INSTITUT FÜR INFORMATIK

Deciding Epistemic and Strategic Properties of Cryptographic Protocols

Henning Schnoor

Bericht Nr. 1012
October 2010

CHRISTIAN-ALBRECHTS-UNIVERSITÄT

ZU KIEL

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

Deciding Epistemic and Strategic Properties of Cryptographic Protocols

Henning Schnoor

Bericht Nr. 1012
October 2010

e-mail: schnoor@ti.informatik.uni-kiel.de

Deciding Epistemic and Strategic Properties of Cryptographic Protocols

Henning Schnoor

October 5, 2010

Abstract

We propose a new, widely applicable model for analyzing knowledge-based (epistemic) and strategic properties of cryptographic protocols. The main result we prove is that the corresponding model checking problem with respect to an expressive epistemic extension of ATL* is decidable. As an application, we prove that abuse-freeness of contract signing protocols is decidable, resolving an open question. Further, we discuss anonymous broadcast and a coin-flipping protocol.

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 NP, even if the adversary is allowed to send arbitrarily complex terms [RT03]. Such techniques have led to algorithmic protocol verification [MS01] and were used to uncover problems in well-known protocols [Low96].

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 models for strategic analysis have the following limitations:

(i) They are not able to express *epistemic* properties of protocols. Such properties are concerned with knowledge of principals in the protocol.

Examples are abuse-freeness of contract-signing [KKW06] or anonymous broadcast.

(ii) They only consider *complete-information strategies*: Both honest principals and the adversary may base their strategic decisions on complete knowledge about the current state, including messages exchanged between other principals and secrets hidden by cryptographic means. Thus, capabilities of all parties are over-approximated, potentially leading to both "false positives" and "false negatives" in the security analysis.

(iii) They do not handle *probabilistic* protocols, where randomness is not only used in cryptographic primitives, but for random *decisions*. These are essential for some security goals [ASW09] and can be used to model random routing in anonymity protocols.

We propose a model that overcomes these shortcomings by a thorough treatment of knowledge and probabilism. Since standard ATL* does not have epistemic or probabilistic features, we need a more expressive logic to describe security goals. 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 (the non-epistemic, non-probabilistic) strategy logic [CHP07], of which QAPI is a proper generalization. This allows to express dependencies between strategies of different coalitions, as for example knowledge that one coalition has about the behavior of others. We use this powerful feature to express abuse-freeness of contract signing protocols. 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) in our model is decidable for both active and passive adversaries.

Our proof implies that relevant strategies can always be finitely represented. This, and the fact that strategies only use "realistic" information, implies that if there is an "attack" on a protocol, the resulting adversary can be implemented in software. Similarly, if a protocol is "secure," the relevant strategies for honest principals can be implemented.

As an example, consider the following coin-flipping protocol: Bob randomly chooses a bit $b_1 \in \{0,1\}$ and a long random value 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 values match the hash. The outcome of the protocol is the bit $b_1 \oplus b_2$, the main security

property is that neither Alice nor Bob have a strategy to dictate the outcome. In the classical, complete-information setting, this security property is not met: Here Alice may choose b_2 depending on the value of b_1, and thus can completely control the resulting bit $b_1 \oplus b_2$. Intuitively, the protocol is secure, since Alice is unable to determine b_1 from the hash value. Our model allows to formally specify and prove the protocol secure.

In this report, we only treat the above toy protocol in detail. As an additional application, we prove that abuse-freeness can be formalized in our model. As a corollary we obtain that abuse-freeness is decidable, resolving an open question from [KKW06].

Related Work. In the above-mentioned [KKT07], a decision algorithm for (non-epistemic, complete-information, non-probabilistic) strategic properties of protocols is given. The authors establish bounds for decidability of protocol verification that preclude generalizations of our results to various extensions of our model that allow infinite protocol runs. 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.

[ASW09] defines a model for probabilistic protocols, however no decidability result is proven. We significantly generalize that model in several directions: First, we take into account explicit knowledge and incomplete information strategies. 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 the protocol model. In Section 3, we briefly introduce the semantics of 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 applications: A treatment of the coin-flipping protocol, our decidability result for abuse-freeness, and a brief discussion of anonymous broadcast. In Section 6, we prove our main result and conclude in Section 7.

1 Syntax: Specifying a Protocol

1.1 Two Examples

Before introducing our formal model, we consider two examples: The coin-flipping protocol (see Introduction), which we fully specify in our model, and

an excerpt of a contract-signing protocol.

The Coin-Flipping Protocol In the protocol, Bob chooses his bit first and thus cannot dictate the outcome of the protocol ("cheating" is noticed by Alice when verifying the hash value). We consider the more interesting case of dishonest Alice: Only the hash function prevents her from dictating the result unilaterally. Hence the more interesting case is when we identify Alice with the adversary, and assume that she will not follow the protocol. In Figure 1, we show the formalizations of both Alice's and Bob's role in the protocol in our model; the structure of Alice's role is shown on the left-hand side, the modeling of Bob's role is shown on the right-hand side. As mentioned above, we first discuss the case that Bob follows the protocol: Dashed lines represent messages received by Bob, solid ones correspond to messages sent by him. The message $\langle \alpha, N \rangle$ is a pair containing the bit α and the long random value N. The probabilities $\frac{1}{2}$ express that Bob chooses the bits 0 and 1 with probability $\frac{1}{2}$ each. Omitted probabilities have the value 1. Note that different messages from Alice (0 or 1) lead to different reactions (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 protocol step when Alice is active. The security property for the protocol is captured in the following formula:

$$\forall_3 \mathrm{S} \neg \left\langle\!\left\langle \mathcal{A} : \mathrm{S} \right\rangle\!\right\rangle^{>0.5} \Diamond \left(\mathsf{fin}_{00}^{B} \vee \mathsf{fin}_{11}^{B} \right).$$

The variable S is instanciated with a universally quantified strategy played by Alice. The formula expresses that for every[1] strategy S, if the adversary Alice follows the strategy, 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 define the logic in detail in Section 3.

For the case that Alice follows the protocol, we also discuss her modelling in our model (see also Figure 1). To increase readability, instead of using the formal operators that allow Alice to extract the message received in the i-th protocol step, we simply write r_i to access the corresponding term. The test (see below for a formal definition of tests) true represents a test that is always true, for example $x = x$, where x is a variable. In the example, this is used when the hash value of Bob's pair $\langle b_1, N \rangle$ is received: At this point of the protocol run, no tests are performed, the value is merely stored for later reference.[2]

[1] The index 3 expresses that the strategy may not break the hash function, see later for details.

[2] Note that of course, Alice could perform syntactical tests at this time to e.g., rule out

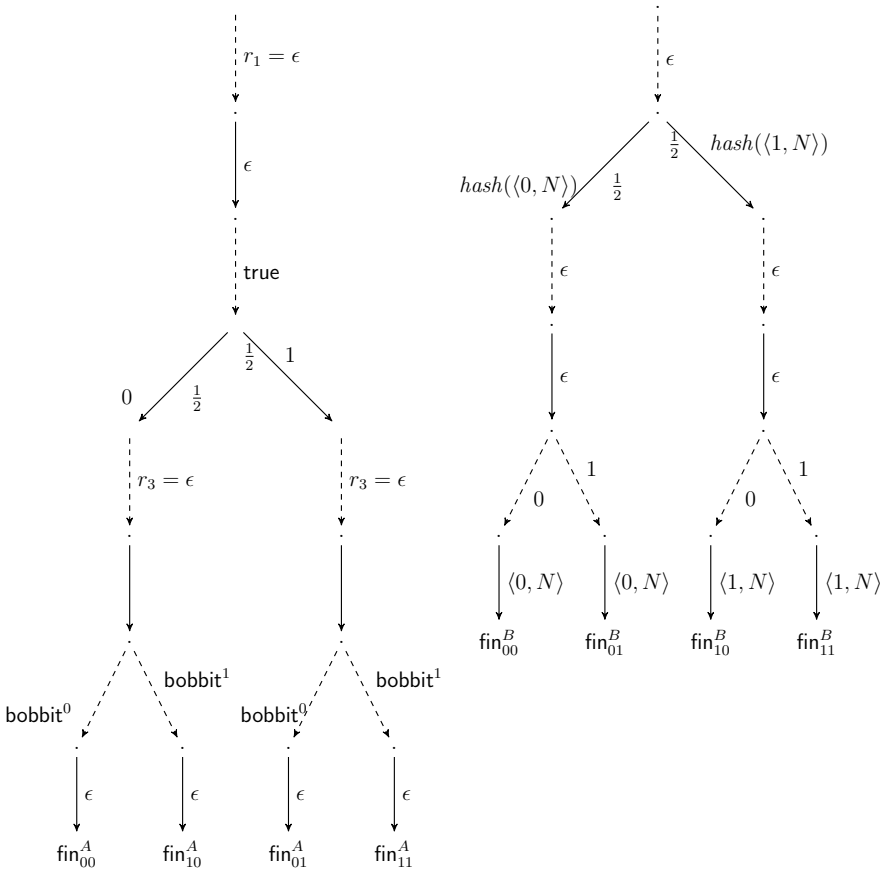

Figure 1: Specification of Coin-Flipping Protocol

The final receive step made by Alice is the most important one: Here she receives the actual pair $\langle b_1, N\rangle$ from Bob. Alice now needs to check that Bob did not cheat (i.e., that this pair is indeed consistent with the hash value received earlier in the protocol run), and to compute the result of the coinflip. In order to implement 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 if and only if the pair sent by Bob in protocol step 4 matches the earlier sent hash value and the bit contained in Bob's commit-

that the received message is a complicated term instead of a simple hash value. However, this is unnecessary, since the test performed in the final step involves comparing the message received here to a hash value which Alice computes in the final step—if this test is successful, then in particular, the value first received from Bob is a hash value.

ment is α. Recall that 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).

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 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 coin-flipping protocol discussed above, and random routing.

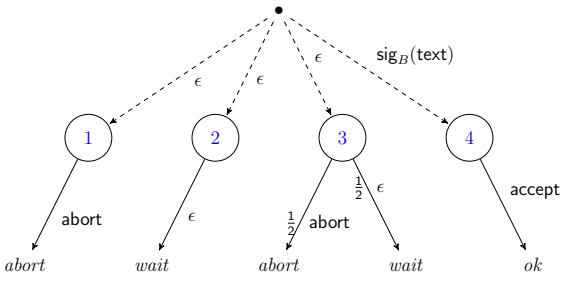

Figure 2: Protocol State Example

The formal protocol definition below is as expected—however we make the following generalization: In the examples, incoming messages are compared to some "expected" message. In general, principals may not know (or be interested in) the exact message, but only some properties of it. Such properties are modeled as *tests* performed on messages. As an example, when Alice receives the second message from Bob in the coin-flipping protocol, she verifies that the hash value of this message is exactly the first message she received. This can be done using the test $hash(r_2) = r_1$, where r_1 and r_2 refer to the two messages received from Bob.

1.2 Formalizing Protocol States

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

$i \in \text{IDs}$, there are constants i, pk_i and sk_i in Σ^{t}, denoting the name, public and private key of i, and that Σ^{t} contains operations to construct tuples and projection functions to access their components. For $C \subseteq \text{IDs}$, the set T_C is the set of terms constructable from $X \cup \bigcup_{i \in C} \mathcal{N}_i \cup \mathcal{N}_{\mathcal{A}}$ 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 (i.e., the adversary has access to their secrets). We omit set brackets for singletons.

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

Figure 3: Equational theory for asymmetric encryption, signatures, and pairing

We additionally assume a convergent equational theory E associated with Σ^{t}, and denote the resulting congruence relation with $\equiv_{\mathrm{E}}$. See Figure 3 for an example theory describing public-key encryption, signatures, and pairing; in the equations x_i refers to an identity, x_t is a term (message), and x_r represents randomization used in the application of cryptographic primitives. The (uniquely determined) *normal form* of a term t, denoted with $[[t]]$, is obtained by exhaustive application of simplification rules from E. For the example theory in Figure 3, if $t = \mathsf{dec}_{\mathsf{sk}_A}\left(\mathsf{enc}_{\mathsf{pk}_A}(\mathsf{abort})^r\right)$, then $[[t]] = \mathsf{abort}$.

More formally, an *equation* over Σ^{t} is simply a pair of Σ^{t}-terms (l, r), which we also write as $l = r$, where l is the *left-hand side*, and r is the *right-hand side* of the equation. An *equational theory* E over Σ^{t} is a set of equations over Σ^{t}. As an example, the equation

$$\mathsf{dec}_{\mathsf{sk}_{x_i}}\left(\mathsf{enc}_{\mathsf{pk}_{x_i}}(x_t)^{x_r}\right) = x_t$$

models that when encrypting a term (represented with the variable X_t) with the public key of some identity (represented with the variable X_i), using some randomness specified in the variable X_r, and decrypting the term with the private key of the same identity, then the result is the originally encrypted term.

The above equation can be seen as a "simplification rule," which allows to transform a complex term representing the ciphertext into a simpler term, representing the plaintext. It is of course possible that the plaintext itself again is a ciphertext, hence the equation can possibly be performed multiple times, each step leading to a simpler term. However, after some finite number

of steps, the resulting plaintext is not an encryption of another term anymore, and the rule cannot be applied further.

More generally, an equational theory E induces a *reduction relation* $\twoheadrightarrow_{\mathrm{E}}$, where for two terms t_1 and t_2, we have that $(t_1, t_2) \in \twoheadrightarrow_{\mathrm{E}}$ (also written as $t_1 \twoheadrightarrow_{\mathrm{E}} t_2$) if t_2 can be obtained from t_1 by applying a rule in E, i.e., if there is an equation $l = r$ in E such that

- the set of variables of l is $X_l = \{x_1^l, \ldots, x_n^l\}$ for some variables $x_1^l, \ldots, x_n^l$,
- the set of variables of r is $X_r = \{x_1^r, \ldots, x_m^r\}$ for some variables $x_1^r, \ldots, x_m^r$ (the sets X_l and X_r need not be disjoint), and
- there is a function $\sigma\colon X_l \cup X_r \to \mathcal{T}_{\Sigma^{\mathrm{t}}}$ such that
  1. $l[x_1^l/\sigma(x_1^l), \ldots, x_n^l/\sigma(x_n^l)] = t_1$,
  2. $r[x_1^r/\sigma(x_1^r), \ldots, x_n^r/\sigma(x_n^r)] = t_2$.

In this case, t_1 is an instance of l, and t_2 is the instance of r which agrees with t_1 on the values on the appearing variables. Hence it is possible to apply the equation $l = r$ to obtain t_2 from t_1.

With $\twoheadrightarrow_{\mathrm{E}}^*$, we denote the reflexive and transitive closure of $\twoheadrightarrow_{\mathrm{E}}$, with $\equiv_{\mathrm{E}}$ we denote the reflexive, symmetric, and transitive closure of $\twoheadrightarrow_{\mathrm{E}}$. Terms t_1 and t_2 are called Σ^{t}-*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.

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

A *message* is a variable-free term in normal form. If x is a variable and t, t' are terms, then with $t[x/t']$ we denote the term resulting from simultaneously replacing in t every occurrence of x with t'.

Definition An *atomic C-test* [KKW06] 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 $M[x/m] \equiv_{\mathrm{E}} M'[x/m]$. 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 in the obvious way. We now define local protocol states. These specify how an incoming message

is handled in a step of the protocol: Depending on the properties of the message (modelled with a set of tests), there are different options (called *choices* below) available. To express randomness in the protocol (as the $\frac{1}{2}/\frac{1}{2}$ probabilities for the coin flip protocol), these options are probability distributions over actions, where an action consists of a reply message and a local state change. In the definition below, the parsing sequence corresponds to the dashed lines in the earlier graphical examples, while the send sequence formalizes the solid lines. A state therefore consists of the dashed lines originating at the same point plus their solid successors.

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

- a *parsing sequence* $t_1, \ldots, t_k$, where each element 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 as the specifications of Alice and Bob in Figure 1. It combines states into a tree, modeling different possible protocol executions. We assume sufficiently many copies of Finished, so that a protocol role may have different final states.

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

- If $w \in V$ is regular with 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 term in every appearing regular protocol state is an i-term, i is also called the *identity* of $\mathcal{R}$.

The requirement that the identity exists ensures that a protocol role 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 and then formally define it with a *game structure* containing all available actions and consequences (see also [KKT07]). We re-use notation from above, in particular, k denotes the number of honest protocol participants. Messages sent by principals are tuples, where for each principal, the incoming message in each state is a $(k+1)$-ary tuple, which in component i contains a 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 $i = k + 1$.

In a protocol run, an honest principal $h \in \{1, \ldots, k\}$ acts as follows: In each state, he analyzes the incoming message tuple, and checks for each test from the parsing sequence whether the message satisfies it. To allow comparing parts of the currently received message to previous messages, the test is applied to the entire sequence of messages received so far by the principal. If the test t_c is satisfied, a number $d \in \{1, \ldots, l\}$ is chosen randomly according to the distribution specified by $\alpha_{c,1}$, $\ldots$, $\alpha_{c,l}$, and the term $s_{c,d}$ is the reply sent by h. This term may contain a variable, which again refers to the sequence of previously received messages, allowing parts of these to be included in the outgoing message. 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, i.e., if more than one c above is possible, the principal can make a *strategic choice* by choosing the one to apply. To avoid cumbersome case distinctions, we require that for every message, there must be a test that it satisfies[3].

The adversary may send arbitrary terms that he can construct.

2.1 Concurrent Game Structures

The formal protocol model combines a set of "global states" of a protocol (a global state essentially contains the local state of every involved party), with the possible actions (called "moves") and consequences thereof for every party. The standard way to specify strategic situations as these are concurrent game structures (CGS). We use the definition from [Sch10b], which models probabilistic games and incomplete information:

[3]This can be achieved by adding dummy tests and reactions; this also models that realistically, principals may choose to ignore incoming messages. If unwanted, one can require that honest principals only use protocol steps obtained from using the dummy test if no other test applies, our results hold for both versions.

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

- 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* ($\pi(p)$ is the set of states where p is true),
- Δ is a *move function* assigning to each state $q \in Q$ and player $a \in \Sigma$ 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 state obtained when in q, all players perform their move as specified by c),
- eq is an *information function* $\mathsf{eq}\colon \{1, \ldots, n\} \times \Sigma \to \mathcal{P}(Q \times Q)$, where n is a natural number, and for each $i \in \{1, \ldots, n\}$ and $a \in \Sigma$, $\mathsf{eq}(i, a)$ is an equivalence relation on Q. Each $i \in \{1, \ldots, n\}$ 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 the player a cannot distinguish states q_1 and q_2 (if i denotes the degree of information available to him). Multiple degrees of information allows to reason about situations where e.g., the adversary is assumed to be able to break cryptography, but not see other principal's internal states, or situations where he is restricted by cryptography.

2.2 The Concurrent Game Structure for Protocol Execution

We now formalize the protocol execution described earlier, by defining a CGS that contains the possible actions of the involved parties (honest principals and the adversary). In the state description below, C is the set of corrupted identities, each honest principal $h \in \{1, \ldots, k\}$ is in state w_h. For each principal $i \in \{1, \ldots, k, \mathcal{A}\}$, 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. Variables of the CGS allow to express facts about the local state of honest principals.

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

- $\Sigma = \{1, \ldots, k, \mathcal{A}\}$,
- Q consists of tuples of the form $q = (C, w_1, \mathcal{M}_1, \ldots, w_k, \mathcal{M}_k, \mathcal{M}_\mathcal{A}, \mathrm{moves}_\mathcal{A})$, where $C \subseteq$ IDs, for each $i \in \{1, \ldots, k\}$, 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 $h \in \{1, \ldots, k\}$ there is a variable st^h_w that is true in q as above if and only if $w_h = w$,
- for a state q as above where for all $h \in \{1, \ldots, k\}$, w_h has k_h choices, randomization degree l_h, parsing sequence $t^h_1, \ldots, t^h_{k_h}$ and send sequence $(s^h_{1,1}, \alpha^h_{1,1}), \ldots, (s^h_{k_h,l_h}, \alpha^h_{k_h,l_h})$, 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 $h \in \{1, \ldots, k\}$, the number $c_h \in \{1, \ldots, k_h\}$ 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 $(c_1, \ldots, c_k)$ and numbers $d_1, \ldots, d_k$, where $1 \leq d_h \leq l_h$, there is a successor state $q' = (C, w'_1, \mathcal{M}'_1, \ldots, w'_k, \mathcal{M}'_k, \mathcal{M}'_\mathcal{A}, \mathrm{moves}_\mathcal{A} \circ m_\mathcal{A})$, where
  - w'_h is the unique successor of w_h in $\mathcal{R}_h$ connected with the edge labeled (c_h, d_h),
  - to define the sequences $\mathcal{M}'_j$, we denote with M_i for $i \in \{1, \ldots, k, \mathcal{A}\}$ the *message sent by* i, which is $[[s^i_{c_i,d_i}[x/\mathcal{M}_i]]]$ if $i \leq k$, or $[[m_\mathcal{A}[x/\mathcal{M}_\mathcal{A}]]]$ if $i = \mathcal{A}$,
  - for all $i \in \{1, \ldots, k, \mathcal{A}\}$, the new sequence $\mathcal{M}'_i$ is obtained by adding to the sequence $\mathcal{M}_i$ a $(k+1)$-ary tuple containing in its j-th component the i-th component of M_j (the i-th component of M_j is 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 the state Finished, he only has dummy moves, i.e., does not change local state, receive or send messages anymore.
- We define three degrees of information: For a player $a \in \Sigma$,
  1. $\mathsf{eq}(1, a)$ is the equality relation (this models complete information),
  2. $\mathsf{eq}(2, a)$ is the equivalence relation where two states are equivalent if and only if the principal is in the same local state[4], 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

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

and only if the principal is in the same local state[4], and components $\mathcal{M}_a$ are a-indistinguishable (C-indistinguishable if $a = \mathcal{A}$).

For each $C \subseteq$ 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 local state, and the adversary has access to the keys of all identities in C. This models *static corruption*, where a set of identities (fixed before the protocol run) is treated as adversial.

Formally, a principal receives a single message in each step. This message is a tuple containing messages from every protocol principal: Several messages can be received, processed, and answered simultaneously. Messages are immediately delivered to the intended recipients: There are direct secure channels between principals. Realistically, use of such channels will be restricted by introducing so-called *buffer principals* which the adversary may instruct to delay/drop messages. These are modeled as ordinary protocol roles relaying messages. Hence our model allows for flexible "implementations" of secure channels. These also allow to model passive adversaries, by only letting honest principals communicate via these buffers, and ignoring incoming messages from the adversary (who is active by default).

There is no formal requirement forcing messages to be the intended $(k+1)$-ary sequences. If messages of a different form are sent by some $i \in \{1, \ldots, k, \mathcal{A}\}$, then some entries in the tuples are not defined, and some possible receivers do not receive a message from i.

3 QAPI: ATL with probabilism, knowledge, and explicit strategies

To express security goals, we use the ATL*-variant QAPI introduced in [Sch10a]. We briefly introduce syntax and semantics of QAPI. We only define the subset of the available features of QAPI that is most relevant to expressing properties of cryptographic protocols. However, we mention that our decidability result holds for the complete language.

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 strate-

gies, 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 $\mathsf{X}\varphi_1$, $\mathsf{P}\varphi_1$, $\mathsf{X}^{-1}\varphi_1$, and $\varphi_1 \mathsf{U} \varphi_2$ are path formulas.

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 $\mathrm{S}_1, \ldots, \mathrm{S}_n$, 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

Strategies are defined as usual: For a player a, an *a-strategy* 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. Our strategies are *memoryless*: In our model, principals store all relevant information—each state in $\mathcal{C}_{Pr}$ has a unique history. We now define the semantics of the subset of QAPI that we use. 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 there is some $t' \leq t$ and $(\lambda, t'), \overrightarrow{\mathsf{S}} \models \psi_1$,
- $(\lambda, t), \overrightarrow{\mathsf{S}} \models \mathsf{X}^{-1}\psi_1$ 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[5] 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 are already fixed as instantiations of the variables S_i. A *quantified strategy formula* is a state formula as above, 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," the semantics is the natural one: $\exists_{i_1} \mathrm{S}_1 \forall_{i_2} \mathrm{S}_2 \ldots \exists_{i_n} \mathrm{S}_n \varphi$ is true in a 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. Quantification and explicit strategies lead to an expressive logic that can express dependencies between strategies, e.g., captures situations where a player has knowledge about strategies played by others. In particular, QAPI allows very flexible treatment of the behavior of the "counter-coalition:" In ATL*, $\langle\langle A \rangle\rangle \varphi$ implicitly quantifies about arbitrary behavior of the players in $\overline{A} := \Sigma \setminus A$. With explicit strategies and quantification, QAPI can express, e.g., that $\overline{A}$ follow (i) their "currently played" strategy, (ii) a specific strategy to stop A from reaching the goal φ (iii) or perform an arbitrary sequence possible not consistent with any uniform strategy.

This can be expressed with $\left\langle\left\langle A : \mathrm{S}_A, \overline{A} : \mathrm{S}_{\overline{A}} \right\rangle\right\rangle \varphi$ for appropriate variables, or (for the third option) by not mentioning $\overline{A}$ in the operator at all—not all

[5] If a player a appears in more than one of the A_{i_j}, he follows strategy S_{i_j} with $j = \min\{j \mid a \in A_{i_j}\}$.

coalitions need to appear in an application of the $\langle\langle A_1 : \mathrm{S}_1, \ldots, A_k : \mathrm{S}_k \rangle\rangle^{\blacktriangleleft\alpha}$-operator. Formal definitions are in [Sch10a].

4 Main Result

We show that security of protocols in our model is decidable:

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

Intuitively, the theorem is true due to the following: Since honest principals only analyze the terms visible to them up to a bounded depth, the content of terms below a certain depth is irrelevant (see also [RT03]). Therefore, one can restrict the adversary to send terms with bounded depth. This gives an essentially finite model and allows application of standard model-checking algorithms. The actual proof is more involved because it has to ensure that not only reachability properties, but also strategic and epistemic properties are identical in the original and the restricted model. The complete proof can be found in Section 6 below.

Extensions Our model can be extended in many ways. For changing network configurations, one can define variations where communication between some principals is not allowed at all, or only allowed after a certain number of protocol steps have been performed, etc.

Also, we can enrich the set of propositional variables of $\mathcal{C}_{Pr}$ with statements $t(\mathcal{M}_i) \equiv_{\mathrm{E}} t'(\mathcal{M}_j)$ for terms t and t' and $i, j \in \{1, \ldots, k, \mathcal{A}\}$, with the obvious semantics: This models tests on messages visible to principals i and j. In combination with the knowledge operator, this allows formulas to explicitly reason about the knowledge that a principal has about properties of messages received by himself or others during the protocol run.

One can also adept our model to obtain a sequential one. Our treatment of strategies allows to reason about fair scheduling, etc. We do not formalize these extensions, however we mention that decidability is maintained for the extended models.

5 Applications

We briefly discuss applications of our model. Treatment of uniform strategies allows more fine-grained analysis of protocols, since the appearing strategies

can in fact be implemented. In particular, if an analysis detects an adversary-strategy "breaking" the protocol, this corresponds to a realistic attack. Similarly, existence of strategies for honest principals (as, e.g., required by the timeliness property for contract signing protocols) also implies the existence of a strategy that honest principals can in fact follow with the available information.[6] Restriction to these strategies allows to show that the coin-flipping protocol is secure, which cannot be done in complete-information models. Other examples for applications are anonymous broadcast protocols and abuse-freeness of contract signing protocols, here epistemic capabilities of our model play a crucial role.

5.1 The Coin-Flipping Protocol

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 the value N. Therefore, a 3-uniform strategy has to choose the same action for both of Bob's possible messages.

5.2 Abuse-freeness

Abuse-freeness of e.g., contract signing protocols is the following: Assume Alice wants to buy a house from Bob for some amount of money, and the following situation arises: Bob has a strategy to "abort" the signing (ensuring that Alice does not get Bob's signature), another strategy to "close the deal" (to receive Alice's signature), and further Bob can prove this fact to an outsider Charlie. Then Bob can convince Charlie to pay more for the house than Alice offers. A protocol where such a situation does not arise is *abuse-free*.

Definitions of abuse-freeness are non-trivial [KR02, KKW06], but it is clear that this property contains both *epistemic* and *strategic* properties: Roughly, a protocol is abuse-free if there is no reachable state where Charlie *knows* that the adversary has strategies satisfying the above conditions.

We prove that the definition from [KKW06] can be expressed in our model. In addition to the obviously required epistemic aspects, our formula expressing abuse-freeness makes extensive use of QAPI's ability to reason about strategies directly: Our modeling of abuse-freeness quite naturally

[6] Using information degree 1, our model can treat complete-information strategies as well.

requires that other principals "know" the set of proofs accepted by Charlie. Since accepting/rejecting a proof is a choice by Charlie, this set of valid proofs corresponds to an accept/reject strategy played by him. The set of accepted proofs needs to be constant throughout the protocol run. Hence for consistent behavior, it is essential that Charlie does not change his strategy, even if other principals change theirs. This requirement cannot be expressed in standard ATL*: Here, an application of the $\langle\langle A \rangle\rangle$-operator lets players who are not members of A "forget" about their currently played strategy. QAPI allows to directly assign an arbitrarily quantified strategy to a coalition, where the same (variable for a) strategy may be used multiple times in a formula. Therefore, the "consistency" of Charlie and similar aspects required in the construction can be expressed in QAPI. The below result that abuse-freeness can be expressed in our model and our main result imply

Corollary 5.2 *Abuse freeness as defined in [KKW06] is decidable.*

The remainder of Section 5.2 proves Corollary 5.2. We first, in Section 5.2.1, recall the definition of abuse-freeness given in [KKW06]. In Section 5.2.2 we prove the result, and in Section 5.2.3 we consider some arguably more natural variations of the definition.

5.2.1 Formal definition of abuse freeness

Abuse-freeness is closely related to *balance*: Let φ_1 and φ_2 be formulas describing protocol outcomes. In the contract-signing example mentioned in Section 5.2, these might be formulas encoding that Bob has obtained a contract with Alice, or that the protocol run (for Alice) is over without Alice having received a contract. A state q of the protocol execution is (φ_1, φ_2)-*unbalanced*, if the adversary has a strategy ensuring that the resulting protocol run satisfies φ_1, and a (potentially different) strategy to ensure that it satisfies φ_2. A protocol is (φ_1, φ_2)-*abusive*, if the adversary can reach an (φ_1, φ_2)-unbalanced state, and additionally the adversary can present a proof of this fact to an outside party Charlie. This informal definition will be formalized in the sequel. In the following, we often only write "abusive" and "unbalanced" for (φ_1, φ_2)-abusive and (φ_1, φ_2)-unbalanced.

When describing abuse-freeness in game-theoretic settings, often an explicit variable *prove2C* [KR02] has been used to label the states in which the adversary can produce a corresponding proof. In [KKW06], Kähler, Küsters, and Wilke presented a definition of abuse-freeness that does not require the explicit labeling of these states, by formalizing what it means for the adversary to convince an outside party of the fact that a state of the protocol

is unbalanced. Their notion of abuse-freeness deals with deterministic protocols only and is *offline*, i.e., Charlie receives only a single message, and does not actively take part in the protocol run. Based on this message alone, Charlie decides whether to believe Bob the adversary or not. Since Charlie cannot force the adversary to present a proof as soon as one is available, the adversary can only convince Charlie that he is or was in an unbalanced state (the protocol run may be completed at the time that Charlie receives the proof from the adversary). Formally, the definition uses *tests* that Charlie may perform on proofs presented by the adversary, where the definition of a test is the same one we use, see Section 1. A test θ is called *convincing* in [KKW06], if every state q in the protocol run in which the adversary can produce a message m that satisfies θ has the property that there is an ancestor state q' of q such that 1. q' is (φ_1, φ_2)-unbalanced, 2. in q', the adversary can produce a message satisfying θ.

The second condition is necessary to, for instance, preclude protocols in which the adversary can produce proofs of unbalance of a previous state only after the protocol run is essentially over. The definition ensures that 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 Charlie (if the latter property is not satisfied, for example the always-false test could be used). A protocol is abuse-free otherwise.

5.2.2 Proof of Expressibility and Decidability

We now show that the question whether a protocol is abuse-free is decidable for protocols that can be expressed in our model[7], resolving an open question from [KKW06].

We mention that the version of abuse-freeness that we treat in this section is a slight modification of the definition in [KKW06]. The difference is the treatment of knowledge of strategies: Our notion of a strategy essentially requires participants not only to *have*, but also to *know* a strategy, while the definition from [KKW06] also allows strategies which exist, even are uniform, but cannot be "identified" by the principals. We feel that requiring strategies to be identifiable is more natural, see also [JvdH04]. However, we stress that the decidability result for the original definition from [KKW06] is true as well. We discuss this issue in more detail in Section 5.2.3. We also mention

[7]The model in [KKW06] is general enough to cover protocols represented by arbitrary Turing machines, hence in general abuse-freeness is obviously undecidable.

that for the protocols treated in [KKW06], this issue does not arise due to the relatively simple set of relevant strategies.

In the following, we assume that Alice is the honest protocol participant (along with other parties required by the protocol like possibly a trusted third party, buffer principals implementing secure channels, etc), and Bob is the adversary. Hence formally, we assume that the adversary has access to Bob's private key. Obviously, the case of dishonest Alice can be treated in the same way. Note that the definition of abuse freeness in [KKW06] is concerned with a deterministic model only, hence in the following we assume that the protocol does not use probabilism. For easier readability, we omit probabilities from the occurring formulas; all goals are supposed to be reached with probability ≥ 1.

We show the following theorem:

Theorem 5.3 *There is an algorithm which, when given a protocol Pr and path formulas φ_1 and φ_2 for protocol, produces a protocol Pr', and formula ψ^{abuse} such that Pr is (φ_1, φ_2)-abusive if and only if $\mathcal{C}_{Pr'}, q_{init}^{\{B\}} \models \psi$.*

Obviously, the analogous statement for Alice instead of Bob is true as well. Due to our main decidability result Theorem 4.1, the above theorem directly implies Corollary 5.2.

The remainder of this section proves Theorem 5.3. We first explain the changes made to the protocol Pr to obtain the protocol Pr'. In essence, Pr' is obtained from Pr by adding a principal representing Charlie (denoted with C in the QAPI-formulas below) and a verifier principal (referred to using V) whose role is to ensure that Charlie only bases his decision whether to accept the proof of the adversary on knowledge that he has in the model of [KKW06]. In detail, the changes are as follows:

- we introduce a principal Charlie who receives a message from the adversary at some point during the protocol run,
- we introduce another principal, the verifier, who receives a message from Charlie at the last step of the protocol, but does not perform any actions (i.e., the protocol role is simply a line), and has access to the same private keys as Charlie[8],
- the protocol role for Charlie proceeds as follows: As the last two steps of the protocol, Charlie

[8]this can be implemented by letting Charlie perform dummy tests which use Charlie's secret key, note that nonces produced by Charlie do not appear in the protocol run and thus are irrelevant

1. forwards the first message received from the adversary to the verifier (additional messages received from the adversary are ignored) in the transition to the second-to-last state,
2. nondeterministically moves into either an "accept" or a "reject" state in the transition to the final state.

Note that in particular, Charlie makes his decision only *after* the verifier received Charlie's message. This is for technical convenience, since it allows us to express the verifier's knowledge about Charlie's decision in an easy way.

The idea of the construction is the following: At some point during the protocol run, the adversary sends a message to Charlie. This gives Charlie additional information beyond the actual content of the message, namely an upper bound on the number of steps performed in the protocol previous to the adversary being able to construct the message. Since in the definition of abuse-freeness from [KKW06], Charlie does not have access to this kind of information (he only knows the message without any kind of timing information), we need to make sure that Charlie's decision—to accept or to reject the proof—is being made independently of the time when the message is received. This is realized by including the additional verifier principal: We require that the verifier *knows* Charlie's decision (i.e., Charlie's final local state), and the only information that the verifier has about Charlie's state is the message which Charlie forwards to him in the final protocol step. Of course, the adversary is free to send proofs of some states being unbalanced to the verifier as well, hence the verifier may very well know whether an unbalanced state has appeared—but this is irrelevant, since it is the task of the verifier to determine whether Charlie accepts the proof he received from the adversary, and not to perform any reasoning as to whether a state is unbalanced. The role of the verifier is merely to "force" Charlie to base his decision only on the information that we want him to use, i.e., the actual message received by the adversary.

We now construct the formula ψ^{abuse} expressing abusiveness—in the formula, we use letters instead of numbers to denote the principals as Charlie and the verifier:

- let φ^{unbal} be the formula $\langle\langle \mathcal{A} : \mathrm{S}_1^{\mathcal{A}} \rangle\rangle\, \varphi_1 \wedge \langle\langle \mathcal{A} : \mathrm{S}_2^{\mathcal{A}} \rangle\rangle\, \varphi_2$, i.e., the formula 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 formulas which are true if Charlie is in a local state where

he accepts, respectively where he rejects, this formula expresses that in the current state, the verifier knows whether Charlie will move into the accepting or rejecting state, given that Charlie plays the strategy referred to with the symbol S_{C}.

– let $\varphi^{\mathcal{A}-\text{greedy}}$ be the formula $\neg\mathsf{P}\,((\langle\langle\mathcal{A}:\mathrm{S}_{\mathcal{A}},\mathsf{C}:\mathrm{S}_{\mathsf{C}}\rangle\rangle\,((\mathsf{Xrec})\wedge(\Diamond\mathsf{acc})))\wedge\neg\mathsf{Xrec})$, where rec is a formula true in exactly those states in which Charlie has received a message from the adversary. This formula expresses that (assuming the strategy $\mathrm{S}_{\mathcal{A}}$ is all-quantified over all complete-information strategies) the strategy played by the adversary up to now in the protocol run is "greedy" in the sense that if at some point in the protocol run it was possible for the adversary to send a message which eventually leads to Charlie (assuming he is following S_{C}) moving into the accepting state, then the adversary in fact did send a message to Charlie in the next step (or previously).

Also, let end be an atomic proposition which is true at the "end of the protocol run," i.e., when the verifier has exactly one final action left. Then the "normal protocol execution," i.e., the actions of the principals of the original protocol, are finished by the construction of Charlie and the verifier. The formula ψ^{abuse} is the following:

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

Here we use $\mathsf{P}_{\Box}$ as abbreviation for $\neg\mathsf{P}\neg$, i.e., $\mathsf{P}_{\Box}\varphi$ expresses that φ is true at every state in the past. Further, with slight abuse of notation Σ denotes the principals in the original protocol, i.e., everyone except Charlie and the verifier, but including the adversary, and rec' is an atomic proposition true exactly in those states where Charlie just received the first non-empty message from the adversary. The reason why the quantification for $\mathrm{S}_1^{\mathcal{A}}$ and $\mathrm{S}_2^{\mathcal{A}}$ only requires strategies to be uniform for information degree 2 (i.e., essentially allow the adversary to break cryptography when deciding on his next move—although not in *executing* this move, i.e., he may base his decisions on the content of hidden plaintexts, but may not send these plaintexts) is because the definition in [KKW06] only requires that strategies for the adversary base their decisions on the "view" of the adversary, which is appropriate in their setting (see comments in Section 5.2.3). Hence we use information degree 2 to obtain exactly their definition, for more general situations information level 3 (i.e., the adversary does not have access to cryptographically hidden information to decide on his actions) may be more appropriate. Also note

that S_Σ is quantified over complete-information strategies, this expresses that a state in which acc is true is reachable (note that since the model treated in [KKW06] is not probabilistic, a state is reachable if and only is the set of all players has a complete-information strategy to reach it).

The three conjuncts require that

1. at the second-to-last protocol step, i.e., after the verifier received the forwarded message from Charlie, the verifier knows whether Charlie will move to an "accept" or a "reject" state in the next transition. This ensures that, as mentioned above, Charlie bases his decision only on the content of the first message received from the adversary, and does not take additional messages or timing information into account. Hence this establishes that the strategy that Charlie uses (which only allows him one strategic move, namely to accept or reject the adversary's proof) is exactly defined by the outcome of a test.
2. whenever Charlie 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 Charlie—as long as the adversary played a "greedy" strategy as explained above. This ensures that the test which Charlie performs only accepts proofs that actually could be generated in an unbalanced state.
3. there is a reachable state in which Charlie accepts a proof from the adversary (where Charlie is using the same strategy as mentioned above).

Note that the verifier does not have any strategic decisions, we are only interested in knowing whether his knowledge suffices to determine Charlie's decision. Also note that the formula makes use of the fact that Charlie "commits" to a fixed strategy in a central way: The formula φ^{ver} states that if Charlie plays his fixed strategy, and the verifier can rely on this, then the verifier knows Charlie's state. For our decidability result it is therefore essential that QAPI provides a mechanism to express that Charlie continues the strategy referred to with S_{C}.

We now prove that a protocol is not abuse-free if and only if the protocol modifies as mentioned above satisfies the formula above.

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 Charlie, 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 Charlie's help to achieve this.[9] After this, the adversary sends a message satisfying the test θ to Charlie.
– Charlie'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 Charlie'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.
– The strategies instantiating $\mathrm{S}_1^{\mathcal{A}}$ and $\mathrm{S}_1^{\mathcal{A}}$ 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.[10]

We claim that this choice of strategies satisfies the formula.

1. the first conjunct is satisfied because by definition, the question whether Charlie 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, Charlie and the verifier have access to the same secret keys, hence they derive the same knowledge from the message (recall that Charlie's nonces do not appear in the protocol run, since Charlie does not construct messages on his own, but only forwards a message received from the adversary).
2. the second conjunct requires that if Charlie accepts at the end of the protocol run, and the strategy played by the adversary is greedy, then the state directly before Charlie 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 Charlie's strategy, he moves into an accepting state.

Hence if the protocol is abusive, then the formula is indeed satisfied in the initial state of the protocol.

[9] 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.

[10] See Section 5.2.3 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 decide on the strategies to apply here.

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. Note that since 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 Charlie's decision only depends on the outcome of a test which the verifier can perform on the message received by Charlie, which is (by construction of Charlie) the first message that Charlie 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 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 Charlie, and then letting the adversary deliver a message satisfying θ to Charlie. Since the message satisfies θ, Charlie 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 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 Charlie receives the first message from the adversary. i.e., the state q, is unbalanced. This concludes the proof. □

Again, note that our proof heavily relies on QAPI's ability to directly refer to strategies in the formula themselves, since this allowed us to express that the verifier knows Charlie's decision—this is only possible if the verifier can rely on the strategy which Charlie players. Another, related, way to achieve a similar effect is to use variants of ATL that use commitment and include a mechanism for players to know that others have committed to a certain strategy.

5.2.3 On variations of abuse-freeness

We note that different notions of abuse-freeness can also be captured in our model. As mentioned in the previous section, 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.

If we assume that Charlie 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 Charlie 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 adversial strategies mentioned in φ^{unbal}) simply states that there is a reachable protocol state in which Charlie knows that the previous state was unbalanced (he cannot know that the *current* state is unbalanced because our model is concurrent and Charlie does not know actions occurring at the same time as the adversary presenting his proof to Charlie).

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 Charlie that he is in state q_1, not in state q_2, which is only possible if Charlie has some knowledge the adversary does not have. Hence for a passive Charlie, 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, if one wants 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 and not in a quantifier prefix), this can easily be achieved by using strategy choices instead of strategies which can be used to allow strategies to depend on the state[11], 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. These issues also have been discussed in various papers on epistemic strategic logics, see for example [JvdH04].

5.3 Anonymous Broadcast

A classic example for anonymous broadcast is the dining cryptographers problem: A group of at least 3 cryptographers have dinner, and discover that someone has paid for it. They want to find out whether it was one of them or an outside party. However in the former case the identity of the payer should not be revealed.

To achieve this, every pair of adjacent cryptographers first agrees on a random bit, then everyone announces the exclusive-or of the two random bits shared with his neighbors and his own secret bit (which is 1 if he payed for

[11]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.

the dinner). The exclusive-or of all publically revealed bits is exactly the disjunction of the secret bits.

We treat the problem as follows: For each cryptographer we introduce a role following the mentioned protocol. An additional party nondeterministically distributes the private bits, where at most one of the bits is 1. At the end of the protocol every cryptographer knows the disjunction of the bits, but does not know, for any strict subset of the remaining cryptographers, that one of their bits is 1 (obviously if the disjunction is 0 everyone knows that all inputs are 0). This allows anonymous broadcast of a single bit, generalizations allow multiple messages and senders [Cha88]. In addition to an analysis of correctness and anonymity properties similarly as [ABvdM10], our model also covers the case of active adversaries. A detailed treatment is out of the scope of this paper.

6 Proof of the Main Result

In this section, we prove the main result, Theorem 4.1. 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 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.

A usual 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, it will establish that there is a finite structure—later called ${}^{\mathcal{C}_{Pr}}/_{\equiv_{\text{fin}}}$—and a probabilistic bisimulation between this one and $\mathcal{C}_{Pr}$. The construction of ${}^{\mathcal{C}_{Pr}}/_{\equiv_{\text{fin}}}$ 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 ${}^{\mathcal{C}_{Pr}}/_{\equiv_{\text{fin}}}$.

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 $\mathcal{C}_{Pr}/_{\equiv_{\text{fin}}}$ such that there is a relation Z which is a probabilistic bisimulation between $\mathcal{C}_{Pr}$ and $\mathcal{C}_{Pr}/_{\equiv_{\text{fin}}}$ for every coalition, and the initial states of $\mathcal{C}_{Pr}$ and $\mathcal{C}_{Pr}/_{\equiv_{\text{fin}}}$ are identical.*

6.2 Some notation

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 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

- $\text{ar}(\Sigma^{\mathsf{t}})$ 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 and *depth* (E),
- if q is not an initial state, then $pred(q)$ denotes the unique predecessor state of q in $\mathcal{C}_{Pr}$,
- $prvst(q)$ denotes the number of steps needed to reach q in a protocol run, i.e., if q is an initial state then $prvst(q) = 0$, and otherwise $prvst(q) = prvst(pred(q)) + 1$.

6.3 Defining State-equivalence

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 $p = (i, path)$, then

- p is a *position*,
- $u \downarrow path$ 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 $u \downarrow path = \texttt{error}$ (where `error` is a special symbol not used anywhere else),
- $u(path)$ is the label of the root node of $u \downarrow path$ (where the label of `error` is `error`),
- $|t| = n$,
- if $path_2$ is a path, then $p \circ path_2$ is defined as the position $(i, path_1 path_2)$ (in this case we say that $p \circ path_2$ is a *suffix* of p, and p is a *prefix* of $p \circ path_2$),
- $t{\downarrow}p = t_i{\downarrow}path$ and $t(p) = t_i(path)$ (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 $|t^1| = |t^2|$ (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, the same set of identities is corrupted, 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 $\mathsf{eqdeg}(.)$. More precisely, 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). Our definition of the probabilistic bisimulation will be based on this idea: States are Z-related if the local states of the principals are identical, and equivalence of terms holds down to the specified degree. 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 can now formally define the function $\mathsf{eqdeg}\,(.)$ as explained above:

Definition For a k-protocol Pr with length ℓ and a number d, let

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

6.4 Move Transfer For Honest Principals

We now show that honest principals have essentially "the same options" in $\mathcal{C}_{Pr}/_{\equiv}$ as they have in $\mathcal{C}_{Pr}$, 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 shows 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 t be a term, and let $r = t[x/\mathcal{M}]$. Then there are a term s and positions $q_1, \ldots, q_n$ such that for all relevant i,*

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

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

Proof. This lemma trivially follows from the observation that when evaluating the term $t[x/\mathcal{M}]$, each application of a rule from the equational theory E only results in a reference with depth *depth* (E) into $\mathcal{M}$. Hence the term s is obtained from t by replacing an operator that results in a reference to $\mathcal{M}$ with a variable, and the position corresponding to the variable is simply the position of the referenced term. Since the nesting degree of these applications is restricted by the maximal depth of the term t, and due to the above, each application only results in an increase of the depth of reference by *depth* (E), the result follows. □

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 pasring sequence. The move of the adversary will be covered in the following Section 6.5.

Lemma 6.5 *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be term sequences, let $m_1, \ldots, m_m$ be terms. Further, let d_1, d_2, and d_3 be natural numbers such that*

- *for all i, we have that depth $(m_i) \leq d_3$ and depth $(m_i) \cdot$ depth $(\mathrm{E}) \leq d_2$,*
- *$\mathcal{M}_1 \sim_{d_1} \mathcal{M}_2$.*

For $a \in \{1, 2\}$, let $\mathcal{M}_a'$ be obtained from $\mathcal{M}_a$ by adding the terms $m_1[x/\mathcal{M}_a]$, $\ldots$, $m_k[x/\mathcal{M}_a]$. Then $\mathcal{M}_1' \sim_d \mathcal{M}_2'$, where $d = d_1 - d_2 - d_3$.

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 (which they can do, due to Proposition 6.6, see below), 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 $depth(E) \cdot \max\{depth(t_j) \mid 1 \leq j \leq m\} \leq d_2$, the depth of each s_i is at most $depth(t_i) \leq d_3$, and $\mathcal{M}'_a$ is obtained from $\mathcal{M}_a$ by adding the terms $s_1[x_1/\mathcal{M}{\downarrow}q_1, \ldots, x_n/\mathcal{M}{\downarrow}q_n], \ldots, s_m[x_1/\mathcal{M}{\downarrow}q_1, \ldots, x_n/\mathcal{M}{\downarrow}q_n]$.

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

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 $|\mathcal{M}_2|$), 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 $s^a_{j_i} = t^a_i$ for all relevant i, it suffices to prove that $u^1 = (s^1_1, \ldots, s^1_m)$ and $u^2 = (s^2_1, \ldots, s^2_m)$ are $(d_1 - d_2 - d_3)$-equivalent (since the original terms from the sequences $\mathcal{M}_1$ and $\mathcal{M}_2$ appear in this list, this implies the claim of the lemma).

Hence let p_1 and p_2 be positions, where for $b \in \{1, 2\}$, we have $p_b = (i_b, path_b)$, and $|path_b| \leq d_1 - d_2 - d_3$. Without loss of generality, we can assume that $i_b = b$.

We first show that $u^1(p_1) = u^2(p_1)$ (note that in this proof, we only use the fact that $depth(p_1) \leq d_1 - d_2$—we will refer to this slightly stronger result in the second part of the proof). By construction, (since $p_1 = (1, path_1)$), we have $u^a(p_1) = s^a_1(path_1)$. If $path_1$ does not visit a position in s_1 which is a variable, then obviously $s^a_1(path_1) = s_1(path_1)$, and it follows that $u^1(p_1) = s^1_1(path_1) = s_1(path_1) = s^2_1(path_1) = u^2(p_1)$ as required. Now assume that when following $path_1$ in s_1, we encounter a variable, without loss of generality the variable x_1. Let $path_1 = w_1 w_2$, such that $s_1(w_1) = x_1$. It then follows that

for $a = 1, 2$, $u^a(p_1) = s_1^a(path_1) = s_1^a(w_1w_2) = (s_1^a \downarrow w_1)(w_2) = x_1[x_1/(u^a \downarrow q_1), \ldots](w_2) = (u^a \downarrow q_1)(w_2) = u^a(q_1 \circ w_2) = t_1^a(qpath_1 w_2)$. Since $|w_2| \leq depth\,(p_1) \leq d_1 - d_2$, and $depth\,(q_1) \leq d_2$, it follows that $depth\,(q_1) + |w_2| \leq d_1$. Hence we know (since $t^1 \sim_{d_1} t^2$, that $u^1(q_1 \circ w_2) = u^2(q_1 \circ w_2)$, and it follows that $u^1(p_1) = u^1(q_1 \circ w_2) = u^2(q_1 \circ w_2) = u^2(p_1)$, as required.

We now show that $u^1 \downarrow p_1 = u^1 \downarrow p_2$ if and only if $u^2 \downarrow p_1 = u^2 \downarrow p_2$. Due to symmetry, it obviously suffices to prove one direction. Hence assume that $u^1 \downarrow p_1 = u^1 \downarrow p_2$. We show the claim by induction over the depth restrictions for the p_b, the q_i, 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 $depth\,(p_b) \leq \alpha$ for $b \in \{1, 2\}$ and $depth\,(s_b) \leq \gamma$ for $b \in \{0, 1\}$, 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 $d_1 - d_2 - d_3$, and terms s_i with $depth\,(s_i) \leq d_3$).

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 $depth\,(p_b) \leq d_2$ 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 $q_b = (b, qpath_b)$. Now if s_b is the variable x_b, then we have (for $a \in \{1, 2\}$):

$$\begin{aligned} u^a \downarrow p_b &= s_b^a \downarrow path_b \\ &= (x_b[\ldots, x_b/(t^a \downarrow q_b)\ldots]) \downarrow path_b \\ &= (t^a \downarrow q_b) \downarrow path_b \\ &= (t^a \downarrow (b, qpath_b)) \downarrow path_b \\ &= (t_b^a \downarrow qpath_b) \downarrow path_b \\ &= t_b^a \downarrow (qpath_b \circ path_b). \end{aligned}$$

If s_b is the constant $cons_b$, then we have (for $a \in \{1, 2\}$): $u^a \downarrow p_b = s_b^a \downarrow path_b = cons_b \downarrow path_b$. 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 $t_1^1 \downarrow (qpath_1 \circ path_1) = u^1 \downarrow p_1 = u^1 \downarrow p_2 = t_2^1 \downarrow (qpath_2 \circ path_2)$. Since $|qpath_b| + |path_b| \leq d_2 + d_1 - d + 2 = d_1$, and $t^1 \sim_{d_1} t^2$, this implies $t_1^2 \downarrow (qpath_1 \circ path_1) = t_2^2 \downarrow (qpath_2 \circ path_2)$. Due to the above (and since both s_b are variables), we therefore have $u^2 \downarrow p_1 = t_1^2 \downarrow (qpath_1 \circ path_1) = t_2^2 \downarrow (qpath_2 \circ path_2) = u^2 \downarrow p_2$, as required.

Assume that both s_1 and s_2 are constants, i.e., $s_1 = cons_1$, and $s_2 = cons_2$. Then $cons_1 \downarrow path_1 = u^1 \downarrow p_1 = u^1 \downarrow p_2 = cons_2 \downarrow path_2$. Hence it follows that

$u^2 \downarrow p_1 = cons_1 \downarrow path_1 = cons_2 \downarrow path_2 = u^2 \downarrow path_2$ 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 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 $t^1 \sim_{d_1} t^2$, 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, $depth(s_1) > 0$ (since the case where both depths are 0 is obviously 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 $t^a \downarrow q_j$). 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 $\mathit{depth}(p_1), \mathit{depth}(p_2) \leq \alpha - 1$, 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 $\mathit{depth}(s_1) = \mathit{depth}(s_2) = \gamma + 1$*,* 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 constructor from Σ^{t}, and $s_{b,i}$ are terms with $\mathit{depth}(s_{b,i}) \leq \gamma$. Analogously to the s_b^a, for $a, b \in \{1, 2\}$, and $i \leq e_b$, we define $s_{b,i}^a$ to be the term obtained from $s_{b,i}$ by replacing every occurrence of a variable x_j with the term $u^a \downarrow q_j$. Now observe that if $\mathit{path}_b = \epsilon$, then $u^a \downarrow p_b = s_b^a \downarrow \epsilon = s_b^a$, and for a step c, we have that $u^a \downarrow (p_b \circ c) = (u^a \downarrow p_b) \downarrow c = s_b^a \downarrow c = s_{b,c}^a$.

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

Assume that $\mathit{depth}(s_1) = \mathit{depth}(s_2) = \gamma + 1$ *and* $\mathit{path}_1 = \mathit{path}_2 = \epsilon$*.* 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 $\mathit{path}_1 = \mathit{path}_2 = \epsilon$, we know that $u^a \downarrow (p_b \circ c) = s_{b,c}^a$. Hence the involved positions can be described with terms s_1', s_2' with $\mathit{depth}(s_1'), \mathit{depth}(s_2') \leq \gamma$, and positions p_1', p_2' with $\mathit{depth}(p_1'), \mathit{depth}(p_2') = 0 \leq \alpha$. 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 $\mathit{depth}(s_1) = \mathit{depth}(s_2) = \gamma + 1$*, one* path_b *is empty, the other is not.* Without loss of generality, assume that $\mathit{path}_1 = \epsilon$, and $\mathit{path}_2 = c_2 \mathit{path}_2'$. Then we know that $u^a(p_1) = s_1^a(\epsilon) = f_1$. 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 $u^a \downarrow (p_1 \circ c) = s_{1,c}^a$, and $u^a \downarrow p_2 \circ c = s_{2,c_2}^a \downarrow (\mathit{path}_2' \circ c)$. 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 $(2, \mathit{path}_2' c)$, which have depth $1 \leq \alpha$ and $\alpha - 1 \leq \alpha$ (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. The result directly follows from Lemma 6.4: By definition of d_{Pr}, the application of a test used in a protocol rule only accesses and compares elements with depth at most d_{Pr}. Hence, the terms compared by the test are identical in q_1 iff they are in q_2 due to the above Lemma. Note that the proposition trivially holds in final states of the protocol as here honest principals only have dummy moves available. □

6.5 Move Transfer for the Adversary

We now show the analogous result of Section 6.4 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 the term substituted for x, i.e., the maximal (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 Let q be a state of $\mathcal{C}_{Pr}$, and let p be a position. We say that p is $\mathcal{A}$-*accessible in* q if

- $\mathit{depth}\,(p) \leq d_{Pr} \cdot \mathit{prvst}(q)$,
- there is an $\mathcal{A}$-term $t_{\mathcal{A}}$ with extraction-depth at most d_{Pr} such that for all states q' obtained from q by replacing $q{\downarrow}p$ with a new term t', we have that $t_{\mathcal{A}}[x/\mathit{terms}(q')] = t'$.

Intuitively, the last point of the above definition expresses 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 $t_{\mathcal{A}}$ does not follow from the fact that $\mathit{depth}\,(p) \leq d_{Pr}$: 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).

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.*

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, and also from the definition it follows that for any 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 $t[x/\mathcal{M}_{\mathcal{A}}]$, 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.

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 $t_{\mathcal{A}}[x_1/t_1, \ldots, x_n/t_n]$ 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 is a position p such that p is $\mathcal{A}$-accessible in q, and $q\!\downarrow p = t$.*

Proof. By choice of t, the term was constructed by a 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' (without loss of generality, we can assume that q' is not an initial state of the protocol). If t does not appear as a subterm in q'', the claim follows, since then t is computed by a principal in the step from q'' to q', and results from principal computations appear with depth at most d_{Pr}. Additionally, since the adversary can construct t in q', this position must be $\mathcal{A}$-accessible.

Hence assume that t appears as a subterm of q'', and that no new copy of the term is generated in the transition from q'' to q' in an $\mathcal{A}$-accessiple position (if this were true, the above case would apply). 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 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 of principal-performed computations, and thus is restricted

in depth by $d_{Pr} \cdot prvst(q'')$. The result of the 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 $prvst(q') \leq prvst(q)$.

Note that the above argument also inductively covers the case when the term uses more than one extraction (for example when accessing a symmetric key which is then used to decrypt another message): These extractions appear in parallel and do not refer to the same position. In the example, if a key appears at a certain depth, this depth does not add to the depth of the decrypted message. Formally, the extraction depth of the resulting term is still bounded by d_{Pr} if every single extraction is. □

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 $d' = \mathsf{eqdeg}(q')$, 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*

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

(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 resulting move transfer function can be constructed such that move uniformity for information degrees 1, 2, and 3 is satisfied.

Proof. Note that by definition of $\sim_d$, the same set of identities C is corrupted in q_1 and q_2. With d, we denote $\mathsf{eqdeg}(q^1) = \mathsf{eqdeg}(q^2)$. From the definition of $\equiv$ and $\mathsf{eqdeg}(.)$, it follows that $2d' + 2mdagdpth_{\mathcal{A}}(d') \leq d$.

Let $m^1 = [[m^1_{\mathcal{A}}[x/\mathcal{M}^1_{\mathcal{A}}]]]$ 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 $t_1, \ldots, t_k$. For a position $p = (i, path)$, with $m^1_{\mathrm{DAG}} \rightarrow p$ we denote the vertex in m^1_{DAG} obtained when following the path $i \circ path$ from *root*, and with $m^1_{\mathrm{DAG}} \downarrow p$, we denote the subterm represented by $m_{\mathrm{DAG}} \rightarrow p$ (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, let $dagdepth(p)$ be the length of a longest path from *root* to $m^1_{\mathrm{DAG}} \to p$ (which may be longer than the path induced by p). From m^1_{DAG}, we obtain m_{DAG} as follows:

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 $dagdepth(p) > mdagdpth_{\mathcal{A}}$, if $m_{\mathrm{DAG}} \to p$ still exists, insert a new adversary nonce into $m_{\mathrm{DAG}} \to p$.*

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. Also note that if $m_{\mathrm{DAG}} \to p$ contains a marker $(r \to path)$, then $|path| \leq dagdepth(p)$. The construction also implies that if $m_{\mathrm{DAG}} \to p$ contains a marker $(r \to path)$, then $m^1 \downarrow p = q^1 \downarrow (r \circ path)$. We prove a few features of the construction for later reference.

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. Now let p^0 be the representative chosen in step i. 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. 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 chosen in step i 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 pairwise incomparable with respect to prefix ordering. 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 $mdagdpth_{\mathcal{A}}(d')$. □

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

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 marker or a newly introduced adversary nonce. In particular, this implies $dagdepth(p) \leq mdagdpth_{\mathcal{A}}(d')$. Hence due to construction, $|path| \leq dagdepth(p) \leq mdagdpth_{\mathcal{A}}(d')$. □

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 $m^2 = (m^2_{\mathrm{DAG}}\downarrow 1, \ldots, m^2_{\mathrm{DAG}}\downarrow k)$. 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 $m^2_{\mathcal{A}}$. 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 exists.

Fact 3 *Let p be a position such that $depth\,(p) \leq mdagdpth_{\mathcal{A}}(d')$. Then $m^1(p) = m^2(p)$.*

Proof. First assume that there is a minimal prefix p_r or 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 $|path| \leq mdagdpth_{\mathcal{A}}(d')$, and from the construction, we know $depth\,(r) \leq d'$. It therefore follows that $m^a(p) = (m^a \downarrow p_r)(w) = (q^a \downarrow (r \circ path))(w) = q^a(r \circ path \circ w)$. We also know that $|w| \leq depth\,(p) \leq mdagdpth_{\mathcal{A}}(d')$. It therefore follows that $depth\,(r \circ path \circ w) = depth\,(r) + |path| + |w| \leq d' + mdagdpth_{\mathcal{A}}(d') + mdagdpth_{\mathcal{A}}(d') \leq d$, 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 containing a marker. Since $dagdepth(p) \leq mdagdpth_{\mathcal{A}}(d')$, there is also no prefix containing a newly introduced adversary nonce, and thus $m^1(p) = m_{\mathrm{DAG}}(p) = m^2(p)$ as required. □

We now prove $s_1 := terms(q^1, m^1) \sim_{d'} terms(q^2, m^2) =: s_2$. 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 and the fact that $d' \leq mdagdpth_{\mathcal{A}}(d')$.

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 $q^1 \sim_d q^2$.

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}. Without loss of generality assume 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 $m^1_{\mathrm{DAG}} \to p_2$, 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 $m_{\text{DAG}} \to p'_i$ contains a marker $(r_i \to path_i)$. Due to Fact 2, $|path_i| \leq mdagdpth_{\mathcal{A}}(d')$. It follows that

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

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 $depth\,(r_i \circ path_i \circ w_i) = depth\,(r_i) + |path_i| + |w_i| \leq d' + mdagdpth_{\mathcal{A}}(d') + d' \leq d$. 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 $m_{\text{DAG}} \to p_r$ 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, $|path| \leq mdagdpth_{\mathcal{A}}(d')$. Note that

$$\begin{array}{rcl@{\qquad}rcl} m^a{\downarrow}p_1 & = & m^a_{\text{DAG}}{\downarrow}(p_r \circ w) & = & (m^a_{\text{DAG}}{\downarrow}p_r){\downarrow}w \\ & = & (q^a{\downarrow}(r \circ path)){\downarrow}w & = & q^a{\downarrow}(r \circ path \circ w). \end{array}$$

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 $depth\,(r \circ path \circ w) = depth\,(r) + |path| + |w| \leq d' + mdagdpth_{\mathcal{A}}(d') + d' \leq d$, 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.

We now show that if $depth\,(p_1)$, $depth\,(p_2) \leq d'$, then $s^2{\downarrow}p_1 = s^2{\downarrow}p_2$ implies $s^1{\downarrow}p_1 = s^1{\downarrow}p_2$. This is trivial if both positions refer into positions from 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 $depth\,(p_2) \leq 2 \cdot d' + mdagdpth_{\mathcal{A}}(d')$ (we will use this stronger result in the sequel). Note that no suffix 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}.

First assume there is a prefix p' of $p_1 \circ w$ such that $p' \circ w' = p_1 \circ w$, and $m_{\text{DAG}} \to p'$ contains the marker $(r \to path)$. From Fact 2, it follows

that $|path| \leq mdagdpth_{\mathcal{A}}(d')$. Since p_1 and p' have a common suffix, they must be prefixes of each other. First assume p' is a prefix of p_1, i.e., there is some w'' such that $p_1 = p' \circ w''$. Then $|w''| \leq depth\,(p_1) \leq d'$. It follows that $m^a \downarrow p_1 = m^a \downarrow (p' \circ w'') = (m^a \downarrow p') \downarrow w'' = (q^a \downarrow (r \circ path)) \downarrow w'' = q^a \downarrow (r \circ path \circ w'')$. In particular, $q^2 \downarrow p_2 = m^2 \downarrow p_1 = q^2 \downarrow (r \circ path \circ w'')$. Since $depth\,(p_2) \leq 2 \cdot d' + mdagdpth_{\mathcal{A}}(d') \leq d$, and $depth\,(r \circ path \circ w'') = depth\,(r) + |path| + |w''| \leq d' + mdagdpth_{\mathcal{A}}(d') + d' \leq d$, the d-equality of q^1 and q^2 implies that $q^1 \downarrow p_2 = q^1 \downarrow (r \circ path \circ w'')$. Hence we obtain $m^1(p_1 \circ w) = (m^1 \downarrow p_1)(w) = (q^1 \downarrow (r \circ path \circ w''))(w) = (q^1 \downarrow p_2)(w) = q^1(p_2 \circ w)$, a contradiction. Now assume p_1 is a prefix of p', and let $p' = p_1 \circ w''$. Since p' contains a marker, due to Fact 2, it follows that $depth\,(d') \leq dagdepth(p') \leq mdagdpth_{\mathcal{A}}(d')$. We have $p_1 \circ w = p' \circ w' = p_1 \circ w'' \circ w'$, and hence $w = w'' \circ w'$. We know that $q^2 \downarrow (p_2 \circ w'') = m^2 \downarrow (p_1 \circ w'') = m^2 \downarrow p' = q^2 \downarrow (r \circ path)$. Note that $depth\,(p_2 \circ w'') = depth\,(p_2) + |w''| \leq 2 \cdot d' + mdagdpth_{\mathcal{A}}(d') + depth\,(p') \leq 2 \cdot d' + 2 \cdot mdagdpth_{\mathcal{A}}(d') \leq d$, and $depth\,(r \circ path) = depth\,(r) + |path| \leq d' + mdagdpth_{\mathcal{A}}(d') \leq d$. Therefore the above equality and the d-equivalence of q^1 and q^2 implies $q^1 \downarrow (p_2 \circ w'') = q^1 \downarrow (r \circ path)$. It therefore follows that $m^1 \downarrow (p_1 \circ w) = m^1 \downarrow (p' \circ w') = (m^1 \downarrow p') \downarrow w' = (q^1 \downarrow (r \circ path)) \downarrow w' = (q^1 \downarrow (p_2 \circ w'')) \downarrow w' = q^1 \downarrow (p_2 \circ w'' \circ w') = q^1 \downarrow (p_2 \circ w)$, a contradiction. Therefore there is no prefix of $p_1 \circ w$ referring to a marker. Since there is also no suffix of p_1 referring to a new adversary nonce, it follows that $m^a(p_1 \circ w) = m_{\text{DAG}}(p_1 \circ w)$. Since no prefix of $p_1 \circ w$ contains a new adversary nonce, we know that $depth\,(p_1 \circ w) \leq dagdepth(p_1 \circ w) \leq mdagdpth_{\mathcal{A}}(d') \leq d$. 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 $m_{\text{DAG}} \rightarrow p_1 = m_{\text{DAG}} \rightarrow p_2$, 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 $(r \rightarrow path)$, i.e., $p_1 = p_r \circ w$ for some w with $|w| \leq depth\,(p_1) \leq d'$. Due to Fact 2, $|path| \leq mdagdpth_{\mathcal{A}}(d')$. It follows that $m^a \downarrow p_1 = m^a \downarrow (p_r \circ w) = (m^a \downarrow p_r) \downarrow w = (q^a \downarrow (r \circ path)) \downarrow w = q^a \downarrow (r \circ path \circ w)$, and hence $m^2 \downarrow p_2 = m^2 \downarrow p_1 = q^2 \downarrow (r \circ path \circ w)$. Since $depth\,(r \circ path \circ w) = depth\,(r) + |path| + |w| \leq d' + mdagdpth_{\mathcal{A}}(d') + d'$, 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 $2 \cdot d' + mdagdpth_{\mathcal{A}}(d')$) 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.

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 without loss of generality there is a prefix p_r

of $p_1 \circ w$ such that $p_1 \circ w = p_r \circ w'$, and $m_{\mathrm{DAG}} \to p_r$ contains the marker $(r \to path)$. From Fact 2, we know that $dagdepth(p_r), |path| \leq mdagdpth_{\mathcal{A}}(d')$. Since no prefix of p_1 contains a marker, and there is a common suffix of p_1 and p_r, p_r must be a suffix of p_1, i.e., there is a path w'' such that $p_r = p_1 \circ w''$. It thus follows that $p_1 \circ w = p_r \circ w' = p_1 \circ w'' \circ w'$, i.e., $w = w'' \circ w'$. Since $dagdepth(p_r) \leq mdagdpth_{\mathcal{A}}(d')$, it follows that $|w''| \leq mdagdpth_{\mathcal{A}}(d')$. Obviously, we have $m_{\mathrm{DAG}} \to (p_1 \circ w) \neq m_{\mathrm{DAG}} \to (p_2 \circ w)$, 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). Due to construction of m_{DAG}, this implies that $depth\,(p_2 \circ w) \leq dagdepth(p_2 \circ w) \leq mdagdpth_{\mathcal{A}}(d')$, and hence in particular, $|w'| \leq mdagdpth_{\mathcal{A}}(d')$. Hence $depth\,(r \circ path \circ w') = depth\,(r) + |path| + |w'| \leq d' + mdagdpth_{\mathcal{A}}(d') + mdagdpth_{\mathcal{A}}(d') \leq d$. Fact 3 implies that $m^1(p_2 \circ w) = m^2(p_2 \circ w)$, and hence

$m^1(p_2 \circ w) = m^2(p_2 \circ w) = m^2(p_1 \circ w) = m^2(p_r \circ w') = q^2(r \circ path \circ w') = q^1(r \circ path \circ w') = m^1(p_r \circ w') = m^1(p_1 \circ w)$, a contradiction.

Therefore, no prefix of $p_1 \circ w$ or $p_2 \circ w$ contains a marker. Since $m_{\mathrm{DAG}} \to (p_1 \circ w) \neq m_{\mathrm{DAG}} \to (p_2 \circ w)$, if a prefix of wlog $p_1 \circ w$ contains a newly introduced adversary nonce, then $m^2(p_1 \circ w) \neq m^2(p_2 \circ w)$ follows, as different positions in m^2_{DAG} contain different new nonces—a contradiction. It therefore follows that $m^1(p_1 \circ w) = m_{\mathrm{DAG}} \downarrow p_1 \circ w = m^2(p_1 \circ w) = m^2(p_2 \circ w) = m_{\mathrm{DAG}}(p_2 \circ w) = m^1(p_2 \circ w)$, a contradiction. Hence $m^1 \downarrow p_1 = m^2 \downarrow p_2$ as claimed.

To make the construction unique, we demand that the new nonces that are introduced are well-determined in the sense 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 $\mathcal{C}_{Pr}$. Note that this does not introduce additional equalities, since different terms are replaced with different nonces, and the introduced nonces are still fresh since f is injective.

It remains to show that there is an adversary move that results in the message m^2 being sent. For this, it is obviously sufficient to prove that for all positions p 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, the term

$q^2 \downarrow (r \circ path)$ can be constructed by the adversary in q^2. We make a case distinction:

1. Assume that $q^2 \downarrow (r \circ path)$ 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.
2. Assume that $q^2 \downarrow (r \circ path)$ 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, and due to Proposition 6.7, p' is $\mathcal{A}$-accessible in q^2 as well (the requirement for d is obviously met). Since $depth\,(r \circ path) \leq d' + mdagdpth_{\mathcal{A}}(d') \leq d$, the d-equivalence of q^1 and q^2 and $q^1 \downarrow (r \circ path) = q^1 \downarrow p'$ 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.
3. Assume that $q^2 \downarrow (r \circ path)$ is a term that the adversary can compute himself, but that has been computed by a principal. Since terms computed by principals alone (i.e., without adversary input) have depth at most $prvst(q^1) \cdot d_{Pr}$, there is some path $path'$ with $|path'| \leq prvst(q^1) \cdot d_{Pr}$ such that $q^2 \downarrow (r \circ path \circ path')$ is of one of the above cases, and the result follows analogously (note that for the above two cases, the requirements for d are still met after adding $prvst(q^1) \cdot d_{Pr}$).

It remains to show that the adversary move resulting in the message sequence m^2 being sent can be computed from the original move $m^1_{\mathcal{A}}$ with only the knowledge available to the adversary in information degree 3 in the state q^1 (obviously the result for information degree 1 and 2 follows). Obviously, the message m^1 can be computed given the state information and the adversary move, and the above case distinction can be performed by the adversary—the only non-trivial aspect is finding the position p', this can be achieved by simply comparing the subterm of m^1 to subterms appearing in $\mathcal{A}$-accessible positions in q^1. Note that due to the above, every marker introduced leads to a position that is $\mathcal{A}$-accessible by the adversary, hence the computation of m^2 using the markers as above can be performed with the knowledge available to the adversary.

Finally, note that it is not necessary for the adversary to compute the entire equivalence relation $\sim$: For the adversary, it is sufficient to know the depth in which new nonces have to be introduced (this depth only depends on the number of steps so far in the protocol run), and to know that terms obtained from principals have to be used in the same positions. □

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 the reason for demanding $mdagdpth_{\mathcal{A}}(d')$-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

- $depth\,(r) = d'$,
- $depth\,(p_i) = 0$ 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 protocols, the involved strategies are usually much simpler. Hence we prefer to prove the bounds as stated in the proof, and leave the proof itself relatively simple.

6.6 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 properties. ${}^{\mathcal{C}_{Pr}}\!/_{\equiv}$ is constructed by simply allowing the adversary to use only the moves that result in applying the construction of Lemma 6.9:

Definition Let Pr be a protocol. Then ${}^{\mathcal{C}_{Pr}}\!/_{\equiv}$ is the induced CGS obtained from $\mathcal{C}_{Pr}$ by restricting the state space to the states which can be reached by the adversary only using moves that appear as the result of the construction

in Lemma 6.9. We call $\mathcal{C}_{Pr}/_{\equiv}$ the *strategy representation of* Pr.

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. Hence $\mathcal{C}_{Pr}/_{\equiv}$ has a finite representation:

Proposition 6.10 *There is a finite CGS* $\mathcal{C}_{Pr}/_{\equiv_{\text{fin}}}$ *that is finite and there is a probabilistic bisimulation between* $\mathcal{C}_{Pr}/_{\equiv}$ *and* $\mathcal{C}_{Pr}/_{\equiv_{\text{fin}}}$.

6.7 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 reachable 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 randomization. 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 $=$ denote the relation where two states $q_1 \in Q_1$ and $q_2 \in Q_2$ are 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 corresponds to the injection function from $\mathcal{C}_{Pr}/_{\equiv}$ to $\mathcal{C}_{Pr}$, which strips off these prefixes.

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

Proof. Obviously, $Z^{-1} \circ =^{-1}$ and $=^{-1} \circ Z^{-1}$ 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 is a probabilistic uniform strong alternating simulation. We first treat the case $=$, i.e., 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 $i\colon {}^{\mathcal{C}_{Pr}}\!/_{\equiv} \to \mathcal{C}_{Pr}$ with $i(q) = q$, except for removal of the prefixes of adversary nonce names. 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, 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 $i(q_2)$ and $i(q_2')$, 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 properties.

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 $\delta_{\dots}^{1\to 2}$ 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.

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: Applying the move resulting from these constructions directly results in $\equiv$-equivalent states. For the honest principals, instantiate d_1 with $\mathsf{eqdeg}\,(q_1)$, d_2 and d_3 with d_{Pr}. Note that the move of the adversary and the principals are performed in parallel and therefore one cannot use terms obtained as the result of the other. By construction, the same local states of honest principals are reached with the same probability, and the terms in the resulting states are equivalent to the degree required in the successor states of q_1 and q_2 due to the mentioned Lemmas.

Uniformity Let q_2 and q_2' be states such that for some player $a \in \{1, \dots, k, \mathcal{A}\}$, we have that $q_2 \sim_{\mathsf{eq}_i(a)} q_2'$. Then this indistinguishability also holds for $Z^{-1}(q_2)$ and $Z^{-1}(q_2')$, since by construction these

are obtained from q_2 and q_2' by consistent replacement of terms with adversary nonces.

Knowledge Transfer Let $q_1' \sim_{\mathsf{eq}_i(A)} q_1$ for $q_1, q_2 \in Q_1$. Let q_2' be the state obtained from q_1' by replacing the adversary nonces introduced by the construction from Lemma 6.9 with the original terms (since unique nonces were introduced for each term, this construction is well-defined). Obviously, $q_2' \equiv q_1'$, and $q_2' \sim_{\mathsf{eq}_i(A)} q_2$, since the same terms correspond to the same nonces in the translation $Z^{-1}(q_2)$ and $Z^{-1}(q_2')$.

□

Decidability now follows, since using transitivity, we know that there is a probabilistic bisimulation between $\mathcal{C}_{Pr}$ and ${}^{\mathcal{C}_{Pr}}/_{\equiv \mathrm{fin}}$, and the latter by definition is a finite CGS. Also, from the construction it is obvious that ${}^{\mathcal{C}_{Pr}}/_{\equiv \mathrm{fin}}$ can be computed on input Pr: Since the length of every term appearing in ${}^{\mathcal{C}_{Pr}}/_{\equiv \mathrm{fin}}$ is bounded by a constant, the set of possible adversary moves can be determined by brute-force, applying all possible terms to construct messages up to this bound. Further, the initial states of $\mathcal{C}_{Pr}$ and ${}^{\mathcal{C}_{Pr}}/_{\equiv \mathrm{fin}}$ are identical; hence Theorem 6.2 and thus Theorem 4.1 are proven, since decidability in the finite model ${}^{\mathcal{C}_{Pr}}/_{\equiv \mathrm{fin}}$ follows from the results in [Sch10a].

6.8 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 ${}^{\mathcal{C}_{Pr}}/_{\equiv \mathrm{fin}}$ 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. Interesting open questions are a complexity analysis of the decision problem, and whether security in our model transfers to a computational setting.

References

[ABvdM10] Omar I. Al-Bataineh and Ron van der Meyden. Epistemic model checking for knowledge-based program implementation: An application to anonymous broadcast. In *SecureComm 2010*. Springer, 2010.

[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.

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

[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.

[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.

[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.

[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.

[Low96] Gavin Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In Tiziana Margaria and Bernhard Steffen, editors, *TACAS*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer, 1996.

[MS01] Jonathan Millen and Vitaly Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *ACM Conference on Computer and Communications Security*, pages 166–175. ACM Press, 2001.

[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.