 in s_1, we encounter a variable, without loss of generality the variable x_1. Let $\mathit{path}_1 = w_1 w_2$, such that $s_1(w_1) = x_1$. It then follows for $a = 1, 2$ that $u^a(p_1) = s^a_1(\mathit{path}_1) = s^a_1(w_1 w_2) = (s^a_1 \downarrow w_1)(w_2) = x_1[(u^a{\downarrow}q_1)/x_1, \ldots](w_2) = (\mathcal{M}_a{\downarrow}q_1)(w_2) = \mathcal{M}_a(q_1 \circ w_2) = t^a_1(q\mathit{path}_1 w_2)$. Since $|w_2| \leq \mathit{depth}(p_1) \leq d_1 - d_2$,

and $\mathit{depth}(q_1) \leq d_2$, it follows that $\mathit{depth}(q_1) + |w_2| \leq d_1$. Hence we know (since $\mathcal{M}_1 \sim_{d_1} \mathcal{M}_2$) that $\mathcal{M}_1(q_1 \circ w_2) = \mathcal{M}_2(q_1 \circ w_2)$, and it follows that $u^1(p_1) = \mathcal{M}_1(q_1 \circ w_2) = \mathcal{M}_2(q_1 \circ w_2) = u^2(p_1)$, as required.

We now show that $u^1 \downarrow p_1 = u^1 \downarrow p_2$ if and only if $u^2 \downarrow p_1 = u^2 \downarrow p_2$. It obviously suffices to prove one direction, hence assume $u^1 \downarrow p_1 = u^1 \downarrow p_2$. We show the claim by induction over the depth restrictions for the p_b, and the s_i. In the following, for $\alpha, \gamma \in \mathbb{N}$, we say that the pair (α, γ) *holds*, if the following implication is true: For all positions p_1, p_2, and terms s_1, s_2, if $\mathit{depth}(p_b) \leq \alpha$ and $\mathit{depth}(s_b) \leq \gamma$ for $b \in \{1, 2\}$, then $u^1 \downarrow p_1 = u^1 \downarrow p_2$ implies $u^2 \downarrow p_1 = u^2 \downarrow p_2$. To prove the lemma, we need to show that (α, γ) holds for all values with $\alpha + \gamma \leq d_1 - d_2$ (the claim of the lemma involves only positions p_1, p_2 with depth at most $d_1 - 2d_2$, and terms s_i with $\mathit{depth}(s_i) \leq d_2$—recall that we assumed without loss of generality that the position p_b refers to the term s_b for $b \in \{1, 2\}$, hence we only consider the first two terms).

For the base of the induction, we show that $(d_1 - d_2, 0)$ holds. In this case, the terms s_1 and s_2 have depth 0, i.e., they are variables or constants (where we treat the empty term ϵ as a constant), and $\mathit{depth}(p_b) \leq d_1 - d_2$ for $b = 1, 2$. We can without loss of generality assume that if s_b is a variable, then it is the variable x_b, and if s_b is a constant, then it is the constant cons_b. Thus only the variables x_1 and x_2, and only the positions q_1 and q_2 are relevant among the q_i. Again, without loss of generality, we assume that $q_b = (b, \mathit{qpath}_b)$. Now if s_b is the variable x_b, then we have (for $a \in \{1, 2\}$):

$$\begin{aligned} u^a \downarrow p_b &= s_b^a \downarrow \mathit{path}_b \\ &= (x_b[\ldots, (\mathcal{M}_a \downarrow q_b) \ldots]) / x_b \downarrow \mathit{path}_b \\ &= (\mathcal{M}_a \downarrow q_b) \downarrow \mathit{path}_b \\ &= (\mathcal{M}_a \downarrow (b, \mathit{qpath}_b)) \downarrow \mathit{path}_b \\ &= (t_b^a \downarrow \mathit{qpath}_b) \downarrow \mathit{path}_b \\ &= t_b^a \downarrow (\mathit{qpath}_b \circ \mathit{path}_b). \end{aligned}$$

If s_b is the constant cons_b, then we have (for $a \in \{1, 2\}$): $u^a \downarrow p_b = s_b^a \downarrow \mathit{path}_b = \mathit{cons}_b \downarrow \mathit{path}_b$. We now make a case distinction.

Assume that both s_1 and s_2 are variables, i.e., $s_1 = x_1$, and $s_2 = x_2$. Then $t_1^1 \downarrow (\mathit{qpath}_1 \circ \mathit{path}_1) = u^1 \downarrow p_1 = u^1 \downarrow p_2 = t_2^1 \downarrow (\mathit{qpath}_2 \circ \mathit{path}_2)$. Since $|\mathit{qpath}_b| + |\mathit{path}_b| \leq d_2 + d_1 - d_2 = d_1$, and $\mathcal{M}_1 \sim_{d_1} \mathcal{M}_2$, this implies $t_1^2 \downarrow (\mathit{qpath}_1 \circ \mathit{path}_1) = t_2^2 \downarrow (\mathit{qpath}_2 \circ \mathit{path}_2)$. Due to the above (and since both s_b are variables), we therefore have $u^2 \downarrow p_1 = t_1^2 \downarrow (\mathit{qpath}_1 \circ \mathit{path}_1) = t_2^2 \downarrow (\mathit{qpath}_2 \circ \mathit{path}_2) = u^2 \downarrow p_2$, as required.

Assume that both s_1 and s_2 are constants, i.e., $s_1 = \mathit{cons}_1$, and $s_2 = \mathit{cons}_2$. Then $\mathit{cons}_1 \downarrow \mathit{path}_1 = u^1 \downarrow p_1 = u^1 \downarrow p_2 = \mathit{cons}_2 \downarrow \mathit{path}_2$. Hence $u^2 \downarrow p_1 = \mathit{cons}_1 \downarrow \mathit{path}_1 = \mathit{cons}_2 \downarrow \mathit{path}_2 = u^2 \downarrow \mathit{path}_2$ as required. (Considering "subterms" of constants here only serves as a unified means to cover the cases where the path is empty (and thus the term is "legal") or not (in which the term is the `error`-symbol.))

Assume that one is a variable, the other a constant, without loss of generality, s_1 is the variable x_1, and s_2 is the constant cons_2. From the above, we

thus know that $t_1^1 \downarrow (qpath_1 \circ path_1) = u^1 \downarrow p_1 = u^1 \downarrow p_2 = cons_2 \downarrow path_2$. Since $\mathcal{M}_1 \sim_{d_1} \mathcal{M}_2$, and $|qpath_1 \circ path_1| \leq d_1$, we know that $t_1^2(qpath_1 \circ path_1) = t_1^1(qpath_1 \circ path_1) = cons_2(path_2)$. Note that this "subterm" is either the constant $cons_2$ or the `error`-symbol. Since each term in the t^a is a well-constructed term over Σ^{t}, the occurrence of $cons_2$ in t_1^2 cannot have a successor, thus equality of elements here implies equality as subterms, hence we have $t_1^2 \downarrow (qpath_1 \circ path_1) = cons_2 \downarrow path_2$. Hence it follows that $u^2 \downarrow p_1 = t_1^2 \downarrow (qpath_1 \circ path_1) = cons_2 \downarrow path_2 = u^2 \downarrow p_2$, as required. This covers all possible cases, and thus completes the proof of the base of the claim that $(d_1 - d_2, 0)$ holds.

Now assume inductively that (α, γ) is true, where $\alpha \geq 1$. We show that $(\alpha - 1, \gamma + 1)$ is true. Since we know from the above that $(d_1 - d_2, 0)$ holds, this completes the proof of (α, γ) for all $\alpha + \gamma \leq d_1 - d_2$: Hence assume that s_1 and s_2 are terms with $depth(s_1), depth(s_2) \leq \gamma + 1$, and assume that $depth(p_1), depth(p_2) \leq \alpha - 1$. Without loss of generality, we can assume that $depth(s_1) \geq depth(s_2)$, and hence in particular, $depth(s_1) = \gamma + 1$ (the case where both depths are at most γ is covered by (α, γ)). Hence, $s_1 = f(s'_1, \ldots, s'_e)$ for an e-ary function symbol f from the signature Σ^{t}. Obviously, $depth(s'_i) < depth(s_1)$. We consider several cases.

Assume that $depth(s_1) > depth(s_2)$, *and* $path_1 \neq \epsilon$, then $path_1 = c \circ path'_1$ for some $c \in \{1, \ldots, e\}$ (the case if $c > e$, i.e., the position leads to an `error`-symbol, is covered by part 1 of the proof, since then in all relevant positions, the `error`-symbol appears). It follows that for $a \in \{1, 2\}$, we have $u^a \downarrow p_1 = s_1^a \downarrow path_1 = s_c'^a \downarrow path'_1$ (here, $s_i'^a$ for some i is defined analogously to s_i^a, where an occurrence of a variable x_j is replaced with $\mathcal{M}_a \downarrow q_j$). Since $u^1 \downarrow p_1 = u^1 \downarrow p_2$, we have that $s_c'^1 \downarrow path'_1 = u^1 \downarrow p_2$

These positions can be described using terms s'_c, s_2, where the depth of each is at most γ, and paths $path'_1, path_2$, where $|path'_1|, |path_2| \leq \alpha - 1 \leq \alpha$. Since (α, γ) holds, we know that the above equality implies $s_c'^2 \downarrow path'_1 = u^2 \downarrow p_2$, and due to the above this is equivalent to $u^2 \downarrow p_1 = u^2 \downarrow p_2$, as required.

Assume that $depth(s_1) > depth(s_2)$, *and* $path_1 = \epsilon$, then $u^a \downarrow p_1 = s_1^a \downarrow path_1 = s_1^a = f(s_1'^a, \ldots, s_e'^a)$. Since $u^1 \downarrow p_1 = u^1 \downarrow p_2$, we know that $u^1(p_2) = u^1(p_1) = f$, and from part 1 of the proof we have that $u^2(p_2) = u^1(p_2) = f$. We further know that for all steps c, we have $u^1 \downarrow (p_1 \circ c) = u^2 \downarrow (p_2 \circ c)$.

Due to the above, we know that $u^2 \downarrow p_1 = s_1^2$, hence we know that $u^2(p_1) = f = u^2(p_2)$. To prove that $u^2 \downarrow p_1 = u^2 \downarrow p_2$, it thus remains to show that for all steps c, we have $u^2 \downarrow (p_1 \circ c) = u^2 \downarrow (p_2 \circ c)$. From the above, it follows that $u^a \downarrow (p_1 \circ c) = (u^a \downarrow p_1) \downarrow c = s_1^a \downarrow c = s_c'^a$. Hence the involved positions in u^a can be described with terms s'_c and s_2, where the depth of these is $\leq \gamma$, and positions p'_1, p'_2 with depth $\leq \alpha$ (instead of p_1 and p_2, where $depth(p_1), depth(p_2) \leq \alpha - 1$, we consider a position p'_1 with depth 0, and a position $p_2 \circ c$, with depth one more that p_2). Since we know that (α, γ) holds, the fact that equality for the involved positions holds in u^1 transfers to equality in u^2, as required.

Assume that $depth(s_1) = depth(s_2) = \gamma + 1$, in this case we have $s_b = f_b(s_{b,1}, \ldots, s_{b,e_b})$, where f_b is an e_b-ary function symbol from Σ^{t}, and $s_{b,i}$ are terms with $depth(s_{b,i}) \leq \gamma$. Analogously to the s_b^a, for $a, b \in \{1, 2\}$, and $i \leq e_b$,

we define $s^a_{b,i}$ to be the term obtained from $s_{b,i}$ by replacing every occurrence of a variable x_j with the term $\mathcal{M}_a \downarrow q_j$. Now observe that if $path_b = \epsilon$, then $u^a \downarrow p_b = s^a_b \downarrow \epsilon = s^a_b$, and for a step c, we have that $u^a \downarrow (p_b \circ c) = (u^a \downarrow p_b) \downarrow c = s^a_b \downarrow c = s^a_{b,c}$.

Analogously, if $path_b = c_b path'_b$, then $u^a \downarrow p_b = s^a_b \downarrow (c_b path'_b) = (s^a_b \downarrow c_b) \downarrow path'_b = s^a_{b,c_b} \downarrow path'_b$, and for a step c, we have that $u^a \downarrow (p_b \circ c) = (u^a \downarrow p_b) \downarrow c = (s^a_{b,c_b} \downarrow path'_b) \downarrow c = s^a_{b,c_b} \downarrow (path'_b \circ c)$.

We now consider two subcases:

Assume that $depth\,(s_1) = depth\,(s_2) = \gamma + 1$ *and* $path_1 = path_2 = \epsilon$. Since $u^1 \downarrow p_1 = u^1 \downarrow p_2$, due to the above we have that $f_1 = u^1(p_1) = u^1(p_2) = f_2$, and hence $e_1 = e_2$ (which we will denote with e). From part 1 of the proof, we know that $u^2(p_1) = u^1(p_1) = f_1$, and analogously $u^2(p_2) = u^1(p_2) = f_1$. Hence it remains to show that for all $c \in \{1, \ldots, e\}$, we have that $u^2 \downarrow (p_1 \circ c) = u^2 \downarrow (p_2 \circ c)$ (where we know that these equalities hold in u^1). From the above, and since $path_1 = path_2 = \epsilon$, we know that $u^a \downarrow (p_b \circ c) = s^a_{b,c}$. Hence the involved positions can be described with terms s'_1, s'_2 with $depth\,(s'_1), depth\,(s'_2) \leq \gamma$, and positions p'_1, p'_2 with $depth\,(p'_1), depth\,(p'_2) = 0 \leq \alpha$. Since subterm-equality for the corresponding positions holds in u^1, and we know that (α, γ) holds, equality also holds in u^2 as required.

Assume that $depth\,(s_1) = depth\,(s_2) = \gamma + 1$*, one* $path_b$ *is empty, the other is not.* Without loss of generality, assume that $path_1 = \epsilon$, and $path_2 = c_2 path'_2$. Then we know that $u^a(p_1) = s^a_1(\epsilon) = f_1$. Since $u^1 \downarrow p_1 = u^1 \downarrow p_2$, it follows that $u^1(p_2) = u^1(p_1) = f_1$, and thus (due to part 1 of the proof), we have $u^2(p_b) = u^1(p_b) = f_1$ for $b = 1, 2$. It remains to show that for all $c \in \{1, \ldots, e_1\}$, we have $u^2 \downarrow (p_1 \circ c) = u^2 \downarrow (p_2 \circ c)$ (where we again know that this equality is true in u^1). Due to the above, we know that $u^a \downarrow (p_1 \circ c) = s^a_{1,c}$, and $u^a \downarrow p_2 \circ c = s^a_{2,c_2} \downarrow (path'_2 \circ c)$. Hence the involved positions can be described with terms $s_{1,c}$ and s_{2,c_2}, which have depth $\leq \gamma$, and positions $(1, c)$ and $(2, path'_2 c)$, which have depth $1 \leq \alpha$ and $depth\,(p_2) \leq \alpha - 1 \leq \alpha$ (note $\alpha \geq 1$).

Since subterm-equality for the corresponding positions holds in u^1, and we know that (α, γ) holds, equality also holds in u^2 as required.

Assume that $depth\,(s_1) = depth\,(s_2) = \gamma + 1$*,* $path_1 = c_1 path'_1$*, and* $path_2 = c_2 path'_2$*,* then due to the above we have that $u^a \downarrow p_b = s^a_{b,c_b} \downarrow path'_b$. Hence the involved positions can be described with terms s_{1,c_1} and s_{2,c_2} with depth at most γ, and positions p'_1 and p'_2 with $depth\,(p'_b) = depth\,(p_b) - 1 \leq \alpha$. Again, we know from induction that (α, γ) holds, and thus equality for the positions in u^1 implies the corresponding equality in u^2. This completes the case distinction and therefore the proof.

To establish the move transfer functions for honest principals, the following proposition is the key in this construction. It states that in "equivalent" states, principals have the same moves available.

Proposition 6.6. *Let Pr be a protocol, and let* q_1, q_2 *be states in* $\mathcal{C}_{Pr}$ *such that* $q_1 \equiv q_2$*. Then for an honest principal* $a \in \{1, \ldots, k\}$*, we have that* $\Delta(q_1, a) = \Delta(q_2, a)$*.*

Proof. Let $d_1 = \mathsf{eqdeg}(q_1)$, which is identical to $\mathsf{eqdeg}(q_2)$, since $q_1 \equiv q_2$. Let $\mathcal{M}_a = terms(q_a)$ for $a \in \{1, 2\}$. We then have $\mathcal{M}_1 \sim_{d_1} \mathcal{M}_2$.

By definition of d_{Pr}, for any test (M, M') appearing in the protocol and $m \in \{M, M'\}$ we have $(depth(m) + 1) \cdot depth(E) \leq d_{Pr}$.

Therefore, Lemma 6.5 implies that the terms constructed as left-hand or right-hand of the comparisons of the tests are $(d_1 - 2 \cdot d_{Pr})$-equivalent. Since $d_1 - 2 \cdot d_{Pr} \geq 0$ if the states are non-final (this follows directly from the definition of $\mathsf{eqdeg}(.)$, since we can without loss of generality assume that the length of the protocol is at least 1), it follows that the resulting terms are identical in the state q_1 if and only if they are identical in q_2. Since due to the definition of $\mathcal{C}_{Pr}$ the available moves of an honest principal only depend on the outcome of the involved tests, it follows that honest principals have the same available moves in q_1 and q_2 in any non-final state. The proposition trivially holds in final states of the protocol as here honest principals only have dummy moves available.

6.4 Move Transfer for the Adversary

We now show the analogous result of Section 6.3 for the adversary: If $q_1 \equiv q_2$, then every move of the adversary in q_1 can be transformed into one in q_2 such that the application of these moves again leads to a pair of equivalent states (provided that the honest principals perform the same moves in q_1 and q_2, as they can due to Proposition 6.6).

The situation for adversary moves is more complicated than for principal moves for several reasons: Adversary moves may be terms of arbitrary complexity, which can reference terms appearing in arbitrary depth in the states q_1 or q_2. When transferring an adversary move from one state to the other, we have to carefully ensure that up to the required depth, the same equalities hold in both resulting states. Since the adversary cannot send arbitrary terms, but only those which result from applications of $\mathcal{A}$-terms to the messages he received previously during the protocol run, we start with an analysis of the structure of adversary-constructable terms. In the following, the extraction-depth of a term t with a variable x is the maximal depth of references into $\mathcal{M}$ that result in replacing x with $\mathcal{M}$ in t and then determining the normal form with respect to the theory E. Due to Lemma 6.4, the extraction-depth of a term t is at most $depth(t) \cdot depth(E)$.

(over all paths in t) sum of, for each operator appearing in the path, the maximal depth of an equations mentioning the operator in the equational theory E

Definition. A position p is $\mathcal{A}$-*accessible* in a state q of $\mathcal{C}_{Pr}$, if there is an $\mathcal{A}$-term $t_{\mathcal{A}}$ with extraction-depth at most $d_{Pr} \cdot prvst(q)$ such that for all states q' obtained from q by replacing the subterm at position p with a term t', we have that $t_{\mathcal{A}}[terms(q')/x] = t'$.

Intuitively, the definition requires that for the adversary, there is a "way to extract the subterm at position p from the state q." However, since the subterm at p may appear in more than one position, the technical definition has to make

sure that the "extraction" performed by the adversary-term t gives the term at position p, no matter what the term actually is. Note that the restriction on the extraction-depth of $t_{\mathcal{A}}$ is stronger than only demanding that $\mathit{depth}(p) \leq d_{Pr} \cdot \mathit{prvst}(q)$: The term $t_{\mathcal{A}}$ might need to access elements in deeper positions that allow him to gain access to the term in position p (as an example, this might be nonces used as symmetric keys). However, if a position p is $\mathcal{A}$-accessible in q, then obviously $\mathit{depth}(q) \leq d_{Pr} \cdot \mathit{prvst}(q)$.

Obviously, $\mathcal{A}$-accessibility of a position is invariant under state-equivalence, as long as equivalence holds up to a sufficient degree—this follows trivially from the definition:

Proposition 6.7. *Let q^1 and q^2 be states in $\mathcal{C}_{Pr}$, such that $q^1 \equiv q^2$. Then a position p is $\mathcal{A}$-accessible in q^1 if and only if it is $\mathcal{A}$-accessible in q^2, and the same term can be used for extracting.*

Proof. This follows since for any state q, we have that $\mathsf{eqdeg}(q) \geq d_{Pr} \cdot \mathit{prvst}(q)$: By definition, this is true for final states of the protocol. For a non-initial state q, we have that $\mathsf{eqdeg}(\mathit{pred}(q)) \geq \mathsf{eqdeg}(q)$, while obviously $\mathit{prvst}(q) > \mathit{prvst}(\mathit{pred}(q))$.

In the following, for a state q, we denote with $d_{\mathcal{A}}(q)$ the set of messages that the adversary can construct in the state q, i.e., the set of terms of the form $[[t[\mathcal{M}_{\mathcal{A}}/x]]]$, where t is a term from $T_{\mathcal{A}}$ and $\mathcal{M}_{\mathcal{A}}$ again denotes the sequence of messages received by the adversary so far in the protocol run leading up to the state q.

The following proposition states that terms t that the adversary can extract from the current state, and that cannot be constructed from the adversary himself have to be present in a position that is $\mathcal{A}$-accessible to the adversary. The technical requirement for t in the proposition expresses that the outmost operation of the term t has not been computed by the adversary, but by an honest principal. As an example, this may be an encryption performed by a principal (where the adversary does not know *both* the nonce used for randomization and the plaintext), or a signature of a principal where the adversary does not have the secret signature key. Intuitively, this is clear, as the results of computations of honest principals appear with limited depth in the state where the computation was first performed, and while it is possible for the adversary to "copy" a term containing the subterm in question to a position with higher depth, this does not help him accessing the subterm: For example, a principal will never decrypt a ciphertext contained so deeply in an adversary-sent term such that the normal protocol rules will never even access that position.

Proposition 6.8. *Let q be a state in $\mathcal{C}_{Pr}$, and let $t \in d_{\mathcal{A}}(q)$ be a term not of the form $t_{\mathcal{A}}[t_1/x_1, \ldots, t_n/x_n]$ for a term $t_{\mathcal{A}} \in T_{\mathcal{A}}$ with depth 1, and $t_1, \ldots, t_n \in d_{\mathcal{A}}(q)$. Then there an $\mathcal{A}$-accessible position p in q with $q{\downarrow}p = t$.*

Proof. By choice of t, the term was constructed by an honest principal. Consider the first state q' in the protocol run leading up to q in which the adversary can

construct t, let q'' be the direct predecessor of q' (Obviously, the choice of t implies that q' is not an initial state of the protocol). First consider the case that the copy of t that the adversary accesses in the state q' is constructed by a principal in the transition from q'' to q'. In this case the claim holds since results from principal computations appear with depth at most d_{Pr}, and since the adversary accesses the new copy of t, the position is $\mathcal{A}$-accessible.

This argument also covers the case when t uses more than one extraction to access the term t: The extraction-depth of each single required extraction is bound by d_{Pr}. Since different extractions are performed in parallel, the depths do not influence each other, i.e., the extraction-depth of the entire term is bound by d_{Pr}. (An example for such a situation is then access to a symmetric key is needed to decrypt t itself; if the key appears at a certain depth, this depth does not add to the depth of the decrypted message.)

Hence assume that the copy of t which the adversary accesses was computed in a transition leading to a (not necessarily direct) predecessor state of q'', or to q'' itself. Since the adversary cannot extract t in q'', a partial extraction must have been performed by a principal, i.e., an honest principal constructed a message using an extraction referring into the superterm of the relevant copy of t. Between the root of the extracted superterm and the appearance of t itself, no adversary-computed subterm can appear, since gaining access to such a term would not help the adversary in extracting t (this term was constructable by the adversary in q'' already).

The path from the root of the extracted subterm to the root of t therefore contains only principal-computed computations, and thus is restricted in depth by $d_{Pr} \cdot prvst(q'')$. The result of the partial extraction appears in q' at depth of at most d_{Pr}. Hence t appears in a position with depth at most $d_{Pr} \cdot prvst(q'') + d_{Pr} = d_{Pr} \cdot prvst(q') \leq d_{Pr} \cdot prvst(q)$. This inequality is true since $prvst(q') = prvst(q'') + 1$, as q' is a direct successor of q'', and since q is a (not necessarily direct) successor of q', it follows that $prvst(q') \leq prvst(q)$.

Due to the same argument as above, extractions appearing in parallel can be treated independently.

The following lemma now establishes "Move Transfer" for the adversary: When two states are equivalent, an adversary move from one can be "transformed" into a move for the other, such that the follow-up states are equivalent—provided that honest principals perform the same moves (as they can due to Proposition 6.6). In the following lemma, note that every possible choice of q' leads to the same number d'.

Lemma 6.9. *Let q^1 and q^2 be non-final states in $\mathcal{C}_{Pr}$ such that $q^1 \equiv q^2$. Let $d' = \mathsf{eqdeg}(q') + 2 \cdot d_{Pr}$, where q' is a successor state of q^1 or q^2. Then for every adversary move $m^1_{\mathcal{A}}$ in q^1, there exists an adversary move $m^2_{\mathcal{A}}$ in q^2 such that*

$$terms(q^1) \circ m^1_{\mathcal{A}}[\mathcal{M}^1_{\mathcal{A}}/x] \sim_{d'} terms(q^2) \circ m^2_{\mathcal{A}}[\mathcal{M}^2_{\mathcal{A}}/x]$$

(where $\mathcal{M}^1_{\mathcal{A}}$ and $\mathcal{M}^2_{\mathcal{A}}$ are the sequences of messages received by the adversary in q^1 and q^2). The move $m^2_{\mathcal{A}}$ can be computed from the move $m^1_{\mathcal{A}}$ and the information that $\mathcal{A}$ has above q^1 and q^2 with information degree 3.

We note that although the formal proof of Lemma 6.9 below is quite long and technical, the construction itself is rather straight-forward: The move $m^2_{\mathcal{A}}$ is "almost" the same one as $m^1_{\mathcal{A}}$, with the following modifications:

1. If $m^1_{\mathcal{A}}$ uses terms that "extract" terms in q^1 that appear in positions with high depth, the references are changed to positions with "low" depth. This "rewriting" of the adversary move can be performed due to Proposition 6.8.
2. Since the main idea of the construction is that "we are only interested in terms down to a certain depth," we need to ensure that terms below this depth do not play a role. In order to ensure that positions appearing with "very high depth" in the adversary move have no effect, we simply "delete" these subterms—we replace them with new adversary nonces (not appearing previously anywhere). This implies that our construction does not introduce any "new equalities" into the system.

The number of cases needed to be considered (considering positions p_1 and p_2 referring either into the new message, components of the old state, etc) make the formal proof technical and tedious, however we stress that the above intuitive idea is the main ingredient to the proof.

Proof. Let $d = \mathsf{eqdeg}\left(q^1\right) = \mathsf{eqdeg}\left(q^2\right)$ (these values are identical since $q^1 \equiv q^2$). By definition, it then follows that $d = \mathsf{eqdeg}\left(\ell - \mathit{prvst}(q^1)\right)$. Also by definition, we know that $d' = \mathsf{eqdeg}\left(q'\right) + 2 \cdot d_{Pr} = \mathsf{eqdeg}\left(\ell - \mathit{prvst}(q')\right) + 2 \cdot d_{Pr}$. Since q' is a successor of q^1 or q^2, we have that $\mathit{prvst}(q') = \mathit{prvst}(q^1) + 1$. Let $s = \ell - \mathit{prvst}(q_1) - 1$. It then follows that $d = \mathsf{eqdeg}\left(s+1\right)$ and $d' = \mathsf{eqdeg}\left(s\right) + 2 \cdot d_{Pr}$. By definition of $\equiv$ we know that $q^1 \sim_d q_2$. From the above and the definition of $\mathsf{eqdeg}\left(.\right)$, it follows that

$$\begin{aligned} d &= 2\mathsf{eqdeg}\left(s\right) + 4d_{Pr} \\ &\quad + 2\mathit{mdagdpth}_{\mathcal{A}}(\mathsf{eqdeg}\left(s\right) + 2 \cdot d_{Pr}) + 2 \cdot \ell \cdot d_{Pr} \\ &= 2d' + 2\mathit{mdagdpth}_{\mathcal{A}}(d') + 2 \cdot \ell \cdot d_{Pr}. \end{aligned}$$

Let $m^1 = [[m^1_{\mathcal{A}}[\mathcal{M}^1_{\mathcal{A}}/x]]]$ be the resulting message sequence sent by the adversary, and let t_i denote the terms in that sequence, i.e., $m^1 = (t_1, \ldots, t_k)$. We construct a directed acyclic graph m^1_{DAG} with root *root* having k outgoing edges leading to trees representing the terms $t_1, \ldots, t_k$ (where k is the number of roles in the protocol). For a position $p = (i, \mathit{path})$, with $m^1_{\mathrm{DAG}} \to p$ we denote the vertex in m^1_{DAG} obtained when following the path $i \circ \mathit{path}$ from *root*, and with $m^1_{\mathrm{DAG}} \downarrow p$, we denote the subterm represented by $m_{\mathrm{DAG}} \to p$ (where the subterm represented by a vertex is interpreted in the canonical way). We use the same notation for the other DAGs appearing in the remainder of the proof. It follows that $m^1 \downarrow p = m^1_{\mathrm{DAG}} \downarrow p$ for all positions p. We say that positions p_1 and p_2 with depth at most d' are *equivalent* (written $p_1 \sim p_2$), if $m^1 \downarrow p_1 = m^1 \downarrow p_2$. We modify m^1_{DAG} as follows:

For each equivalence class, let p^0 be a representative, and for all $p' \sim p^0$, redirect all incoming edges of $m^1_{\mathrm{DAG}} \to p'$ to $m^1_{\mathrm{DAG}} \to p^0$.

The construction ensures that if $p_1 \sim p_2$, then $m^1_{\mathrm{DAG}} \to p_1 = m^1_{\mathrm{DAG}} \to p_2$. The terms represented by the involved positions remain invariant, i.e., for all positions p, we have $m^1 \downarrow p = m^1_{\mathrm{DAG}} \downarrow p$. In particular, the resulting graph is acyclic: A cycle would imply the existence of an infinite subterm that is not present in m^1. For a position $p = (i, path)$, let $dagdepth(p)$ be the length of a longest path from the root of the term representing t_i to $m^1_{\mathrm{DAG}} \to p$ (which may be longer than $path$).

We first establish a bound on the resulting depth for positions that originally have a depth of at most d', i.e., those positions for which we want to establish equivalence.

Fact 1 *If p is a position with $depth\,(p) \leq d'$, then $dagdepth(p) \leq mdagdpth_{\mathcal{A}}(d')$.*

Proof. (of Fact 1) Let $dagdepth(p, i)$ be the dagdepth of p after i redirection steps. We claim that $dagdepth(p, i) \leq 2^i \cdot (d' + 1)$, if $depth\,(p) \leq d'$. For $i = 0$, this is obvious, since $dagdepth(p, 0) = depth\,(p) \leq d'$. Note that on each path in m^1_{DAG}, at most one edge is redirected in each step. Let p be a position with $depth\,(p) \leq d'$ whose dagdepth changes in step i. Now let p^0 be the representative chosen in step i. Then there is a position p' with $depth\,(p') \leq d'$ and a path $path$ such that $m^1_{\mathrm{DAG}} \to (p' \circ path) = m^1_{\mathrm{DAG}} \to p$ after step $i - 1$, and the set of incoming edges of $m^1_{\mathrm{DAG}} \to p'$ changes in step i (either because p' is the representative p^0 and thus the vertex gets additional incoming edges, or the incoming edges of $m^1_{\mathrm{DAG}} \to p'$ get rerouted in this step). Let $path$ be a longest path such that $m^1_{\mathrm{DAG}} \to (p' \circ path) = m^1_{\mathrm{DAG}} \to p$ before step i. It follows that $|path| \leq dagdepth(p, i - 1)$.

After step i, $m^1_{\mathrm{DAG}} \to p' = m^1_{\mathrm{DAG}} \to p^0$. Since on each path, at most one edge is redirected in the step i, it follows that $path$ is still the longest path from p' to p in m^1_{DAG} after step i. Since we assumed that $dagdepth(p, i) \neq dagdepth(p, i-1)$, we know that the longest path from $root$ to $m^1_{\mathrm{DAG}} \to p$ after step i is one visiting p'. Hence $dagdepth(p, i) = dagdepth(p', i)+|path|$. We also know $dagdepth(p', i) = dagdepth(p'', i-1)$, where p'' is the position in the equivalence class of p^0 with the maximal dagdepth before step i. It follows that $dagdepth(p, i) = dagdepth(p', i)+ |path| = dagdepth(p'', i - 1) + |path| \leq dagdepth(p'', i - 1) + dagdepth(p, i - 1)$. Due to induction, since $depth\,(p'')\,, depth\,(p) \leq d'$, we have $dagdepth(p'', i - 1), dagdepth(p, i - 1) \leq 2^{i-1} \cdot (d' + 1)$, and hence $dagdepth(p, i) \leq 2 \cdot (2^{i-1} \cdot (d' + 1)) = 2^i \cdot (d' + 1)$ as claimed.

The number of steps in the construction is the number $\#e$ of equivalence classes. Since a pair of positions where one is a proper prefix of the other cannot be equivalent, $\#e$ is bounded by the number of positions with depth at most d' that are no prefixes of each other. This is the number of leaves in a tree at level d', where the root vertex has out-degree k, and the remaining vertices have an out-degree of at most the maximal arity of an operator from Σ^{t}. Hence if $depth\,(p) \leq d'$, then $dagdepth(p) = dagdepth(p, \#e) \leq (2^{\#e}) \cdot (d' + 1)$, which is exactly the definition of $mdagdpth_{\mathcal{A}}(d')$.

From m^1_{DAG}, we now obtain m_{DAG} as follows: The idea of the construction is to ensure that every term that appears in the message that is sent to the

adversary which contains a reference to a subterm of a term appearing in depth at most d' in q^1, is replaced with a marker which indicates the subterm that appears at this position. The message that the adversary needs to send in q^2 to obtain $\sim_{d'}$-equivalence is obtained by instantiating these markers with the corresponding subterms appearing in q^2.

1. *For all positions p^{DAG}, p, r and paths path such that $m^1_{\mathrm{DAG}}{\downarrow}p^{\mathrm{DAG}} = q^1{\downarrow}r$, depth $(r) \leq d'$, and $m^1_{\mathrm{DAG}} \to p = m^1_{\mathrm{DAG}} \to (p^{\mathrm{DAG}} \circ path)$, replace the vertex at $m_{\mathrm{DAG}} \to p$ with a vertex containing the marker $(r \to path)$. Remove all vertices from m_{DAG} that are not reachable from root anymore.*
2. *For all positions p with $dagdepth(p) > mdagdpth_{\mathcal{A}}(d')$, if $m_{\mathrm{DAG}} \to p$ still exists, insert a new adversary nonce into $m_{\mathrm{DAG}} \to p$, unless on all outgoing branches, one of the following is true: After $\ell \cdot d_{Pr}$ steps, the branch stops or contains a marker referring to an adversary-constructed term or an $\mathcal{A}$-accessible position in q^1. Replace all entries appearing in positions with dagdepth more than $mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$ with new adversary nonces.*

To make the construction well-determined, we demand that there is an injective function f such that in the i-th protocol step (i.e., if $prvst(q^1) = prvst(q^2) = i-1$), the term t is replaced with $f(t,i)$; without loss of generality we assume that the nonces in the image of f do not appear in the original adversary moves by using a unique prefix for the name of the newly introduced nonces that does not appear in the names of nonces in $\mathcal{C}_{Pr}$. By construction, the adversary can access the terms t that need to be replaced (since they do not appear below markers, they have been constructed by the adversary) and hence the appearing values $f(t,i)$ can be computed with the knowledge available to the adversary.

Note that $m^1_{\mathrm{DAG}} \to p_1 = m^1_{\mathrm{DAG}} \to p_2$ does not necessarily imply $m_{\mathrm{DAG}} \to p_1 = m_{\mathrm{DAG}} \to p_2$ (there might be a prefix p' of p_1 such that $m_{\mathrm{DAG}} \to p'$ contains a marker, then $m_{\mathrm{DAG}} \to p_1$ does not exist, while $m_{\mathrm{DAG}} \to p_2$ still does). In particular, the above can "fail" if for a prefix p' of p, $m_{\mathrm{DAG}} \to p'$ has already been overwritten with a marker. In this case, the "replace" operation does nothing. By construction, if $m_{\mathrm{DAG}} \to p$ contains a marker $(r \to path)$, then $|path| \leq dagdepth(p)$. and $m^1{\downarrow}p = q^1{\downarrow}(r \circ path)$.

Fact 2 *Let p be a position such that $m_{\mathrm{DAG}} \to p$ contains a marker $(r \to path)$. Then $|path| \leq dagdepth(p) \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$.*

Proof. (of Fact 2) Since $m_{\mathrm{DAG}} \to p$ contains the marker $(r \to path)$, there is no prefix p' of p such that $m_{\mathrm{DAG}} \to p'$ contains a newly introduced adversary nonce. In particular, this implies $dagdepth(p) \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$. Hence due to construction, $|path| \leq dagdepth(p) \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$.

Let m^2_{DAG} be the graph obtained from m_{DAG} by replacing every vertex containing a marker $(r \to path)$ with the term $q^2 \downarrow (r \circ path)$, and define $m^2 = \left(m^2_{\mathrm{DAG}}{\downarrow}1, \ldots, m^2_{\mathrm{DAG}}{\downarrow}k\right)$. In particular, if $m_{\mathrm{DAG}} \to p$ contains a marker $(r \to path)$, then $m^2 \downarrow p = q^2 \downarrow (r \circ path)$. In our construction, m^2 will be the

message sequence actually sent by the adversary as a consequence of the application of the move $m^2_{\mathcal{A}}$, i.e., $m^2 = m^2_{\mathcal{A}}[\mathcal{M}^2_{\mathcal{A}}/x]$. We first show that m^2 satisfies the required properties, and then prove that an adversary move $m^2_{\mathcal{A}}$ resulting in this message to be sent can be constructed given the adversary's information about q^1, q^2, and the adversary move $m_{\mathcal{A}}$. The following Fact 3 establishes the first condition of $\sim_{d'}$-equivalence. We prove a stronger statement than required by $\sim_{d'}$-equivalence since we will require it later in the proof.

Fact 3 *Let p be a position such that $depth\,(p) \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$ and there is no prefix of p that contains a new adversary nonce. Then $m^1(p) = m^2(p)$.*

Proof. (of Fact 3) First assume that there is a minimal prefix p_r of p such that $m_{\mathrm{DAG}} \to p_r$ contains a marker $(r \to path)$. Let $p = p_r \circ w$. From Fact 2, it follows that $|path| \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$, and from the construction, we know $depth\,(r) \leq d'$. It therefore follows that for $a \in \{1, 2\}$, we have that

$$\begin{aligned} m^a(p) = &\quad (m^a \downarrow p_r)(w) \\ &= (q^a \downarrow (r \circ path))(w) \\ &= \quad q^a(r \circ path \circ w)\,. \end{aligned}$$

We also know that $|w| \leq depth\,(p) \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$. It therefore follows that $depth\,(r \circ path \circ w) = depth\,(r) + |path| + |w| \leq d' + mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr} + mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr} \leq d$, and thus due to d-equivalence of q^1 and q^2, it follows that $m^1(p) = q^1(r \circ path \circ w) = q^2(r \circ path \circ w) = m^2(p)$ as required.

Now assume there is no prefix of p containing a marker. Since there is also no prefix of p containing a newly introduced adversary nonce, and it follows that $m^1(p) = m_{\mathrm{DAG}}(p) = m^2(p)$ as required.

We now prove $s_1 := terms(q^1) \circ m^1 \sim_{d'} terms(q^2) \circ m^2 =: s_2$. Let p be a position with $depth\,(p) \leq d'$. We show that $s_1(p) = s_2(p)$. If p is a position referring into q^1/q^2, the claim holds since $q^1 \sim_d q^2$ and $d' \leq d$. If p refers into m^1/m^2, the equality follows from Fact 3 which we can apply since $d' \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$, and since $depth\,(p) \leq d'$, we know that due to Fact 1, $dagdepth(p) \leq mdagdpth_{\mathcal{A}}(d')$, and thus no prefix of p contains a new adversary nonce.

Now assume $s^1 \downarrow p_1 = s^1 \downarrow p_2$ for positions p_1, p_2 with $depth\,(p_1)$, $depth\,(p_2) \leq d'$. Again, when p_b is a position of s^a referring into q^a (or m^a), we write $q^a \downarrow p_b$ (or $m^a \downarrow p_b$) for the term contained in q^a (or m^a) addressed by p_b. In the case that both positions refer into q^1/q^2, the claim follows: Since $d' \leq d$ and $q^1 \sim_d q^2$, we have that $q^1 \downarrow p_1 = q^1 \downarrow p_2$ implies $q^2 \downarrow p_1 = q^2 \downarrow p_2$ as required. We make a case distinction for the non-trivial cases.

Assume both p_1 and p_2 refer to a term from m^1/m^2. By construction, since $p_1 \sim p_2$, $m^1_{\mathrm{DAG}} \to p_1 = m^1_{\mathrm{DAG}} \to p_2$. We need to show that $m^2_{\mathrm{DAG}} \downarrow p_1 = m^2_{\mathrm{DAG}} \downarrow p_2$. Obviously, if $m_{\mathrm{DAG}} \to p_1 = m_{\mathrm{DAG}} \to p_2$, this follows trivially. Hence assume this is not the case. In particular, a prefix of one of these positions has been modified in the construction of m_{DAG} from m^1_{DAG}. Since for $b \in \{1, 2\}$

we have $depth\,(p_b) \leq d'$, Fact 1 implies that $dagdepth(p_b) \leq mdagdpth_{\mathcal{A}}(d')$. In particular, no new adversary-nonces have been written into a prefix of p_b. Therefore, we assume without loss of generality that there is a prefix p of p_1 such that $m_{\mathrm{DAG}} \to p$ contains a marker. Since in m^1_{DAG}, there is a path from $m^1_{\mathrm{DAG}} \to p$ to $m^1_{\mathrm{DAG}} \to p_1 = m^1_{\mathrm{DAG}} \to p_2$, a marker was also written into a prefix of $m^1_{\mathrm{DAG}} \to p_2$, unless there already was a prefix of p_2 containing a marker. Thus both p_1 and p_2 have prefixes containing markers, i.e., for $i = 1, 2$, there are positions p'_i which still exist in m_{DAG} and $p_i = p'_i \circ w_i$, where $depth\,(p'_i), |w_i| \leq depth\,(p_i) \leq d'$, and $m_{\mathrm{DAG}} \to p'_i$ contains a marker $(r_i \to path_i)$, where $depth\,(r_i) \leq d'$. Due to Fact 2, $|path_i| \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$. It follows that

$$\begin{aligned} m^a{\downarrow}p_i &= m^a{\downarrow}(p'_i \circ w_i) \\ &= (m^a{\downarrow}p'_i){\downarrow}w_i \\ &= (q^a{\downarrow}(r_i \circ path_i)){\downarrow}w_i \\ &= q^a{\downarrow}(r_i \circ path_i \circ w_i). \end{aligned}$$

Hence $q^1{\downarrow}(r_1 \circ path_1 \circ w_1) = m^1{\downarrow}p_1 = m^1{\downarrow}p_2 = q^1{\downarrow}(r_2 \circ path_2 \circ w_2)$. Note that $depth\,(r_i \circ path_i \circ w_i) = depth\,(r_i) + |path_i| + |w_i| \leq d' + mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr} + d' \leq d$. From $q^1 \sim_d q^2$ and the above it follows that $q^2{\downarrow}(r_1 \circ path_1 \circ w_1) = q^2 \downarrow (r_2 \circ path_2 \circ w_2)$, and therefore $m^2 \downarrow p_1 = q^2 \downarrow (r_1 \circ path_1 \circ w_1) = q^2 \downarrow (r_2 \circ path_2 \circ w_2) = m^2{\downarrow}p_2$ as required.

Assume p_1 refers to a term from m^1/m^2, and p_2 to a term from q^1/q^2. Since $m^1{\downarrow}p_1 = q^1{\downarrow}p_2$, and $depth\,(p_2) \leq d'$, by the construction of m_{DAG}, there is a prefix p_r of p_1 such that $m_{\mathrm{DAG}} \to p_r$ contains a marker $(r \to path)$ for some path $path$ and position r with $depth\,(r) \leq d'$, and $p_1 = p_r \circ w$ for some w with $|w| \leq depth\,(p_1) \leq d'$. Due to Fact 2, $|path| \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$. Note that

$$\begin{aligned} m^a{\downarrow}p_1 = m^a_{\mathrm{DAG}}{\downarrow}(p_r \circ w) \quad &= (m^a_{\mathrm{DAG}}{\downarrow}p_r){\downarrow}w \\ = (q^a{\downarrow}(r \circ path)){\downarrow}w &= q^a{\downarrow}(r \circ path \circ w). \end{aligned}$$

It follows that $q^1{\downarrow}p_2 = m^1{\downarrow}p_1 = q^1{\downarrow}(r \circ path \circ w)$. Since $depth\,(p_2) \leq d' \leq d$, and $depth\,(r \circ path \circ w) = depth\,(r) + |path| + |w| \leq d' + mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr} + d' \leq d$, the prerequisite $q^1 \sim_d q^2$ implies $q^2{\downarrow}p_2 = q^2{\downarrow}(r \circ path \circ w)$, and hence we conclude that $q^2{\downarrow}p_2 = q^2{\downarrow}(r \circ path \circ w) = m^2{\downarrow}p_1$, as required.

This completes the proof showing that for positions p_1, p_2 with depth at most d', if $s^1{\downarrow}p_1 = s^1{\downarrow}p_2$, then also $s^2{\downarrow}p_1 = s^2{\downarrow}p_2$. We now show the other direction: If $s^2{\downarrow}p_1 = s^2{\downarrow}p_2$, then also $s^1{\downarrow}p_1 = s^1{\downarrow}p_2$. We use an analogous case distinction as in the proof of the previous direction. Again the claim is trivial if both positions refer into q^1/q^2.

Assume that p_1 refers to a term from m^1/m^2, and p_2 to a term from q^1/q^2. With notation as earlier, then $m^2{\downarrow}p_1 = q^2{\downarrow}p_2$. We show $m^1{\downarrow}p_1 = q^1{\downarrow}p_2$, where we only require that $depth\,(p_1) \leq d'$, and $depth\,(p_2) \leq 2 \cdot d' + mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$ (we will use this stronger result in the sequel). Note that no extension of p_1 can contain a newly introduced adversary nonce: Otherwise, equality with a subterm

from q^2 would not hold. Assume $m^1{\downarrow}p_1 \neq q^1{\downarrow}p_2$, and let w be a minimal path such that $m^1(p_1 \circ w) \neq q^1(p_2 \circ w)$.

We first show that there is no prefix of $p_1 \circ w$ that refers to a marker in m_{DAG}. Assume indirectly that there is a prefix p' of $p_1 \circ w$ such that $p' \circ w' = p_1 \circ w$ for some path w, and $m_{\text{DAG}} \to p'$ contains the marker $(r \to \mathit{path})$. From Fact 2, it follows that $|\mathit{path}| \leq \mathit{mdagdpth}_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$. Since $p' \circ w' = p_1 \circ w$, one of $\{p', p_1\}$ must be a prefix of the other. We first assume p' is a prefix of p_1, i.e., there is some w'' such that $p_1 = p' \circ w''$. Then $|w''| \leq \mathit{depth}\,(p_1) \leq d'$. It follows that

$$\begin{aligned} m^a{\downarrow}p_1 = &\quad m^a{\downarrow}(p' \circ w'') \\ = &\quad (m^a{\downarrow}p'){\downarrow}w'' \\ = &\ (q^a{\downarrow}(r \circ \mathit{path})){\downarrow}w'' \\ = &\ q^a{\downarrow}(r \circ \mathit{path} \circ w''). \end{aligned}$$

In particular, $q^2 \downarrow p_2 = m^2 \downarrow p_1 = q^2 \downarrow (r \circ \mathit{path} \circ w'')$. Since $\mathit{depth}\,(p_2) \leq 2 \cdot d' + \mathit{mdagdpth}_{\mathcal{A}}(d') + \ell \cdot d_{Pr} \leq d$, and $\mathit{depth}\,(r \circ \mathit{path} \circ w'') = \mathit{depth}\,(r) + |\mathit{path}| + |w''| \leq d' + \mathit{mdagdpth}_{\mathcal{A}}(d') + \ell \cdot d_{Pr} + d' \leq d$, the d-equivalence of q^1 and q^2 implies that $q^1{\downarrow}p_2 = q^1{\downarrow}(r \circ \mathit{path} \circ w'')$. Hence we obtain $m^1(p_1 \circ w) = (m^1{\downarrow}p_1)(w) = (q^1{\downarrow}(r \circ \mathit{path} \circ w''))(w) = (q^1{\downarrow}p_2)(w) = q^1(p_2 \circ w)$, a contradiction to the choice of w.

Now assume p_1 is a prefix of p', and let $p' = p_1 \circ w''$ for some path w''. Since p' contains a marker, due to Fact 2, it follows that $\mathit{depth}\,(p') \leq \mathit{dagdepth}(p') \leq \mathit{mdagdpth}_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$. We have $p_1 \circ w = p' \circ w' = p_1 \circ w'' \circ w'$, and hence $w = w'' \circ w'$. We know that $q^2{\downarrow}(p_2 \circ w'') = m^2{\downarrow}(p_1 \circ w'') = m^2{\downarrow}p' = q^2{\downarrow}(r \circ \mathit{path})$. Note that $\mathit{depth}\,(p_2 \circ w'') = \mathit{depth}\,(p_2) + |w''| \leq 2 \cdot d' + \mathit{mdagdpth}_{\mathcal{A}}(d') + \ell \cdot d_{Pr} + \mathit{depth}\,(p') \leq 2 \cdot d' + 2 \cdot \mathit{mdagdpth}_{\mathcal{A}}(d') + \ell \cdot d_{Pr} + \ell \cdot d_{Pr} \leq d$, and $\mathit{depth}\,(r \circ \mathit{path}) = \mathit{depth}\,(r) + |\mathit{path}| \leq d' + \mathit{mdagdpth}_{\mathcal{A}}(d') + \ell \cdot d_{Pr} \leq d$. Therefore the above equality and the d-equivalence of q^1 and q^2 implies $q^1 \downarrow (p_2 \circ w'') = q^1 \downarrow (r \circ \mathit{path})$. It therefore follows that $m^1 \downarrow (p_1 \circ w) = m^1 \downarrow (p' \circ w') = (m^1 \downarrow p') \downarrow w' = (q^1 \downarrow (r \circ \mathit{path})) \downarrow w' = (q^1 \downarrow (p_2 \circ w'')) \downarrow w' = q^1 \downarrow (p_2 \circ w'' \circ w') = q^1 \downarrow (p_2 \circ w)$, a contradiction.

Since we obtained a contradiction in both cases, we know that there is no prefix of $p_1 \circ w$ referring to a marker.

Since there is also no extension of p_1 referring to a new adversary nonce, it follows that $m^a(p_1 \circ w) = m_{\text{DAG}}(p_1 \circ w)$. Due to minimality of w, we know that $m^1(p_1 \circ w) \neq \texttt{error}$, i.e., the position exists. Since no prefix of $p_1 \circ w$ contains a new adversary nonce or a marker, we know that $|w| \leq \mathit{depth}\,(p_1 \circ w) \leq \mathit{dagdepth}(p_1 \circ w) \leq \mathit{mdagdpth}_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$. Hence $\mathit{depth}\,(p_2 \circ w) = \mathit{depth}\,(p_2) + |w| \leq 2d' + \mathit{mdagdpth}_{\mathcal{A}}(d') + \ell \cdot d_{Pr} + \mathit{mdagdpth}_{\mathcal{A}}d' + \ell \cdot d_{Pr} \leq d$. It therefore follows from the d-equivalence of q^1 and q^2 that $m^1(p_1 \circ w) = m_{\text{DAG}}(p_1 \circ w) = m^2(p_1 \circ w) = q^2(p_2 \circ w) = q^1(p_2 \circ w)$, again a contradiction.

Assume both p_1 and p_2 refer to a term from m^1/m^2. If $m_{\text{DAG}} \to p_1 = m_{\text{DAG}} \to p_2$, then by construction, $m^1{\downarrow}p_1 = m^1{\downarrow}p_2$ as required. Hence assume this is not the case. First assume there is a prefix p_r of p_1 such that p_r contains a marker

$(r \to path)$, i.e., $p_1 = p_r \circ w$ for some w with $|w| \leq depth(p_1) \leq d'$. Due to Fact 2, $|path| \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$. It follows that

$$\begin{aligned} m^a{\downarrow}p_1 = &\quad m^a{\downarrow}(p_r \circ w) \\ = &\quad (m^a{\downarrow}p_r){\downarrow}w \\ = &\ (q^a{\downarrow}(r \circ path)){\downarrow}w \\ = &\ q^a{\downarrow}(r \circ path \circ w), \end{aligned}$$

and hence $m^2{\downarrow}p_2 = m^2{\downarrow}p_1 = q^2{\downarrow}(r \circ path \circ w)$.

Since $depth(r \circ path \circ w) = depth(r) + |path| + |w| \leq d' + mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr} + d'$, and $depth(p_2) \leq d'$, the above case (where we only required the position referring into q^1/q^2 to have a depth bounded by $2 \cdot d' + mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$) implies $q^1{\downarrow}(r \circ path \circ w) = m^1{\downarrow}p_2$, and thus $m^1{\downarrow}p_2 = q^1{\downarrow}(r \circ path \circ w) = m^1{\downarrow}p_1$ as required. The case that a prefix of p_2 contains a marker is symmetric.

Hence assume no prefix of p_1 or p_2 leads to a position in m_{DAG} containing a marker. Assume there is a minimal path w such that $m^1(p_1 \circ w) \neq m^1(p_2 \circ w)$. First assume that there is a prefix p_r of $p_1 \circ w$ or of $p_2 \circ w$ containing a marker, without loss of generality assume that $p_1 \circ w = p_r \circ w'$, and $m_{\text{DAG}} \to p_r$ contains the marker $(r \to path)$. From Fact 2, we know that $dagdepth(p_r), |path| \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$. Since no prefix of p_1 contains a marker, and there is a common extension of p_1 and p_r, p_r must be an extension of p_1, i.e., there is a path w'' such that $p_r = p_1 \circ w''$ for some w'' with $|w''| \leq depth(p_r) \leq dagdepth(p_r) \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$. It thus follows that $p_1 \circ w = p_r \circ w' = p_1 \circ w'' \circ w'$, i.e., $w = w'' \circ w'$. Obviously, we have $m_{\text{DAG}} \to (p_1 \circ w) \neq m_{\text{DAG}} \to (p_2 \circ w)$, and there is no prefix of $p_2 \circ w$ such that m^2 at the position of this prefix contains a newly introduced adversary nonce (otherwise, equality of $m^2{\downarrow}(p_2 \circ w)$ with a subterm of q^2 would not hold, but we know that $m_2{\downarrow}(p_2 \circ w) = m_2{\downarrow}(p_1 \circ w) = m_2{\downarrow}(p_r \circ w)$, which is a term appearing in q^2).

We now show that there is also no prefix of $p_2 \circ w$ that contains a marker (where we are still considering the case that such a prefix does exist for $p_1 \circ w$). Assume that this is the case, i.e. (since we know that no prefix of p_2 itself contains a marker), there is some prefix w''' of w such that $p_2 \circ w''' = p'_r$ and $m_{\text{DAG}} \to p'_r$ contains a marker $(r' \to path')$, where $depth(r') \leq d'$, and we know due to Fact 2 that $|path'| \leq mdagdpth_{\mathcal{A}}(d') + \ell d_{Pr}$. Since w'' is also a prefix of w, one of $\{w'', w'''\}$ must be a prefix of the other. Without loss of generality assume that $w'' = w'''' \circ w'''$ for some w''' with $|w'''| \leq |w''| \leq mdagdpth_{\mathcal{A}}(d') + \ell d_{Pr}$. It follows that $w = \underbrace{w'''' \circ w'''}_{=w''} \circ w'$, and for $a \in \{1, 2\}$, we have

$$m^a{\downarrow}(p_1 \circ w'') = m^a{\downarrow}p_r = q^a{\downarrow}(r \circ path),$$

and

$$\begin{aligned} m^a{\downarrow}(p_2 \circ w'') = m^a{\downarrow}(p_2 \circ w'''' \circ w''') &= (m^a{\downarrow}p'_r){\downarrow}w''' \\ &= q^a{\downarrow}(r' \circ path' \circ w'''). \end{aligned}$$

Since $m^2{\downarrow}p_1 = m^2{\downarrow}p_2$, it follows that $q^2{\downarrow}(r \circ path) = m^2{\downarrow}(p_2 \circ w'') = m^2{\downarrow}(p_2 \circ w'') = q^2{\downarrow}(r' \circ path' \circ w''')$. Since $depth(r \circ path) = depth(r) + |path| \leq$

$d' + mdagdpth_{\mathcal{A}}(d') + \ell d_{Pr} \leq d$, and $depth\left(r' \circ path' \circ w'''\right) = depth\,(r') + \left|path'\right| + |w'''| \leq d' + mdagdpth_{\mathcal{A}}(d') + \ell d_{Pr} + mdagdpth_{\mathcal{A}}(d') + \ell d_{Pr} \leq d$, and $q^1 \sim_d q^2$, it follows that $q^1 \downarrow (r \circ path) = q^1 \downarrow (r' \circ path' \circ w''')$, and therefore $m^1(p_1 \circ w) = m^1(p_1 \circ w'' \circ w') = m^1(p_1 \circ w'')(w') = q^1\left(r' \circ path' \circ w'''\right)(w') = q^1(r \circ path)(w') = m^1{\downarrow}(p_1 \circ w'')(w') = m^1(p_1 \circ w'' \circ w') = m^1(p_1 \circ w)$, which is a contradiction to the choice of w. We therefore know that no prefix of $p_2 \circ w$ can contain a marker. Since we also know, due to the above, that no prefix of $p_2 \circ w$ contains a new adversary nonce, the construction of m_{DAG} implies that $depth\,(p_2 \circ w) \leq dagdepth(p_2 \circ w) \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$, and hence in particular, $|w'| \leq |w|\, mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$. Hence $depth\,(r \circ path \circ w') = depth\,(r) + |path| + |w'| \leq d' + mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr} + mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr} \leq d$. Since due to the above, we know that $depth\,(p_2 \circ w) \leq mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$ and no prefix of this position contains a new adversary nonce, we can apply Fact 3 and obtain $m^1(p_2 \circ w) = m^2(p_2 \circ w)$. Since $q^1 \sim_d q^2$, it follows that $m^1(p_2 \circ w) = m^2(p_2 \circ w) = m^2(p_1 \circ w) = m^2(p_r \circ w') = q^2(r \circ path \circ w') = q^1(r \circ path \circ w') = m^1(p_r \circ w') = m^1(p_1 \circ w)$, a contradiction. Therefore, no prefix of $p_1 \circ w$ or $p_2 \circ w$ contains a marker.

Now assume that a prefix of $p_1 \circ w$ contains a new adversary nonce. Since $depth\,(p_1) \leq d'$, Fact 1 implies that $dagdepth(p_1) \leq mdagdpth_{\mathcal{A}}(d')$, and hence no prefix of p_1 can contain a new adversary nonce.

Therefore, $m_{\mathrm{DAG}} \to (p_1 \circ w')$ contains an adversary nonce introduced for a term t and for a paths w' such that $w = w' \circ w''$ for some w''. Since $m^2{\downarrow}p_1 = m^2{\downarrow}p_2$, it follows that $m_{\mathrm{DAG}} \to (p_2 \circ w')$ contains the same new adversary nonce introduced for the same term t. By construction, this implies that $m^1{\downarrow}(p_1 \circ w')$ and $m^1{\downarrow}(p_2 \circ w')$ both contain the same term t, and thus $m^1{\downarrow}(p_1 \circ w) = (m^1{\downarrow}(p_1 \circ w')) \downarrow w'' = t \downarrow w'' = (m^1 \downarrow (p_2 \circ w')) \downarrow w'' = m^1 \downarrow (p_2 \circ w)$, which is a contradiction to the choice of w. Hence, no prefix of $p_1 \circ w$ contains a newly introduced adversary nonce, analogously the same is true for p_2.

Therefore, no prefix if $p_1 \circ w$ or $p_2 \circ w$ points to a marker or a new adversary nonce, and it follows that $m^1(p_1 \circ w) = m_{\mathrm{DAG}}((p_1 \circ w)) = m^2(p_1 \circ w) = m^2(p_2 \circ w) = m_{\mathrm{DAG}}(p_2 \circ w) = m^1(p_2 \circ w)$, a contradiction. Hence $m^1 \downarrow p_1 = m^2{\downarrow}p_2$ as claimed.

We now show that there is an adversary move $m^2_{\mathcal{A}}$ that results in the message m^2 being sent, and that the adversary can compute this move from the original move $m^1_{\mathcal{A}}$ and the knowledge available to him in the states q^1 and q^2 in information degree 3 (obviously the result for information degrees 1 and 2 follows). Note that by definition of $\sim_d$, the same set of identities C is corrupted in q_1 and q_2. In particular, the set of moves available to the adversary is the same in q^1 and q^2.

The message m^2 that the adversary has to send is obtained from m^1 (which is available to the adversary, as it is computable from $m^1_{\mathcal{A}}$ and the information available in q^1) by replacing every position p such that $m_{\mathrm{DAG}} \to p$ contains

a marker $(r \to path)$ with the term $q^2 \downarrow (r \circ path)$, and afterwards replacing remaining terms at a certain depth with new adversary nonces.

In the following, we show how the adversary can perform these replacement operations for markers $(r \to path)$ such that the position $r \circ path$ is one that the adversary can access, i.e., a term computed by a principal that the adversary accesses, or a subterm of a message sent by the adversary previously in the protocol run which the adversary constructed himself. Afterwards, we show that performing the replacements for these cases also ensures that positions with markers not satisfying these conditions contain the correct terms. Hence, let p be a position such that $m_{\mathrm{DAG}} \to p$ contains a marker $(r \to path)$, and there is no proper prefix p' of p such that $m_{\mathrm{DAG}} \to p'$ contains a marker, and the position $r \circ path$ is either adversary-constructed, or adversary-accessible. We consider these two cases separately. Note that for every position p such that $m_{\mathrm{DAG}} \to p$ contains a marker, the term $m^1 \downarrow p$ can be computed by the adversary—otherwise, a prefix of that position already must contain a marker.

1. Assume that $q^1 \downarrow (r \circ path)$ is a term computed by the adversary in the past (i.e., there is a subterm of a past adversary move that results in this term). In this case, the term obviously can be constructed by using the same subterm. To recognize this case, the adversary merely has to compare every subterm of m^1 to every term sent by the adversary previously in the protocol run leading up to q^1, and substitute the corresponding term sent in the protocol run leading to q^2. Note that by construction of the protocol model, the adversary has access to all terms that he sent to principals in the protocol run (these are stored in the sequence $\mathcal{M}_{\mathcal{A}}$).
2. Assume that $q^1 \downarrow (r \circ path)$ is a term that the adversary cannot compute himself, i.e., a term not of the form mentioned in the statement of Proposition 6.8. Since the adversary can construct $q^1 \downarrow (r \circ path)$ in q^1, due to Proposition 6.8 there is an $\mathcal{A}$-accessible position p' containing this term in q^1. Since $depth\,(r \circ path) \leq d' + mdagdpth_{\mathcal{A}}(d') + \ell \cdot d_{Pr} \leq d$, the d-equivalence of q^1 and q^2 and $q^1 \downarrow (r \circ path) = q^1 \downarrow p'$ and $depth\,(p') \leq \ell \cdot d_{Pr}$ imply that $q^2 \downarrow (r \circ path) = q^2 \downarrow p'$. Due to Proposition 6.7, p' is $\mathcal{A}$-accessible in q^2, and hence $q^2 \downarrow (r \circ path)$ is constructable by the adversary. To recognize this case and determine the correct position p', the adversary only has to compare every subterm of the message m^1 to all terms $q^1 \downarrow p'$ for $\mathcal{A}$-accessible positions p'. The adversary is able to do that by performing a complete search over all terms t_e having extraction depth of at most $d_{Pr} \cdot prvst(q^1)$. Note that a finite bound for the number of terms to be considered can be established, however this is not necessary for the proof, as we only need to show that the resulting adversary move depends only on the available information—the notion of bisimulation only requires the move transfer functions to exist, without further requirements of computability or complexity. Due to Proposition 6.7, the adversary can use the same term to extract the corresponding subterm as he used in $m^1_{\mathcal{A}}$.

We now show that these operations also ensure that for the remaining markers, the corresponding positions contain the correct terms. Hence assume that

$q^1 \downarrow (r \circ \mathit{path})$ is a term that the adversary computed himself, but that has (also) been computed by a principal, i.e., in the message m^1, the term from $q^1 \downarrow (r \circ \mathit{path})$ is (partially) recreated and not accessed. Since terms computed by principals alone (i.e., without adversary input) have depth at most $\ell \cdot d_{Pr}$, we know that on every branch starting from p, after at most $\ell \cdot d_{Pr}$ steps, there is a marker referring to one of the cases treated above (or the end of the term), and this marker has been replaced with the corresponding subterm of q^2. Also, by construction no prefix of the position containing these markers has been replaced with a new adversary nonce. For the positions above these markers, we know that, since $m_{\mathcal{A}}$ contains a copy of the entries of the subterm $q^1 \downarrow (r \circ \mathit{path})$ up to depth $\ell \cdot d_{Pr}$, the d-equivalence of q^1 and q^2 implies that these terms are also a copy of the entries of $q^2 \downarrow (r \circ \mathit{path})$ up to depth $\ell \cdot d_{Pr}$: Since $\mathit{depth}\,(r) \leq d'$ and $|\mathit{path}| \leq \mathit{mdagdpth}_{\mathcal{A}}(d') + \ell \cdot d_{Pr}$, the depth of the occurring positions is bound by $d' + \mathit{mdagdpth}_{\mathcal{A}}(d') + \ell \cdot d_{Pr} + \ell \cdot d_{Pr} \leq d$. Therefore, the occurring depths are bound by d, thus equality of the relevant positions of q^1 and q^2 holds.

Therefore, the above two steps already ensure that these positions contain the correct terms.

It remains to show that the adversary can determine the set of positions in which new nonces are to be introduced given the available knowledge. This is true since the depth in which new nonces are introduced depends only on the protocol and the question whether terms appearing in that depth are identical to terms that the adversary can access. Finally, the adversary can compute $\mathit{dagdepth}(p)$ for every position p where a new nonce has to be introduced, since the involved positions do not appear below markers, and therefore they and all of their prefixes are accessible by the adversary, who can therefore compute the first step of the transformation which reroutes edges in m^1_{DAG} for the positions that do not appear below markers.

For a more efficient construction, it would be desirable to replace the function $\mathsf{eqdeg}\,()$ with one that grows more slowly in the number of steps of the protocol. However, in the above proof, note that there does not seem to be a straight-forward way to significantly lower the requirements on $d = \mathsf{eqdeg}\,(q_1)$ if we want to show that d'-equivalent moves always exist. We illustrate that the current proof approach needs to require at least $\mathit{mdagdpth}_{\mathcal{A}}(d')$-equivalence of the states q^1 and q^2 with an example: Assume that there are a position r, positions $p_1, \ldots, p_n$, and paths $w_1, \ldots, w_{n-1}$ such that

- $\mathit{depth}\,(r) = d'$,
- $\mathit{depth}\,(p_i) = 0$ for all i,
- $|w_i| = d'$ for all i,
- $m^1 \downarrow p_1 = q^1 \downarrow r$,
- $m^1 \downarrow p_{i+1} = m^1 \downarrow (p_i \circ w_i)$.

Since all stated equalities concern positions with depth at most d', the same equalities must hold in m^2/q^2. Now note that $m^a \downarrow p_n = m^a \downarrow p_1 \circ w_1 \circ \cdots \circ w_{n-1} = q^a \downarrow r \circ w_1 \cdots \circ w_{n-1}$. Since we want that $m^1(p_n) = m^2(p_n)$, it follows that $q^1(r \circ w_1 \cdots \circ w_{n-1}) = q^2(r \circ w_1 \cdots \circ w_{n-1})$. The depth of this position can only

be restricted by showing a bound on the number of elements in this "chain" as done in the proof above. However, a better bound than simply the number of inequivalent positions can probably be shown: In the situation described above, we compare positions with different depth (since $depth\,(p_i) = 0$, and $depth\,(p_i \circ w_i) = d'$). The situation does not arise when all involved positions have the maximal depth d', which was used in the proof to obtain the bound on the number of equivalence classes. Hence a finer analysis will probably result in a better bound, and thus a lower requirement for d (i.e., a slower growing function $\mathsf{eqdeg}\,()$). However, for realistic applications, the involved strategies are usually much simpler. Therefore, the decidability procedure can be optimized for each security property separately when applying it to real-world problems.. Hence we prove the bounds as stated in the proof, and leave the proof itself relatively simple.

6.5 The strategy representation of a protocol

We now define ${\mathcal{C}_{Pr}}/{\equiv}$, which as mentioned serves as a finite representation of $\mathcal{C}_{Pr}$ that contains all of the latter's strategic and epistemic properties. ${\mathcal{C}_{Pr}}/{\equiv}$ is essentially constructed by allowing the adversary to use only the moves that result in applying the construction of Lemma 6.9. Note that by construction, every term appearing in a state in ${\mathcal{C}_{Pr}}/{\equiv}$ has depth limited by a constant: The depth of terms which the adversary may introduce is limited by the construction of Lemma 6.9, and the honest principals only introduce terms of limited depth by construction. Hence ${\mathcal{C}_{Pr}}/{\equiv}$ is infinite, but only because of an infinite number of adversary nonces that may be used. Since there are only finitely many positions in which the adversary can introduce new nonces, we can without generality assume that the adversary only uses finitely many nonces. We therefore also use ${\mathcal{C}_{Pr}}/{\equiv}$ as name of the finite representation.

The results on move transfer for honest principals and the adversary show that $\mathcal{C}_{Pr}$ and ${\mathcal{C}_{Pr}}/{\equiv}$ are "strategically" equivalent. To obtain our bisimulation and result, we also require that they are epistemically equivalent, i.e., that principals can distinguish terms in a state of $\mathcal{C}_{Pr}$ if and only if they can do this in ${\mathcal{C}_{Pr}}/{\equiv}$. However, in ${\mathcal{C}_{Pr}}/{\equiv}$ as defined above, principals have "more information" than they have in $\mathcal{C}_{Pr}$: In the construction of Lemma 6.9, when subterms are replaced with new adversary nonces, a function f is used that replaces a term t with the nonce $f(t, i)$, where i is the number of the protocol step in which the term was used. Therefore, the terms in ${\mathcal{C}_{Pr}}/{\equiv}$ carry additional information about the protocol step in which a term was sent.

To "remove" this information from ${\mathcal{C}_{Pr}}/{\equiv}$, we simply add, to all of the epistemic equivalence relations for all players, each pair of states q_1 and q_2 such that when we define q_1' and q_2' as resulting from q_1 and q_2 by identifying all nonces of the form $f(t, i)$ for the same term t, the states q_1' and q_2' are indistinguishable with respect to the original equivalence relation. Obviously, these relations can be computed from the original ones. From now on, we denote with ${\mathcal{C}_{Pr}}/{\equiv}$ the thus-modified structure. We call ${\mathcal{C}_{Pr}}/{\equiv}$ the *strategy representation* of $\mathcal{C}_{Pr}$. We observe:

Theorem 6.10. *There is an algorithm which on input Pr constructs* $\mathcal{C}_{Pr}/\equiv$.

Proof. The algorithm constructs the structure in the straight-forward way. By construction, the depth of the terms allowed to be sent by the adversary in $\mathcal{C}_{Pr}/\equiv$ is restricted. Therefore, in each reachable state the adversary only has finitely many moves, and these can be enumerated in the obvious way. Honest principals only have finitely many moves and finitely many probabilistic choices by construction. For convergent subterm theories, the normal form of each term can be computed, therefore, for each state q and total move c, the possible successor states and their probabilities can be computed. In [AC06], it was proven that static equivalence is decidable for convergent subterm theories. This allows the algorithm to compute the pairs of indistinguishable states for each principal. The additional indistinguishabilities that we introduce can be computed in the straight-forward way.

6.6 Putting it all together: Proof of Strategy Simulation

We now show that $\equiv$ induces a probabilistic uniform strong alternating simulation in *both* directions, i.e., from $\mathcal{C}_{Pr}$ to $\mathcal{C}_{Pr}/\equiv$ and vice versa. In the following, let Q_1 and Q_2 be the sets of states of $\mathcal{C}_{Pr}/\equiv$ and $\mathcal{C}_{Pr}$, respectively. Let $Z \subseteq Q_1 \times Q_2$ be the relation defined as $(q_1, q_2) \in Z$ if and only if q_1 is the state obtained from q_2 as follows: Let λ_2 be the protocol run that reaches q_2. Then let λ_1 be obtained from λ_2 by exchanging each adversary move by the one obtained from the construction in Lemma 6.9, and letting the honest principals perform the same moves and random choices. Note that by Lemmas 6.5 and 6.9, it follows that $q_1 \equiv q_2$. On the other hand, in the following let $=_{inj}$ denote the relation containing two states $q_1 \in Q_1$ and $q_2 \in Q_2$ if they identical except that the adversary nonces in q_1 have the prefix introduced by the construction in Lemma 6.9. Hence, seen as a simulation from $\mathcal{C}_{Pr}$ to $\mathcal{C}_{Pr}/\equiv$, the relation $=_{inj}$ corresponds to the injection function *inj* from $\mathcal{C}_{Pr}/\equiv$ to $\mathcal{C}_{Pr}$, which strips off these prefixes. Note that the function *inj* is the relation $(=_{inj})^{-1}$.

Theorem 6.11. *The pair* $(Z, =_{inj})$ *is a probabilistic bisimulation between* $\mathcal{C}_{Pr}/\equiv$ *and* $\mathcal{C}_{Pr}$.

Proof. Obviously, $Z^{-1} \circ inj$ and $inj \circ Z^{-1}$ are idempotent, since both concatenations represent projection to the representative in $\mathcal{C}_{Pr}/\equiv$ with introduction or removal of nonce name prefixes. Hence it remains to show that each of the relations Z and $=_{inj}$ is a probabilistic uniform strong alternating simulation. We first treat the case $=_{inj}$, in this case the function Z^{-1} from the definition of a probabilistic uniform strong alternating simulation is *not* the converse of the relation Z introduced above, but is the injection function *inj* mentioned above. Propositional equivalence is trivial, the move properties follows using the identity as move transfer functions (again, with consistent renaming of adversary nonces), this function trivially is uniform, hence move uniformity is satisfied. Recall that due to Proposition 6.6, honest principals have the same available moves in equivalent states. Uniformity is trivial as well: Clearly, if $q_2 \sim_{\mathsf{eq}_i(a)} q_2'$,

then this indistinguishability also holds for $inj(q_2)$ and $inj(q_2')$, as this function only permutes adversary nonces. Uniqueness is satisfied by definition. For knowledge transfer, assume that q_1, q_1' are states of $\mathcal{C}_{Pr}$ with $q_1 \sim_{\mathsf{eq}_i(A)} q_1'$, and let q_2 be a state of ${}^{\mathcal{C}_{Pr}}\!/_{\equiv}$ with $(q_1, q_2) \in =$, in particular then $q_2 = Z^{-1}(q_1)$, where Z is the relation defined above. Obviously $q_2' = Z^{-1}(q_1')$ satisfies the required property that $q_2' \sim_{\mathsf{eq}_i(A)} q_2$, since by construction, different nonces introduced for the same adversary-sent terms are identities in the indistinguishibility relations of ${}^{\mathcal{C}_{Pr}}\!/_{\equiv}$.

Hence, consider the converse direction, in this case, $Z^{-1} \colon Q_2 \to Q_1$ is exactly the converse of the relation Z as defined above. By construction of Z, Z^{-1} is a function, i.e., uniqueness holds as required. The move transfer functions $\delta^{1\to 2}_{\dots}$ are those resulting from the construction of Lemma 6.9 for the adversary, and the identity function for the honest principals. Again, this choice is valid due to Proposition 6.6. Let $(q_1, q_2) \in Z$, i.e., let $q_1 = Z^{-1}(q_2)$.

Propositional Equivalence
This is trivial, as the propositional variables only depend on the local states of the principals, and these are the same in $\equiv$-equivalent states.

Move Uniformity
For honest principals this is trivial, as the move transfer function is simply the identity. For the adversary the claim follows from Lemma 6.9 for the direction of transferring moves from $\mathcal{C}_{Pr}$ to ${}^{\mathcal{C}_{Pr}}\!/_{\equiv}$.

Uniqueness
Follows from the construction of Z: There is exactly one state q_1 such that $(q_1, q_2) \in Z$ for every reachable state q_2.

Move Transfer
This directly follows from Lemmas 6.5 and 6.9. The adversary and the honest principals perform their moves in parallel, however since the result of the application of the honest principal's move does not depend on the adversary's move (we can without loss of generality assume that no term sent by the adversary in the current round is addressed by the terms appearing in the current move, otherwise we replace these with empty terms), we can assume that the adversary's move is performed first, and then the honest principals perform their actions.
Hence, let $d = \mathsf{eqdeg}\,(q_1) = \mathsf{eqdeg}\,(q_2)$. Applying the move transfer construction for the adversary results in a pair of intermediate states that are, due to Lemma 6.9, equivalent with degree $d'\mathsf{eqdeg}\,(q') + 2 \cdot d_{Pr}$, where q' is one of the two successor states. For the following move of the honest principals, we apply Lemma 6.5, where we instantiate d_1 with d', and d_2 with d_{Pr}. It follows that the states resulting from the application of both moves are equivalent with degree $d' - 2 \cdot d_{Pr} = \mathsf{eqdeg}\,(q')$ as required. By construction, the same local states of honest principals are reached with the same probability.

Uniformity and Knowledge Transfer
These follow in the same way as in the above direction, using the translation used for uniformity there for knowledge transfer in this direction and vice versa. Note that the arguments above trivially do not depend on whether we consider a single player or a coalition.

Decidability now follows, since we know due to Theorem 6.10 that $\mathcal{C}_{Pr}/\equiv$ can be computed from the representation of Pr, and there is a probabilistic bisimulation between $\mathcal{C}_{Pr}$ and $\mathcal{C}_{Pr}/\equiv$. Since the initial states of $\mathcal{C}_{Pr}$ and $\mathcal{C}_{Pr}/\equiv$ are identical, we have proven Theorem 6.2. This implies Theorem 4.1, since decidability in the finite model $\mathcal{C}_{Pr}/\equiv$ follows from the results in [Sch10a].

6.7 Decidability for Extension of the Protocol Model

The decision procedure for the extended protocol model suggested in Section 4 proceeds as follows: We add, for every test in the formula, an additional principal to the protocol system that performs this test as part of its protocol role (and modify existing principals to forward the necessary messages to the newly introduced test principal). The effect of this addition is that the construction used for the proof of the standard model ensures that the results of the tests are invariant under bisimulation, this follows directly from Proposition 6.6. Note that in this case, the structure $\mathcal{C}_{Pr}/\equiv$ does not only depend on the protocol Pr, but also on the formula that is to be evaluated. Also note that the condition that every principal only uses the secret keys and nonces of a single identity is not necessary for the decidability result, but is only required to obtain realistically executable programs.

It is clear that the addition about dynamically available channels does not pose a problem for the decidability procedure, since timing information is invariant under the bisimulation used in the main proof. This holds more generally for every situation in which the set of available channels is a function of the current protocol states of the principals.

7 Conclusion and Future Research

We introduced a decidable model that treats epistemic and strategic properties of probabilistic cryptographic protocols. We demonstrated that the expressiveness of the logic QAPI allows to express complex epistemic and probabilistic security properties. Advanced features as quantification, explicit strategies, and probabilistic reasoning were central in our modeling of the treated security properties. Open questions are a complexity analysis of the model checking problem, and extending decidability to a larger class of equational theories.

References

[AC06] Martín Abadi and Véronique Cortier. Deciding knowledge in security protocols under equational theories. *Theoretical Computer Science*, 367(1-2):2–32, 2006.

[AF01] Martín Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In *POPL*, pages 104–115, 2001.

[AHK02] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[ASW98] Nadarajah Asokan, Victor Shoup, and Michael Waidner. Asynchronous protocols for optimistic fair exchange. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 86–99. IEEE Computer Society Press, 1998.

[ASW09] Mihhail Aizatulin, Henning Schnoor, and Thomas Wilke. Computationally sound analysis of a probabilistic contract signing protocol. In Michael Backes and Peng Ning, editors, *ESORICS*, volume 5789 of *Lecture Notes in Computer Science*, pages 571–586. Springer, 2009.

[AT91] Martín Abadi and Mark R. Tuttle. A semantics for a logic of authentication (extended abstract). In *PODC*, pages 201–216, 1991.

[BAN90] Michael Burrows, Martín Abadi, and Roger M. Needham. A logic of authentication. *ACM Trans. Comput. Syst.*, 8(1):18–36, 1990.

[BM93] Colin Boyd and Wenbo Mao. On a limitation of ban logic. In *EUROCRYPT*, pages 240–247, 1993.

[BOGMR90] Michael Ben-Or, Oded Goldreich, Silvio Micali, and Ronald L. Rivest. A fair protocol for signing contracts. *IEEE Transactions on Information Theory*, 36(1):40–46, 1990.

[Cha88] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptology*, 1(1):65–75, 1988.

[CHP07] Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Strategy logic. In Luís Caires and Vasco Thudichum Vasconcelos, editors, *CONCUR*, volume 4703 of *Lecture Notes in Computer Science*, pages 59–73. Springer, 2007.

[DY83] Danny Dolev and Andrew Chi-Chih Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–207, 1983.

[EG83] Shimon Even and Oded Goldreich. On the security of multi-party ping-pong protocols. In *FOCS*, pages 34–39. IEEE, 1983.

[GJM99] Juan A. Garay, Markus Jakobsson, and Philip D. MacKenzie. Abuse-free optimistic contract signing. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 449–466. Springer, 1999.

[HT93] Joseph Y. Halpern and Mark R. Tuttle. Knowledge, probability, and adversaries. *J. ACM*, 40(4):917–962, 1993.

[JÅ06] Wojciech Jamroga and Thomas Ågotnes. What agents can achieve under incomplete information. In Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone, editors, *AAMAS*, pages 232–234. ACM, 2006.

[JvdH04] Wojciech Jamroga and Wiebe van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2-3):185–219, 2004.

[JYH] Ron van der Meyden Joesph Y. Halpern, Riccardo Pucella. Revisiting the foundations of authentication logics. manuscript.

[KKT07] Detlef Kähler, Ralf Küsters, and Tomasz Truderung. Infinite state AMC-model checking for cryptographic protocols. In *LICS*, pages 181—192. IEEE Computer Society, 2007.

[KKW06] Detlef Kähler, Ralf Küsters, and Thomas Wilke. A Dolev-Yao-based definition of abuse-free protocols. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 95–106. Springer, 2006.

[KKW09] Detlef Kähler, Ralf Küsters, and Thomas Wilke. Deciding properties of contract-signing protocols. *Transactions on Computational Logic*, 2009.

[KR02] Steve Kremer and Jean-François Raskin. Game analysis of abuse-free contract signing. In *CSFW*, pages 206–. IEEE Computer Society, 2002.

[KR03] Steve Kremer and Jean-François Raskin. A game-based verification of non-repudiation and fair exchange protocols. *Journal of Computer Security*, 11(3):399–430, 2003.

[KRS10] Steve Kremer, Mark Ryan, and Ben Smyth. Election verifiability in electronic voting protocols. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *ESORICS*, volume 6345 of *Lecture Notes in Computer Science*, pages 389–404. Springer, 2010.

[KST10] Ralf Küsters, Henning Schnoor, and Tomasz Truderung. A formal definition of online abuse-freeness. In Sushil Jajodia and Jianying Zhou, editors, *SecureComm*, volume 50 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 484–497. Springer, 2010.

[KT09] Ralf Küsters and Tomasz Truderung. An epistemic approach to coercion-resistance for electronic voting protocols. In *IEEE Symposium on Security and Privacy*, pages 251–266. IEEE Computer Society, 2009.

[KTV10a] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Accountability: Definition and relationship to verifiability. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM Conference on Computer and Communications Security*, pages 526–535. ACM, 2010.

[KTV10b] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. A game-based definition of coercion-resistance and its applications. In *CSF*, pages 122–136. IEEE Computer Society, 2010.

[PRST08] David C. Parkes, Michael O. Rabin, Stuart M. Shieber, and Christopher Thorpe. Practical secrecy-preserving, verifiably correct and trustworthy auctions. *Electronic Commerce Research and Applications*, 7(3):294–312, 2008.

[RT03] Michaël Rusinowitch and Mathieu Turuani. Protocol insecurity with a finite number of sessions, composed keys is NP-complete. *Theoretical Computer Science*, 1-3(299):451–475, 2003.

[Sch10a] Henning Schnoor. Explicit strategies and quantification for ATL with incomplete information and probabilistic games. Technical Report 1008, Institut für Informatik, Christian-Albrechts-Universität zu Kiel, 2010.

[Sch10b] Henning Schnoor. Strategic planning for probabilistic games with incomplete information. In Wiebe van der Hoek, Gal A. Kaminka, Yves Lespérance, Michael Luck, and Sandip Sen, editors, *AAMAS*, pages 1057–1064. IFAAMAS, 2010.

[Sch12] Henning Schnoor. Deciding epistemic and strategic properties of cryptographic protocols. In *Proceedings of ESORICS 2012*. Springer, 2012. *to appear*.