

A Complete Axiomatization of Knowledge and Cryptography

Mika Cohen Mads Dam
KTH CSC
Stockholm, Sweden
{mikac,mfd}@nada.kth.se

Abstract

The combination of first-order epistemic logic and formal cryptography offers a potentially very powerful framework for security protocol verification. In this article, we address two main challenges towards such a combination; First, the expressive power, specifically the epistemic modality, needs to receive concrete computational justification. Second, the logic must be shown to be, in some sense, formally tractable. Addressing the first challenge, we provide a generalized Kripke semantics that uses permutations on the underlying domain of cryptographic messages to reflect agents' limited computational power. Using this approach, we obtain logical characterizations of important concepts of knowledge in the security protocol literature, namely Dolev-Yao style message deduction and static equivalence. Answering the second challenge, we exhibit an axiomatization which is sound and complete relative to the underlying theory of cryptographic terms, and to an omega rule for quantifiers. The axiomatization uses largely standard axioms and rules from first-order modal logic. In addition, it includes some novel axioms for the interaction between knowledge and cryptography. To illustrate the usefulness of the logic we consider protocol examples using mixes, a Crowds style protocol, and electronic payments. Furthermore, we provide embedding results for BAN and SVO.

1 Introduction

Many goals of cryptographic communication concern knowledge. Authenticity, for instance, may mean that a receiver knows the sender of a message, and anonymity may mean that the sender is unknown to an eavesdropper. In these contexts, knowledge is used as a semantical rather than a cognitive concept: The intention is that local interactions (for instance, the local observations of the receiver) force some global system property (for instance, that the message has a certain sender).

Knowledge, in this sense, is often tied to the concept of indistinguishability.

This applies, for instance, in security protocol analysis using some form of observational equivalence (cf. [17]), in multi-agent system semantics using local history identity (cf. [16, 27]), and in information flow theory using low-level observability (cf. [33]). For cryptographic communication the definition of a suitable indistinguishability relation is somewhat delicate. The baseline is computational indistinguishability in the sense of modern cryptography (cf. [20]), i.e., the absence of a computationally feasible experiment to tell two ciphertexts apart. On the other hand, in order to serve as a basis for a useful logic, the relation should be amenable to formal treatment.

To this end, static equivalence [17] has recently emerged as a natural starting point. Static equivalence collects cryptographic terms that are visible to an agent (or the environment) in a frame, roughly a sequence s of terms. Two frames s and s' are equivalent if they satisfy the same equality tests. For instance, if $encrypt(s(i), s(j)) = s(k)$ then $encrypt(s'(i), s'(j)) = s'(k)$, where $s(i)$ is the i :th term in sequence s . Static equivalence is parametrized on an underlying equational theory of cryptographic terms over a signature of “feasibly computable” operators. Depending on the specific choice of theory, strong links between static equivalence and computational indistinguishability can sometimes be established. Computational soundness, static equivalence implying computational indistinguishability, has recently received particular attention (cf. [1]).

In this paper, we use static equivalence as the starting point for building a program logic. Our first step is the observation that static equivalence implicitly defines a correspondence between messages, with messages deduced by the same sequence of computations corresponding to each other: Message $encrypt(s(i), s(j))$ at frame s corresponds to message $encrypt(s'(i), s'(j))$ at frame s' , and so on. This correspondence can be lifted to assignments V of messages to variables x, y, z, \dots of a logic: V at s corresponds to V' at s' if $V(x) = encrypt(s(i), s(j))$ implies that $V'(x) = encrypt(s'(i), s'(j))$, and so on. Motivated by this observation, we instantiate the multi-agent system framework [16], using frames as states and sub-frames as local states. We say that property F is known to agent A at global state s under assignment V , $s, V \models \Box_A F$, if and only if $s', V' \models F$ for all global states s' and assignments V' such that agent A 's local state in s is statically equivalent to A 's local state in s' , and V at A 's local state in s corresponds to V' at A 's local state in s' . We follow counterpart semantics [24] by checking F at s' under a corresponding assignment V' , rather than under the original assignment V . This interpretation has a number of interesting properties.

Firstly, our use of counterpart semantics provides a handle on the difficult issue of mathematical omniscience in epistemic logic. Existing multi-agent system semantics (cf [5, 21, 28, 32, 34]) follow basic Kripke semantics [9], where the assignment V is kept constant as indistinguishable states are scanned: $s, V \models \Box_A F$, if and only if $s', V \models F$ for all indistinguishable states s' . Assuming mathematical equalities depend only on the assignment V , and not on the

global state s , agents are *cryptographically omniscient*, i.e., know all equalities:

$$t = t' \rightarrow \Box_A t = t' \tag{1}$$

for any open terms t and t' built from cryptographic operators f and variables x . For example, $x = \text{decrypt}(y, z) \rightarrow \Box_A x = \text{decrypt}(y, z)$ holds even when agent A has not obtained the key z . Thus, the epistemic modality is vacuous on cryptographic statements. Various suggestions, based on explicit knowledge [16], have been made towards addressing this issue (cf. [6, 22, 25, 29, 30]). However, work on explicit knowledge abandon Kripke-style semantics, lose many of the useful logical properties of knowledge, and provide a very syntactic account of knowledge extraction.

By contrast, we avoid cryptographic omniscience (1) even as we stay within a Kripke-style semantics and preserve most logical properties of knowledge. This is shown by the second result, which is also the main result of the paper: A sound and complete axiomatization, based on axioms and rules from standard first-order logic and modal S5 logic. In addition, the axiomatization includes some novel axioms for the interaction between knowledge and cryptography. The first interaction axiom is a weakening of (1):

$$A \text{ deduces } \bar{x} \rightarrow (y = f(\bar{x}) \rightarrow \Box_A y = f(\bar{x}))$$

for each feasibly computable operator f . The predicate *deduces* is message deduction in the sense of [17], and is definable in terms of the epistemic modality, as will be explained below. Another interaction axiom says that the agent knows a property of some non-deduced values \bar{x} only if this property holds of any non-deduced values \bar{z} :

$$\neg A \text{ deduces } \bar{x}, \bar{z} \rightarrow \Box_A F \rightarrow F[\bar{z}/\bar{x}]$$

In order to obtain completeness with respect to any given theory of abstract cryptography, the axiomatization uses some infinitary machinery. Firstly, we add as axioms all equalities and inequalities from the underlying theory of cryptographic terms. Secondly, we add a second kind of quantifiers, $\forall m$ -quantifiers, with an omega-rule. Intuitively, the formula $\forall m. F[m/x]$ expresses the infinite conjunction $F[M_1/x] \wedge F[M_2/x] \wedge \dots$, where M_1, M_2, \dots lists all ground message terms. As we explain in section 10, the axiomatization is a significant improvement over our earlier result for a propositional epistemic logic [12].

The third result is epistemic characterizations of message deduction [17] and static equivalence. For the former, we obtain $A \text{ deduces } x \leftrightarrow \exists y. \Box_A y = h(x)$, where h is a one-way hash operator (cf. [2] for a related use of the hash operator). The logical characterization of static equivalence, which is rather immediate, gives added credence to our semantics, and allows the transfer of computational soundness results, such as that of [1], to our epistemic logic. It follows, for instance, that if the same properties are known by agent A in global states s and s' then A 's local state in s and s' are computationally indistinguishable.

We illustrate the language in three example protocols, using mixes, a Crowds style protocol [31], and electronic payments. Furthermore, we illustrate the axiomatization by embedding characteristic rules from authentication logics BAN [10] and SVO [34], including an infinitary weakening of necessitation appropriate for BAN. The protocol specifications and the embedding results (as well as the characterization of message deduction above) all rely on the absence of cryptographic omniscience.

2 Messages and Static Equivalence

Let f range over a countable set Σ of public, feasibly computable operators, each equipped with an arity. Let A, B, \dots range over a finite, non-empty set $\mathcal{A} \subseteq \Sigma$ of 0-arity operators, representing public names of distinct agents; Other 0-arity operators in Σ also represent public values, “plain texts”. Let c range over a countably infinite set SEC of secret constants, and x, y, z, \dots range over a countably infinite set VAR of variables. Message terms t are:

$$t ::= x \mid c \mid f(t_1, \dots, t_n)$$

where f has arity n . Write $Var(t)$ for the set of variables in t . Let M, K, N, \dots range over the set \mathcal{T} of ground terms (terms with no occurrences of variables). An abstract model of cryptography is given as a congruence \equiv over ground terms, typically via an equational theory. The set of messages is the set \mathcal{T}_{\equiv} of all equivalence classes with respect to \equiv . Overloading notation, we write M for the equivalence class $[M]_{\equiv}$, and f for its induced operation on classes.

Example 1 *To model pairing and asymmetric encryption we assume the least congruence over ground terms satisfying the following: $fst(pair(M, M')) \equiv M$, $snd(pair(M, M')) \equiv M'$ and $dec(enc(M, pk(K)), K) \equiv M$. Informally, fst/snd picks out first/second components, pk derives a public key from a private key, and enc/dec encrypts/decrypts the first argument using the second as key.*

Throughout this paper, we assume that agent names in \mathcal{A} are non-equivalent. In some results, we assume there is a special unary operator $h \in \Sigma$, with $h(h(M)) \not\equiv M$ and such that if $h(M) \equiv h(M')$ then $M \equiv M'$; We call such an operator a *hash operator*.

Assume a non-empty, countable set LOC of store locations. A state (“store”) over LOC is a partial function s from LOC to \mathcal{T}_{\equiv} . A message is inferable (“deducible”) from a state if the message is directly given by the state, i.e., belongs to the range, or if the message can be obtained from already inferred messages through some $f \in \Sigma$.

Definition 1 *Inferable(s), the messages inferable from s , is the least extension of $ran(s)$ closed under all $f \in \Sigma$.*

Constant c need not be in $Inferable(s)$, but 0-arity f must.

We introduce a second kind of terms, s -terms:

$$\alpha ::= l \mid f(\alpha_1, \dots, \alpha_n)$$

where $l \in \text{dom}(s)$ and $f \in \Sigma$. Each s -term represents an inference path available at s . We extend s to a mapping on s -terms, i.e., $s(f(\alpha_1, \dots, \alpha_n)) = f(s(\alpha_1), \dots, s(\alpha_n))$. The following corollary corresponds to proposition 1 in [2].

Corollary 1 $\text{Inferable}(s) = \{s(\alpha) : \alpha \in s\text{-terms}\}$.

Proof. \subseteq : By induction on the inference length. \supseteq : By induction on α . \square

Two states are statically equivalent if they satisfy the same equality tests:

Definition 2 States s and s' are statically equivalent, written $s \approx s'$, if and only if, $\text{dom}(s) = \text{dom}(s')$ and:

$$s(\alpha) = s(\alpha') \Leftrightarrow s'(\alpha) = s'(\alpha'), \text{ all } s\text{-terms } \alpha, \alpha'$$

In relation to [17], constants c corresponds to private/fresh names, states s correspond to frames, $\text{Inferable}(s)$ corresponds to messages deduction (\vdash) from the frame s , and $s \approx s'$ is static equivalence between (finite) frames s and s' .

3 Indistinguishability under Permutation

To fit in a counterpart semantics, we reformulate static equivalence in a manner strongly reminiscent of framed bisimulation [3]. Assuming $s \approx s'$, the message $s(\alpha)$ at s corresponds to the message $s'(\alpha)$ at s' in that both messages are reached through the same inference path. Motivated by this intuition, we introduce an indistinguishability \sim between states, which is relativized to a permutation ρ on \mathcal{T}_{Ξ} . Informally, if $s \sim^\rho s'$, then any message M at s corresponds to $\rho(M)$ at s' . To qualify as a witness for state indistinguishability, a permutation ρ must respect locations as well as all operations in Σ on inferable messages:

Definition 3 $s \sim^\rho s'$, if and only if, $\text{dom}(s) = \text{dom}(s')$ and:

- $\rho \circ s = s'$.
- $\rho(f(\overline{M})) = f(\overline{\rho(M)})$, if all $M_i \in \text{Inferable}(s)$.

Lemma 1 If $s \sim^\rho s'$ then $\rho(\text{Inferable}(s)) = \text{Inferable}(s')$.

Proof. By induction on inference length. \square

Proposition 1 The following hold:

1. $s \sim^{\text{Id}} s$
2. If $s \sim^\rho s'$ and $s' \sim^{\rho'} s''$ then $s \sim^{\rho' \circ \rho} s''$.

3. If $s \sim^\rho s'$ then $s' \sim^{\rho^{-1}} s$.

Proof. (1) Immediate. (2) From lemma 1. (3) From lemma 1. \square

Write $\overline{\text{Inferable}(s)}$ for the complement of $\text{Inferable}(s)$. Messages in $\overline{\text{Inferable}(s)}$ are anonymous in that every permutation of $\text{Inferable}(s)$ is “epistemically possible”:

Corollary 2 *Assume a permutation π on $\overline{\text{Inferable}(s)}$. Extend π to a permutation ρ on \mathcal{T}_{\equiv} such that $\rho(M) = M$ for $M \in \text{Inferable}(s)$. Then, $s \sim^\rho s$.*

A state s is *normal* if s has countably infinite many non-inferred messages, i.e., $\overline{\text{Inferable}(s)}$ is countably infinite. This corresponds to the assumption in [17] that there always are fresh private names available. In the following two results, relating \sim to \approx , we assume states are normal.

Lemma 2 *$s \sim^\rho s'$ if, and only if, $\text{dom}(s) = \text{dom}(s')$ and $\rho(s(\alpha)) = s'(\alpha)$ for all s -terms α .*

Proof. From corollary 1. \square

Theorem 1 *$s \approx s'$, if and only if, $\exists \rho : s \sim^\rho s'$.*

Proof. Assume $s \approx s'$. Define ρ by: (i) $\rho(s(\alpha)) = s'(\alpha)$, for all s -terms α , and (ii) $\rho(M_i) = N_i$ where M_1, M_2, \dots is an enumeration (without repetitions) of $\overline{\text{Inferable}(s)}$ and N_1, N_2, \dots is an enumeration (without repetitions) of $\overline{\text{Inferable}(s')}$. By corollary 1, $s \sim^\rho s'$. The converse is immediate from lemma 2. \square

4 Systems and Statements

Multi-Agent System We instantiate the multi-agent system framework [16] to our notion of state. A state space is a non-empty set S of states s over LOC , intuitively the set of possible states of some underlying program. An agent projection $|$ assigns a set $LOC|A \subseteq LOC$ of locations observed (accessed) by agent A . The agent projection is lifted to states: $s|A$ is the restriction of s to locations in $LOC|A$. A multi-agent system, or simply a system, is a structure $\mathcal{S} = \langle LOC, S, | \rangle$ of a set LOC of store locations, a state space S and an agent projection $|$.

Example 2 *We model a system where either agent A or agent B posts a message, but agent C cannot observe whom. Assume the message congruence from example 1. Assume two locations: $LOC = \{\text{sender}, \text{post}\}$. The state space is $S = \{s : LOC \rightarrow \mathcal{T}_{\equiv} \mid s(\text{sender}) \in \{A, B\}\}$. Agent C observes only the post location: $LOC|C = \{\text{post}\}$. The system is $\mathcal{S} = \langle LOC, S, | \rangle$.*

Inference and indistinguishability naturally relativize to an agent A : $\text{Inferable}(A, s) =_{df} \text{Inferable}(s|A)$; $s \sim_A^\rho s'$, if and only if, $s|A \sim^\rho s'|A$.

Statements Statements $F \in \mathcal{F}$ are defined by:

$$F ::= t = t' \mid p(t_1, \dots, t_n) \mid \forall x.F \mid \forall m.F[m/x] \mid \Box_A F \mid F \wedge F' \mid \neg F$$

where p is from a countable set \mathcal{P} of predicates, A is an agent identifier in \mathcal{A} , m is from a countably infinite set of “place holders”, and $F[m/x]$ is the result of uniformly replacing free occurrences of variable x by place holder m throughout F . Note that a statement may contain unbound variables, but not unbound place holders.

The language has two types of quantifier reflecting the *de re/de dicto* dichotomy familiar from first order modal logic [9]. Intuitively, $\forall m$ ranges over structured values, while $\forall x$ ranges over their representations “on the wire”: The formula $\forall m.F[m/x]$ expresses the countably infinite conjunction $\bigwedge_M F[M/x]$, where M ranges over all ground terms. Thus, nesting of $\forall m$ -quantifiers and modalities can only express knowledge of (closed) propositions, so called knowledge *de dicto*. By contrast, nesting of $\forall x$ -quantifiers and modalities can express knowledge of objects (messages), so called knowledge *de re*; The statement $\Box_A F(x)$ says of the message (“bitstring”) referred to by x , that agent A knows that it satisfies F . For this reason, we call the $\forall x$ -quantifier the *de re* quantifier, and the $\forall m$ -quantifier the *de dicto* quantifier. To illustrate, the statement $\forall m.(m = M \rightarrow \Box_A m = M)$ is intuitively valid, while the corresponding statement $\forall x.(x = M \rightarrow \Box_A x = M)$ is not. In section 11, we show that the *de dicto* quantifier cannot be reduced to the *de re* quantifier. Although we do believe the use of both types of quantifier is of independent interest, our motivation for the *de dicto* quantifier is technical. To obtain a complete axiomatization, we need an axiom stating that each variable x is “grounded”, i.e., that x refers to some message M . Using the *de dicto* quantifier, we can express this grounding by the statement $\exists m.x = m$.

Interpreted System A predicate interpretation I on a system \mathcal{S} assigns, to each predicate p and state $s \in \mathcal{S}$, a relation $I(p, s)$ in \mathcal{T}_{\equiv} (matching the arity of p). An interpreted system based on a system $\mathcal{S} = \langle LOC, \mathcal{S}, | \rangle$ is a structure $\mathcal{I} = \langle LOC, \mathcal{S}, |, I \rangle$ where I is an interpretation on \mathcal{S} . In some examples and propositions, we explicitly introduce the special unary predicates *A infers* and $@_l$, for $A \in \mathcal{A}$ and $l \in LOC$. When we do so, we implicitly require that $I(A \text{ infers}, s) = \text{Inferable}(A, s)$ and $I(@_l, s) = \{s(l)\}$.

5 Cryptographic Counterpart Semantics

In this section, we interpret the epistemic modality through a counterpart semantics based on the relativized indistinguishability of section 3. Roughly, an agent knows a statement if the statement holds with respect to *corresponding* messages at indistinguishable states. Assume an interpreted system \mathcal{I} , and an assignment $V : VAR \rightarrow \mathcal{T}_{\equiv}$. Assignments are extended homomorphically to terms in the usual way, and $V[x \mapsto M]$ is V except that x is assigned M .

Definition 4 (Truth)

$$\begin{aligned}
s, V \models_{\mathcal{I}} \Box_A F &\Leftrightarrow \forall s' \in S : \forall \rho : s \sim_A^\rho s' \Rightarrow \rho \circ V \models_{\mathcal{I}} F \\
s, V \models_{\mathcal{I}} t = t' &\Leftrightarrow V(t) = V(t') \\
s, V \models_{\mathcal{I}} p(t_1, \dots, t_n) &\Leftrightarrow \langle V(t_1), \dots, V(t_n) \rangle \in I(p, s) \\
s, V \models_{\mathcal{I}} \forall x. F &\Leftrightarrow \forall M \in \mathcal{T}_{\equiv} : s, V[x \mapsto M] \models_{\mathcal{I}} F \\
s, V \models_{\mathcal{I}} \forall m. F[m/x] &\Leftrightarrow \forall M \in \mathcal{T} : s, V \models_{\mathcal{I}} F[M/x]
\end{aligned}$$

For Boolean operators we assume standard truth conditions. Cryptographic omniscience (1) fails, since $V(t) = V(t')$ need not imply that $(\rho \circ V)(t) = (\rho \circ V)(t')$. For instance, say $V(x) = V(M) = M$. Then, $(\rho \circ V)(x) = \rho(M)$, but $(\rho \circ V)(M) = M$. The departure from basic Kripke semantics should not be overstressed: \sim_A^ρ induces a two-dimensional indistinguishability relation between evaluation points: $s, V \sim_A s', V' \Leftrightarrow \exists \rho : s \sim_A^\rho s' \wedge V' = \rho \circ V$. Trivially, $s, V \models_{\mathcal{I}} \Box_A F$, if and only if, $\forall s' \in S : \forall V' : s, V \sim_A s', V' \Rightarrow s', V' \models_{\mathcal{I}} F$. Validity is defined as usual: A statement F is valid in interpreted system \mathcal{I} , written $\models_{\mathcal{I}} F$, if for all $s \in S$ and all assignments V , we have $s, V \models_{\mathcal{I}} F$. Statement F is valid in system \mathcal{S} , written $\models_{\mathcal{S}} F$, if F is valid in all interpreted systems based on \mathcal{S} . Statement F is valid, in symbols $\models F$, if F is valid in all systems. Statement F is valid at a state s , written $s \models F$, if $s, V \models_{\mathcal{I}} F$ for all assignments V and all interpreted systems \mathcal{I} containing s .

Example 3 Consider the interpreted system \mathcal{I} from example 2. Since $sen \notin LOC|C$, agent C does not know the sender: $\models_{\mathcal{I}} @_{sen} x \rightarrow \neg \Box_C @_{sen} x$. However, since $post \in LOC|C$, agent C knows (as “bitstring”) what message is posted: $\models_{\mathcal{I}} @_{post} x \rightarrow \Box_C @_{post} x$. On the other hand, C need not know the structure of the posted message: $\not\models_{\mathcal{I}} @_{post} M \rightarrow \Box_C @_{post} M$. From the former validity and the latter invalidity, it follows that cryptographic omniscience (1) fails: $\not\models_{\mathcal{I}} x = M \rightarrow \Box_C x = M$.

In the following theorem, assume a hash operator h and assume that, for each $s \in S$, there are at least two messages that A cannot infer at s , i.e., $|\overline{Inferable}(A, s)| \geq 2$.

Theorem 2 (Characterization of Inference) *The following is valid:*

$$A \text{ Infers } x \leftrightarrow \exists y. \Box_A y = h(x)$$

Proof. Assume $s, V \models_{\mathcal{I}} A \text{ Infers } x$. By corollary 1, if $s \sim_A^\rho s'$ then $\rho(h(V(x))) = h(\rho(V(x)))$, i.e., $s, V[y \mapsto h(V(x))] \models_{\mathcal{I}} \Box_A y = h(x)$, i.e., $s, V \models_{\mathcal{I}} \exists y. \Box_A y = h(x)$. Conversely, assume $V(x) \notin \overline{Inferable}(A, s)$. Assume $V(y) = h(V(x))$ for some given y . Pick a message M such that $M \notin \overline{Inferable}(A, s)$ and $V(x) \neq M$. (There are at least two non-inferred messages, by our restriction on systems.) Let permutation ρ be identity on \mathcal{T}_{\equiv} , except that $\rho(V(x)) = M$ and $\rho(M) = V(x)$. By corollary 2, $s \sim_A^\rho s$. We consider two cases. Case 1: $M \neq h(V(x))$. Then, $\rho \circ V(y) = V(y) = h(V(x)) \neq h(M)$, by the requirement that h is injective and, by assumption above, $V(x) \neq M$. Since $\rho \circ V(x) = M$, we have

$s, \rho \circ V \not\models_{\mathcal{I}} y = h(x)$, and so $s, V \not\models_{\mathcal{I}} \Box_A y = h(x)$. Case 2: $M = h(V(x))$. Then $\rho \circ V(y) = V(x)$ and $\rho \circ V(x) = M$. Thus, $h(\rho \circ V(x)) = h(M) = h(h(V(x))) \neq V(x) = \rho \circ V(y)$, by the requirement that $h(h(M')) \neq M'$ for all M' . Thus, $s, \rho \circ V \not\models_{\mathcal{I}} y = h(x)$, i.e., $s, V \not\models_{\mathcal{I}} \Box_A y = h(x)$. \square

In theorem 2, recall that the interpretation of predicate A infers at state s is $Inferable(A, s)$. In light of theorem 2, we introduce $\Box_A x$, read “ A knows x ”, as an abbreviation for the statement $\exists y. \Box_A y = h(x)$. We write $\Box_A \bar{x}$ for $\bigwedge_i \Box_A x_i$, and we write $\neg \Box_A \bar{x}$ for $\bigwedge_i \neg \Box_A x_i$.

In the following theorem, we assume local states $s|A$ and $s'|A$ are normal. Moreover, we assume predicates \mathcal{P} includes $@_l$, for $l \in LOC$.

Theorem 3 (Logical Characterization of \approx) *The following are equivalent:*

1. $s|A \approx s'|A$.
2. $s \models \Box_A F$ iff $s' \models \Box_A F$, for all statements F .

Proof. (1) \Rightarrow (2): By proposition 1 and theorem 1. (2) \Rightarrow (1): Assume (1) fails. Then, there is a statement $F =_{df} \exists \bar{x}. t = t' \wedge \bigwedge_i @_{l_i}(x_i)$, where locations $l_i \in LOC|A$ and t and t' are built only from variables x_i and operators in Σ , such that $s \models F$ but $s' \not\models F$. But, $s \models F \rightarrow \Box_A F$, since: $s \models @_{l_i}(x_i) \rightarrow \Box_A @_{l_i}(x_i)$, and $s \models t = t' \rightarrow \Box_A VAR(t) \rightarrow \Box_A t = t'$. The latter can be shown directly, or from lemma 3.10 and soundness theorem 4. \square

6 Security Protocol Examples

6.1 Mix

Consider a mix in the style of [11]. The mix inputs a sequence of encryptions $enc(M_1, pk(K), N_1), \dots, enc(M_l, pk(K), N_l)$, where $pk(K)$ is the mix’s public key, generated by a secret K , and N_i as random seed. The mix later outputs their content in random order: $M_{\pi(1)}, \dots, M_{\pi(l)}$ for some random permutation π on $\{1 \dots l\}$. A spy should not be able to link inputs to outputs:

$$mix\ inputs\ x \wedge mix\ outputs\ y \rightarrow \neg \Box_{spy} x\ contains\ y \quad (3)$$

where $x\ contains\ y$ abbreviates $\exists z. \exists z'. x = enc(y, z, z')$. Moreover, all links should appear possible to the spy:

$$mix\ inputs\ x \wedge mix\ outputs\ y \rightarrow \Diamond_{spy} x\ contains\ y \quad (4)$$

Our concern might be that the mix detects, rather than prevents, information leakage, i.e., whenever the spy determines a link, the mix knows this:

$$mix\ inputs\ x \wedge mix\ outputs\ y \rightarrow \Box_{spy} x\ contains\ y \rightarrow \Box_{mix} \Box_{spy} x\ contains\ y \quad (5)$$

We check the above security goals in an interpreted system implementing the protocol. Let \equiv be the least congruence over ground terms satisfying: $dec(enc(M, pk(K), N), K) \equiv M$, in addition to the equations for pairing in example 1. Write $M_1 \dots M_n$ for the list $pair(M_1, pair(M_2, \dots))$. Fix a number $l > 2$ as the size of message buffer of the mix. For any $k \in SEC$ as the private key of the mix, we assume an input-output relation $InOut_k$ which relates any input list:

$$enc(m_1, pk(k), n_1) \cdot \dots \cdot enc(m_l, pk(k), n_l)$$

where $n_i, m_i \in SEC$ and $m_i \neq k$ and $m_i \neq n_j$, to each output list of the form:

$$m_{\pi(1)} \cdot \dots \cdot m_{\pi(l)}$$

where π is an arbitrary permutation on $\{1 \dots l\}$. We assume four store locations, $LOC = \{in, out, priv, pub\}$, where *in* stores the input list, *out* stores the output list, *priv* stores the private key of the mix, and *pub* stores the public key of the mix. The state space, induced by the input-output relation, is:

$$\mathcal{S}_1 = \{s : LOC \rightarrow \mathcal{T}_{\equiv} \mid \langle s(in), s(out) \rangle \in InOut_{s(priv)} \wedge s(pub) = pk(s(priv))\}$$

The mix observes all store locations, i.e., $LOC|mix = LOC$, while the spy does not observe *priv*: $LOC|spy = \{in, out, pub\}$. The multi-agent system is $\mathcal{S}_1 = \langle LOC, \mathcal{S}_1, | \rangle$. We assume location predicates $@_l$, for $l \in LOC$, and introduce some abbreviations: *mix inputs* x abbreviates $\exists y_1 \dots \exists y_l. @_l (y_1 \dots y_l) \wedge \bigvee_i x = y_i$; *mix outputs* x abbreviates $\exists y_1 \dots \exists y_l. @_l (y_1 \dots y_l) \wedge \bigvee_i x = y_i$.

Proposition 2 \mathcal{S}_1 satisfies (5), but neither of (3) and (4).

Proof. (i): \mathcal{S}_1 satisfies (5): Since $s|spy \subseteq s|mix$, $\models_{\mathcal{S}_1} \Box_{spy} F \rightarrow \Box_{mix} F$. But, $\models \Box_{spy} F \rightarrow \Box_{spy} \Box_{spy} F$ by proposition 1.2. (ii) \mathcal{S}_1 does not satisfy (3): Pick a system state $s \in \mathcal{S}_1$ were all inputs are identical: $s(in) = M \dots M$ and $s(out) = N \dots N$ for some messages M and N . Pick any $s' \in \mathcal{S}_1$ and a permutation ρ such that $s \sim_{spy}^{\rho} s'$. Since $\rho \circ s|spy = s'|spy$, we have $s'(in) = \rho \circ s(in) = \rho(M \dots M) =$ (By the equations for pairing and since ρ is a homomorphism from inferred messages) $= \rho(M) \dots \rho(M)$. Similarly, $s'(out) = \rho(N) \dots \rho(N)$. But, since $s' \in \mathcal{S}_1$, each output in state s' is part of some input, i.e., $s', V[x := \rho(M), y := \rho(N)] \models x \text{ links to } y$. I.e., $s', \rho \circ V[x := M, y := N] \models x \text{ links to } y$. Since s' and ρ were chosen at random, $s, V[x := M, y := N] \models \Box_{spy} x \text{ links to } y$. Thus, (3) fails in \mathcal{S}_1 . (iii): \mathcal{S}_1 does not satisfy (4): Pick a system state $s \in \mathcal{S}_1$ were exactly two inputs are identical. (4) fails at s , since $l > 2$. \square

We modify the implementation so that the mix checks for replays but the spy performs size-comparisons. To achieve the latter, we add a length-computing operator *len*, and equations: $len(enc(M, K, N)) \equiv len(M)$; $len(c) \equiv len(c')$ if $c =_{len} c'$, where $=_{len}$ is a fixed equivalence in SEC . (We assume there are at least l constants of different length.) The length of pairs is irrelevant.

We disallow replays by adding a restriction on the domain of the input-output relation $InOut_k$:

$$enc(m_i, pk(k), n_i) \neq enc(m_j, pk(k), n_j), i \neq j \quad (6)$$

Let S_2 be the resulting new state space and let $\mathcal{S}_2 = \langle LOC, S_2, | \rangle$ be the new multi-agent system.

Proposition 3 \mathcal{S}_2 satisfies (5), but neither of (3) and (4).

Proof. (i): \mathcal{S}_2 satisfies (5): See proof of proposition 2. (ii) \mathcal{S}_2 does not satisfy (3): Pick $s \in S_2$ and assignment V such that $s(in) = V(x_1) \cdot \dots \cdot V(x_i)$ and $s(out) = V(y_1) \cdot \dots \cdot V(y_l)$ and $len(V(x_i)) \neq len(V(x_j))$ for all $i \neq j$. Assume $s, V \models_{\mathcal{S}_2} x_i \text{ contains } y_j$. By equations for length, $len(V(x_i)) = len(V(y_j))$. Assume $s \sim_{spy}^\rho s'$. Since $V(x_i), V(y_j) \in Inferable(spy, s)$ and since ρ is a homomorphism from inferred messages, we get $len(\rho \circ V(x_i)) = len(\rho \circ V(y_j))$. By the above assumption that all inputs have different sizes, we get $s', \rho \circ V \models_{\mathcal{S}_2} x_i \text{ contains } y_j$. Since s' and ρ are arbitrary, $s, V \models_{\mathcal{S}_2} \Box_{spy} x_i \text{ contains } y_j$. (iii) \mathcal{S}_2 does not satisfy (4): Shown similarly to (ii). \square

In both \mathcal{S}_1 and \mathcal{S}_2 , the spy determines what is inside of an input $enc(m_i, pk(k), n_i)$ although the spy cannot infer the matching secret key k . Instead, the spy determines what is inside based on knowledge of the state space. In particular, the spy knows that at every possible state, each output is part of some input.

6.2 Crowds

We consider a Crowds-style protocol [31], which allows members of a crowd to communicate without non-crowd members knowing who is talking to whom. The agents of a set *Crowd* share a symmetric key K . Crowd member A sends a message M anonymously to crowd member B , by sending $enc(pair(B, M), K)$ to some random crowd member C_1 , who in turn sends the message to B or to a random forwarder C_2 , and so on until the message reaches B . For each crowd member A , we assume a local active adversary spy_A . For receiver anonymity, we specify:

$$A \text{ sends } x \rightarrow \Diamond_{spy_A} x \text{ intends destination } B$$

for $B \in Crowds - \{A\}$, where $x \text{ intends destination } B$ abbreviates $\exists y. \exists z. x = enc(pair(B, y), z)$. (We assume that no crowd member initiates communication to itself.) Since spy_A can block messages that A sends, the statement $x \text{ intends destination } B$ must be defined in terms of message structure. For sender anonymity, we specify:

$$A \text{ receives } x \wedge \neg A \text{ originates } x \rightarrow \Diamond_{spy_A} B \text{ originates } x \quad (7)$$

for $B \in Crowds - \{A\}$.

Although sender anonymity (7) does not directly specify knowledge of cryptographic structure, cryptographic omniscience is problematic also for (7). Assume, for instance, an implementation of the protocol where each message M

has a source field; Say, M has the form $pair(A, M')$, where the first component indicates A as the source of the message. Assume source fields are reliable:

$$x \text{ has source field } A \rightarrow \neg B \text{ originates } x, B \neq A$$

where $x \text{ has source field } A$ abbreviates $\exists y. \exists z. x = enc(pair(B, pair(A, y)), z)$. By the rule of normality (i.e., the necessitation rule together with axiom K) and cryptographic omniscience,

$$x \text{ has source field } A \rightarrow \Box_{spy_A} \neg B \text{ originates } x, B \neq A$$

Thus, (7) fails, although, intuitively, the implementation might very well achieve sender anonymity, i.e., spy_A need not know where the messages A receives originate from.

6.3 Dual Signature

Consider a purchasing protocol involving three parties, a customer C , a merchant M , and a bank B . To order an item x_i using payment data (credit card number, etc.) x_p , the customer produces a dual signature [26] using the private signing key x_s :

$$dual(x_i, x_p, x_s) =_{df} sign(pair(h(x_i), h(x_p)), x_s)$$

The merchant receives $dual(x_i, x_p, x_s)$, x_i and $h(x_p)$, while the bank receives $dual(x_i, x_p, x_s)$, $h(x_i)$ and x_p . The dual signature hides the order item x_i from the bank, and the payment data x_p from the merchant, but still the dual signature links x_i to x_p so that their correspondence cannot later be disputed. We consider in more detail what the bank learns during protocol execution. Let variable $x_d = dual(x_i, x_p, x_s)$ refer to the dual signature that C creates in the current run. At the end of the protocol, the bank knows that the dual signature was produced by the customer's private signing key:

$$\Box_B C \text{ signed } x_d$$

where $C \text{ signed } x_d =_{df} \exists x_i. \exists x_p. \exists x_s. x_s \text{ private sign key of } C \wedge x_d = dual(x_i, x_p, x_s)$, with $private \text{ sign key of}$ as a primitive predicate. Using $h(x_i)$ and x_p , the bank can determine the payment data x_p inside:

$$\Box_B x_d \text{ contains payment } x_p$$

where $x_d \text{ contains payment } x_p$ abbreviates $\exists x_i. \exists x_s. x_d = dual(x_i, x_p, x_s)$. But, the bank cannot determine the order item:

$$\neg \Box_B x_d \text{ contains item } x_i$$

where $x_d \text{ contains item } x_i$ abbreviates $\exists x_p. \exists x_s. x_d = dual(x_i, x_p, x_s)$. Finally, the bank is assured that the merchant can determine the order item:

$$\Box_B \exists x_i. \Box_M x_d \text{ contains item } x_i$$

First-order

- (Ins x) $\forall x.F \rightarrow F[y/x]$
- (Ins m) $\forall m.F[m/x] \rightarrow F[M/x]$
- (Bound x) $\forall x.F \leftrightarrow F$, if x is not free in F
- (Bound m) $\forall m.F[m/x] \leftrightarrow F$, if x is not free in F
- (Dist x) $\forall x.(F \rightarrow F') \rightarrow \forall x.F \rightarrow \forall x.F'$
- (Dist m) $\forall m.(F[m/x] \rightarrow F'[m/x]) \rightarrow \forall m.F[m/x] \rightarrow \forall m.F'[m/x]$
- (Subst) $t = t' \rightarrow F[t/x] \rightarrow F[t'/x]$, if F has no modality
- (Ins t) $\forall x.F \rightarrow F[t/x]$, if F has no modality
- (Eq) $t = t$
- (m x) $\exists m.x = m$
- (Taut) F , if F is truth functional tautology
- (Gen x) $\frac{F}{\forall x.F}$
- (MP) $\frac{F \rightarrow F', F}{F'}$

Modal S5

- (K) $\Box_A(F \rightarrow F') \rightarrow (\Box_A F \rightarrow \Box_A F')$
- (T) $\Box_A F \rightarrow F$
- (4) $\Box_A F \rightarrow \Box_A \Box_A F$
- (5) $\neg \Box_A F \rightarrow \Box_A \neg \Box_A F$
- (Nec) $\frac{F}{\Box_A F}$

Knowledge and Cryptography

- (□1) $\Box_A \bar{x} \rightarrow (y = f(\bar{x}) \rightarrow \Box_A y = f(\bar{x}))$
- (□2) $x = y \rightarrow \Box_A x = y$
- (□3) $y = f(\bar{x}) \rightarrow \Box_A \bar{x} \rightarrow \Box_A y$
- (□4) $\Box_A F(\bar{x}, \bar{y}) \rightarrow \Box_A \bar{y} \rightarrow \neg \Box_A \bar{x}, \bar{z} \rightarrow \bigwedge_{i,j} (x_i = x_j \leftrightarrow z_i = z_j) \rightarrow F[\bar{z}/\bar{x}]$
- (□5) $\exists x. \exists y. x \neq y \wedge \neg \Box_A x \wedge \neg \Box_A y$

Omega

- (\equiv) $M = M'$, if $M \equiv M'$
- (\neq) $M \neq M'$, if $M \not\equiv M'$
- (Gen m) $\frac{F[M/x], \text{ all } M \in \mathcal{T}}{\forall m.F[m/x]}$

Figure 1: Axioms and Rules

7 Axiomatization

In table 1, we define a Hilbert-style axiomatization, relative to a message congruence \equiv with a hash operator h . The first group of axioms and rules is inherited from first-order logic, and includes a (less standard) axiom connecting the two kinds of quantifier. The second group is modal S5, as expected for introspective knowledge. The third group contains five axioms for the interaction between knowledge and cryptography. While axiom $(\Box 2)$ is well-known from first-order modal logic, the other four axioms are novel. Axiom $(\Box 1)$ reflects the assumption that each operator f is feasible to compute. Axiom $(\Box 3)$ states that inferred messages are closed under operators f . Axiom $(\Box 4)$ reflects the assumption that non-inferred values are “anonymous”: The statement says that the agent knows a property of some non-inferred values \bar{x} only if this property holds of any non-inferred values \bar{z} with the same pattern of identities. More precisely, assume \bar{x}, \bar{y} are all variables free in F . Assume A infers messages \bar{y} but A cannot infer any of messages \bar{x}, \bar{z} . Assume, finally, that \bar{x} and \bar{z} have the same pattern of identities. Then, $\Box_A F \rightarrow F[\bar{z}/\bar{x}]$. Axiom $(\Box 5)$ reflects the restriction on systems needed for theorem 2, namely that there are at least two messages that agent A does not infer. In appendix A, we provide correspondence results for axioms $(\Box 1)$ and $(\Box 4)$. The fourth group includes all equalities and inequalities from \equiv and an omega-rule for the de dicto quantifier. Write $\vdash F$ when F is a derivable theorem.

Lemma 3 *The following are theorems:*

1. $\forall x. \Box_A F \leftrightarrow \Box_A \forall x. F$
2. $\forall m. \Box_A F[m/x] \leftrightarrow \Box_A \forall m. F[m/x]$
3. $\Box_A x \rightarrow \Box_A \Box_A x$
4. $\neg \Box_A x \rightarrow \Box_A \neg \Box_A x$
5. $x \neq y \rightarrow \Box_A x \neq y$
6. $x = f \rightarrow \Box_A x$, if f is 0-arity
7. $x = y \rightarrow (F[x/z] \rightarrow F[y/z])$
8. $\exists x. x = t$
9. $\Box_A F(\bar{x}, \bar{y}) \rightarrow \Box_A \bar{y} \rightarrow \neg \Box_A \bar{x}, \bar{z} \rightarrow \bigwedge_{i,j} (x_i = x_j \leftrightarrow z_i = z_j) \rightarrow \Box_A F[\bar{z}/\bar{x}]$
10. $x = t \rightarrow \Box_A VAR(t) \rightarrow \Box_A x = t$, if $t \cap SEC = \emptyset$

Proof. (1), (2), (3) and (4): First-order and S5. (5): S5 and axiom $(\Box 2)$. (6): Axiom $(\Box 3)$. (7): First-order, S5 and axiom $(\Box 2)$. (8): Axioms (Ins t) and (Eq). (9): Axioms (4), $(\Box 2)$ and $(\Box 4)$, Lemma 3.3, 3.4 and 3.5. (10): By induction on t . Base case: Axiom $(\Box 2)$. Induction step: Axiom $(\Box 1)$. \square

For a message congruence \equiv without a hash operator, we obtain a sound and complete axiomatization if we take $\Box_A x$ as a primitive unary predicate and add the schema in lemma 3.3 as an additional axiom. The completeness construction in the following section is not effected.

8 Soundness and Completeness

We arrive at the main results. We consider only systems where $|\overline{\text{Inferable}(A, s)}| \geq 2$, for all $s \in S$ and all $A \in \mathcal{A}$.

Theorem 4 (Soundness) $\vdash F \Rightarrow \models F$

Proof. ($\Box 1$): Theorem 2. ($\Box 2$): Since ρ is a function. ($\Box 3$): Theorem 2. ($\Box 4$): Corollary 2 and theorem 2. ($\Box 5$): Theorem 2 and our restriction on systems. (T), (4) and (5): Proposition 1. (K) and (Nec): Independent of the definition of the relativized \sim_A . Non-epistemic axioms and rules are routine. \square

Theorem 5 (Completeness) $\models F \Rightarrow \vdash F$

In the rest of this section, we build the completeness construction. The following sections - sections 9 and 10 - can be read independently. The completeness construction uses abstract counterpart models, with arbitrary states (“possible worlds”) w , arbitrary domain of quantification, arbitrary accessibility relation \longrightarrow_A^ρ and arbitrary (non-rigid) interpretation of function symbols. The first step is a standard canonical Kripke model \mathcal{K} , which is transformed into a counterpart model \mathcal{K}^* by adding some epistemic transitions. For each transition $w \longrightarrow_A w'$ in \mathcal{K} , a transition $w \longrightarrow_A^\pi w'$ is added, where π is any permutation of non-inferred items at w , i.e., items satisfying $\neg\Box_A x$ at w . Continuing, we define a morphism d , which morphs \mathcal{K}^* into a counterpart model $d(\mathcal{K}^*)$ with a rigid interpretation of functions symbols f , given by the background message equivalence \equiv . Finally, a morphism w transforms $d(\mathcal{K}^*)$ into a counterpart model $w(d(\mathcal{K}^*))$, which is equivalent to an interpreted system.

8.1 Abstract Counterpart Model

We review some basics from (a variant of) counterpart semantics (cf. [18]). An abstract counterpart model is a structure $\mathcal{C} = \langle W, D, \longrightarrow, I \rangle$, defined as follows. W is a non-empty set of worlds w , and D is a non-empty domain of objects d . For $A \in \mathcal{A}$, $\longrightarrow_A \subseteq W \times (D \longrightarrow D) \times W$ is the epistemic accessibility relation. Informally, $w \longrightarrow_A^\rho w'$ means that w and w' are indistinguishable for A and each $d \in D$ at w corresponds for A to $\rho(d)$ at w' . I is a world-relative interpretation, i.e., $I(c, w)$ is a member of D , $I(f, w)$ is an operation in D matching the arity of f , and $I(p, w)$ is a relation in D matching the arity of p . Thus, the interpretation of f and c is left open, and need not be rigid. An assignment in \mathcal{C} is a function $V : VAR \longrightarrow D$. Assignments are extended to arbitrary terms with respect to a world w as usual: $V(x, w) = V(x)$,

$V(c, w) = I(c, w)$, $V(f(t_1, \dots, t_n), w) = I(f, w)(V(t_1, w), \dots, V(t_n, w))$. Truth conditions are as expected:

$$\begin{aligned}
w, V \models_{\mathcal{C}} \Box_A F &\Leftrightarrow \forall w' \in W : \forall \rho : w \xrightarrow{\rho}_A w' \Rightarrow w', \rho \circ V \models_{\mathcal{C}} F \\
w, V \models_{\mathcal{C}} t = t' &\Leftrightarrow V(t, w) = V(t', w) \\
w, V \models_{\mathcal{C}} p(t_1, \dots, t_n) &\Leftrightarrow \langle V(t_1, w), \dots, V(t_n, w) \rangle \in I(p, w) \\
w, V \models_{\mathcal{C}} \forall x. F &\Leftrightarrow \forall d \in D : w, V[x \mapsto d] \models_{\mathcal{C}} F \\
w, V \models_{\mathcal{C}} \forall m. F[m/x] &\Leftrightarrow \forall M \in \mathcal{T} : w, V \models_{\mathcal{C}} F[M/x]
\end{aligned}$$

Throughout section 8, ρ ranges over mappings $D \rightarrow D$ instead of permutations on \mathcal{T}_{\equiv} . Any interpreted system $\mathcal{I} = \langle LOC, S, |, I \rangle$ determines a counterpart model $\mathcal{C}_{\mathcal{I}} = \langle S, \mathcal{T}_{\equiv}, \sim, I' \rangle$, where \sim_A is defined as in section 4 and $I'(p, s) = I(p, s)$ and $I'(f, w) = f$ and $I'(c, w) = c$. We say that $\mathcal{C}_{\mathcal{I}}$ is induced by \mathcal{I} .

Corollary 3 $s, V \models_{\mathcal{I}} F$ iff $s, V \models_{\mathcal{C}_{\mathcal{I}}} F$.

A counterpart model \mathcal{C} is Kripkean if $w \xrightarrow{\rho}_A w'$ implies that $\rho = Id$, where Id is identity on D . When \mathcal{C} is Kripkean, we omit the index Id , and write $w \xrightarrow{Id}_A w'$ for the transition $w \xrightarrow{Id}_A w'$. We say that substitutions are bijective in \mathcal{C} , if $w \xrightarrow{\rho}_A w'$ implies ρ is a permutation on D .

Assume a counterpart model $\mathcal{C} = \langle W, D, \xrightarrow{\cdot}_A, I \rangle$. Assume a set W' of worlds and a domain D' . A morphism from \mathcal{C} to W' and D' is a pair w, d such that:

- $w : W \rightarrow W'$ is a bijective map
- $d_w : D \rightarrow D'$ is a bijective map, for each $w \in W$

The morphism w, d is a domain-morphism, if $W = W'$ and w is identity on W . The morphism w, d is a world-morphism, if $D = D'$ and d_w is identity on D . For domain-morphisms, we leave the identity w implicit. Similarly, for world-morphisms, we leave the mapping d implicit. Let w, d be a morphism from \mathcal{C} to W' and D' . The application of w, d on \mathcal{C} is $wd(\mathcal{C}) = \langle W', D' \xrightarrow{wd}_A, I^{wd} \rangle$, where

- $w(w) \xrightarrow{wd}_A w(w')$ iff $w \xrightarrow{\rho'}_A w'$ where $\rho' = d_w^{-1} \circ \rho \circ d_w$.
- $I^{wd}(o, w(w)) = d_w(I(o, w))$, $o \in SEC \cup \Sigma \cup \mathcal{P}$.

Thus, $wd(\mathcal{C})$ is the result of pointwise “relabelling” \mathcal{C} through w and d .

Lemma 4 $w, V \models_{\mathcal{C}} F \Leftrightarrow w(w), d_w \circ V \models_{wd(\mathcal{C})} F$.

Proof. By induction on t , $(d_w \circ V)(t, w(w)) = d_w(V(t, w))$. The lemma follows by induction on F . \square

8.2 Canonical Kripke Model

In this section, we reach the truth lemma for a canonical Kripke model in a standard way [19]. A statement F is derivable from a set Γ of statements, in

symbols $\Gamma \vdash F$, if there is a finite number of statements $F_1, \dots, F_n \in \Gamma$ such that $\vdash F_1, \dots, F_n \rightarrow F$. The set Γ is consistent if $\Gamma \not\vdash \perp$, and Γ is maximal consistent if it is consistent and no larger set is. The set Γ is omega-complete if whenever $\Gamma \vdash F[y/x]$ for all $y \in VAR$ then $\Gamma \vdash \forall x.F$ and, also, whenever $\Gamma \vdash F[M/x]$ for all $M \in \mathcal{T}$ then $\Gamma \vdash \forall m.F[m/x]$. The set Γ is saturated if it is maximal consistent and omega-complete. We obtain standard lemmas for omega-completion and saturation.

Lemma 5 \emptyset is omega-complete.

Proof. Immediate from axioms (Gen m) and (Gen x). □

Lemma 6 If Γ is omega-complete then so is $\Gamma \cup \{F\}$.

Proof. Omega-completion for de re quantifiers: Standard. Assume Γ is omega-complete w.r.t. de dicto quantifiers. Assume $\Gamma, F_0 \vdash F[M/x]$ all $M \in \mathcal{T}$, i.e., $\Gamma \vdash F_0 \rightarrow F[M/x]$ all $M \in \mathcal{T}$. Pick x' not free in F_0 . Then, $F_0 \rightarrow F[M/x]$ is $(F_0 \rightarrow F[x'/x])[M/x']$. So, $\Gamma \vdash (F_0 \rightarrow F[x'/x])[M/x']$ all $M \in \mathcal{T}$. By omega-completeness of Γ , we get $\Gamma \vdash \forall m.(F_0 \rightarrow F[x'/x])[m/x']$, i.e., $\Gamma \vdash \forall m.(F_0[m/x'] \rightarrow F[x'/x][m/x'])$, i.e., $\Gamma \vdash \forall m.(F_0 \rightarrow F[m/x])$, i.e., by axiom (Dist m), $\Gamma \vdash \forall m.F_0 \rightarrow \forall m.F[m/x]$, i.e., by axiom (Bound m), $\Gamma \vdash F_0 \rightarrow \forall m.F[m/x]$, i.e., $\Gamma, F_0 \vdash \forall m.F[m/x]$. □

Lemma 7 If Γ is omega-complete then so is $\Gamma|A$.

Proof. Omega-completion for de re quantifiers: Standard. Assume Γ is omega-complete w.r.t. de dicto quantifiers. Assume $\Gamma|A \vdash F[M/x]$ all $M \in \mathcal{T}$. By axiom (K) and rule (Nec), $\Box_A \Gamma|A \vdash \Box_A F[M/x]$ all $M \in \mathcal{T}$, i.e., $\Gamma \vdash \Box_A F[M/x]$ all $M \in \mathcal{T}$. By omega-completeness of Γ , $\Gamma \vdash \forall m.\Box_A F[m/x]$. By lemma 3.2, $\Gamma \vdash \Box_A \forall m.F[m/x]$, i.e., $\Gamma|A \vdash \forall m.F[m/x]$. □

Lemma 8 (Extension Lemma) Every consistent and omega-complete set can be extended to a saturated set.

Proof. We follow a standard generalization of the Lindenbaum construction. Assume a consistent and omega-complete set Γ . Assume an enumeration F_1, F_2, \dots of all statements. We define a sequence of extensions of Γ as follows:

- $\Gamma_0 = \Gamma$.
- If $\Gamma_{n-1}, F_n \vdash \perp$, $F_n = \forall m.F[m/x]$,
then $\Gamma_n = \Gamma_{n-1} \cup \{\neg \forall m.F[m/x], \neg F[M/x]\}$
- else if $\Gamma_{n-1}, F_n \vdash \perp$, $F_n = \forall x.F$,
then $\Gamma_n = \Gamma_{n-1} \cup \{\neg \forall x.F, \neg F[y/x]\}$
- else if $\Gamma_{n-1}, F_n \vdash \perp$, then $\Gamma_n = \Gamma_{n-1} \cup \{\neg F_n\}$
- else $\Gamma_n = \Gamma_{n-1} \cup \{F_n\}$.

where M and y are chosen so that Γ_n is consistent; We show that there are such M and y . Assume Γ_{n-1} is consistent. Assume $\Gamma_{n-1}, \forall m.F[m/x] \vdash \perp$. Assume there is no appropriate M , i.e., assume $\Gamma_{n-1}, \neg\forall m.F[m/x], \neg F[M/x] \vdash \perp$ for all M , i.e., $\Gamma_{n-1}, \neg\forall m.F[m/x] \vdash F[M/x]$ all M . By lemma 6, the set $\Gamma_{n-1} \cup \{\neg\forall m.F[m/x]\}$ is omega-complete. Thus, $\Gamma_{n-1}, \neg\forall m.F[m/x] \vdash \forall m.F[m/x]$, i.e., $\Gamma_{n-1} \vdash \forall m.F[m/x]$. By assumptions, $\Gamma_{n-1}, \forall m.F[m/x] \vdash \perp$, and so, $\Gamma_{n-1} \vdash \perp$, contrary to assumptions. In the same way, lemma 6 tells us that there is an appropriate y . Thus, Γ_n is consistent, and, consequently, $\Gamma^* = \bigcup_n \Gamma_n$ is a maximal consistent set. Trivially, Γ^* is omega-complete. Thus, Γ^* is saturated and $\Gamma \subseteq \Gamma^*$. \square

Given a saturated set w_0 , the canonical Kripke model $\mathcal{K} = \langle W, D, \rightarrow, I \rangle$ is defined as follows. The set W of worlds is the set of all saturated sets which contain $x = y$ just in case w_0 does. The domain D is the set of equivalence classes $|x| = \{y : x = y \in w_0\}$. The epistemic accessibility is given by: $w \rightarrow_A w' \Leftrightarrow w|A \subseteq w'$, where $w|A$ is $\{F : \Box_A F \in w\}$. Finally, the interpretation is defined as follows: $I(f, w)(|x_1|, \dots, |x_n|) = |y|$ iff $(f(x_1, \dots, x_n) = y) \in w$, and $I(c, w) = |y|$ iff $(c = y) \in w$. The canonical assignment $V_{\mathcal{K}}$ assigns $|x|$ to variable x .

Lemma 9 (Truth Lemma for \mathcal{K}) $w, V_{\mathcal{K}} \models_{\mathcal{K}} F \Leftrightarrow F \in w$

Proof. From lemmas 6, 7 and 8. The proof is standard. \square

Corollary 4 $w \rightarrow_A w' \Leftrightarrow w|A = w'|A$

Proof. S5. \square

8.3 Anonymous Non-inferred Items

We transform \mathcal{K} into a model where non-inferred items, i.e., items satisfying $\neg\Box_A x$, are anonymous in the sense that every permutation of such items is “epistemically possible”. The transformation relies on axiom ($\Box 4$). Assume a counterpart model $\mathcal{C} = \langle W, D, \rightarrow, I \rangle$. Write $Inferable_{\mathcal{C}}(A, w)$ for the set of items inferred by agent A at world w , i.e., $Inferable_{\mathcal{C}}(A, w)$ is $\{d \in D \mid w, V[\mapsto d] \models_{\mathcal{C}} \Box_A x\}$. The anonymization of \mathcal{C} is the model $\mathcal{C}^* = \langle W, D, \xrightarrow{*}, I \rangle$, where $\xrightarrow{*}$ is the least extension of \rightarrow such that

$$w \xrightarrow{*}_A \rho w' \Rightarrow w \xrightarrow{*}_A \rho \circ \pi w'$$

for every permutation π on $\overline{Inferable_{\mathcal{C}}(A, w)}$. (π is extended to the whole domain D in the expected way: $\pi(d) = d$ if $d \in Inferable_{\mathcal{C}}(A, w)$.)

Corollary 5 $w \xrightarrow{*}_A \rho w'$, if and only if, there is ρ' and π such that $\rho = \rho' \circ \pi$ and $w \xrightarrow{\rho'}_A w'$ and π is a permutation on $\overline{Inferable_{\mathcal{C}}(A, w)}$.

Proof. Immediate. \square

Lemma 10 *Assume \mathcal{C} validates the schema in lemma 3.9. Assume substitutions are bijective in \mathcal{C} . Then, $w, V \models_{\mathcal{C}} F \Leftrightarrow w, V \models_{\mathcal{C}^*} F$.*

Proof. By induction on complexity of F . Base case, and induction step for Boolean operators and quantifiers: Immediate. Induction step for modal operators: If $w, V \models_{\mathcal{C}^*} \Box_A F$ then $w, V \models_{\mathcal{C}} \Box_A F$, since $\xrightarrow{*}_A \supseteq \longrightarrow_A$, from corollary 5. For the converse, assume

$$w, V \models_{\mathcal{C}} \Box_A F \quad (8)$$

Let $x_1, \dots, x_m, y_1, \dots, y_n$ be a listing of all free variables in F such that

$$w, V \models_{\mathcal{C}} \neg \Box_A x_i \quad (9)$$

$$w, V \models_{\mathcal{C}} \Box_A y_i \quad (10)$$

Assume $w \xrightarrow{*}_A \rho w'$. By corollary 5, there is ρ' and π such that $\rho = \rho' \circ \pi$ and $w \xrightarrow{\rho'}_A w'$ and π is a permutation on $\overline{\text{Inferable}_{\mathcal{C}}(A, w)}$. Thus,

$$\rho'(V(y_i)) = \rho(V(y_i)) \quad (11)$$

Since ρ, ρ' and π are bijective, there are $d_1, \dots, d_m \in D$ such that: .

$$\rho'(d_i) = \rho(V(x_i)) \quad (12)$$

and

$$d_i = d_j \Leftrightarrow V(x_i) = V(x_j) \quad (13)$$

Pick fresh variables z_1, \dots, z_m (i.e., fresh w.r.t. F). By (13),

$$w, V[z_1 \mapsto d_1, \dots, z_m \mapsto d_m] \models_{\mathcal{C}} z_i = z_j \leftrightarrow x_i = x_j \quad (14)$$

We have:

$$w, V[z_1 \mapsto d_1, \dots, z_m \mapsto d_m] \models_{\mathcal{C}} \neg \Box_A z_i \quad (15)$$

To see this, assume $d_i \in \text{Inferable}_{\mathcal{C}}(A, w)$. Then, $\rho(d_i) = \rho' \circ \pi(d_i) = \rho'(d_i) =$ (by (12)) $= \rho(V(x_i))$. Since ρ is bijective, $d_i = V(x_i)$, contradicting (9). From (8), (9), (10), (14) and (15) and the assumption that \mathcal{C} validates the schema in lemma 3.9,

$$w, V[z_1 \mapsto d_1, \dots, z_m \mapsto d_m] \models_{\mathcal{C}} \Box_A F[\bar{z}/\bar{x}]$$

Thus,

$$w', \rho' \circ V[z_1 \mapsto d_1, \dots, z_m \mapsto d_m] \models_{\mathcal{C}} F[\bar{z}/\bar{x}]$$

By induction assumption,

$$w', \rho' \circ V[z_1 \mapsto d_1, \dots, z_m \mapsto d_m] \models_{\mathcal{C}^*} F[\bar{z}/\bar{x}]$$

By (11) and (12), we get $w', \rho \circ V \models_{\mathcal{C}^*} F$. Since w' and ρ were chosen arbitrarily, we conclude that $w, V \models_{\mathcal{C}^*} \Box_A F$. \square

Lemma 11 $w, V \models_{\mathcal{K}} F \Leftrightarrow w, V \models_{\mathcal{K}^*} F$.

Proof. From lemma 10, since the assumptions for that lemma hold: Substitutions are bijective in \mathcal{K} : $w \xrightarrow{\rho}_A w'$ implies that ρ is identity on D , i.e., a bijection. \mathcal{K} validates the schema in lemma 3.9: Lemma 3.9 and lemma 9. \square

8.4 Rigid Operators

We define a domain-morphism \mathbf{d} , which morphs \mathcal{K}^* into a model $\mathbf{d}(\mathcal{K}^*)$ where operators f and constants c have their intended, rigid denotation, given by the background equivalence \equiv . The transformation relies on axioms $(m\ x)$, (\equiv) and (\neq) . For each $w \in W$, we relate D and \mathcal{T}_{\equiv} by the relation:

$$\mathbf{d}_w = \{\langle |x|, M \rangle \mid x = M \in w\}$$

Lemma 12 \mathbf{d} is a morphism from \mathcal{K}^* to W and \mathcal{T}_{\equiv} .

Proof. From axioms $(m\ x)$, (\equiv) , (\neq) and (Subst), and lemma 3.8. \square

Let $\mathbf{d}(\mathcal{K}^*) = \langle W, \mathcal{T}_{\equiv} \xrightarrow{\mathbf{d}}, I^{\mathbf{d}} \rangle$ be application of \mathbf{d} on \mathcal{K}^* .

Lemma 13 $I^{\mathbf{d}}(f, w) = f$ and $I^{\mathbf{d}}(c, w) = c$.

Proof. From axioms (\equiv) , (\neq) and (Subst) and lemma 12. \square

We end this sub-section with two lemmas that will be used in the final transformations step. We say that ρ respects Σ on $X \subseteq \mathcal{T}_{\equiv}$ if

$$\rho(f(\overline{M})) = f(\overline{\rho(M)}), \text{ if all } M_i \in X \text{ and } f \in \Sigma$$

Lemma 14 Assume $w \xrightarrow{\mathbf{d}}_A^{\rho} w'$.

1. ρ is a permutation on \mathcal{T}_{\equiv} .
2. ρ respects Σ on $\mathbf{d}(\text{Inferable}_{\mathcal{K}}(w, A))$.
3. $\rho(M) = \mathbf{d}_{w'} \circ \mathbf{d}_w^{-1}(M)$ if $M \in \mathbf{d}(\text{Inferable}_{\mathcal{K}}(w, A))$.

Proof. Assume $w \xrightarrow{\mathbf{d}}_A^{\rho} w'$. (1): From corollary 5 and lemma 12. (3): By construction of $\xrightarrow{\mathbf{d}}_A^{\rho}$, $w \xrightarrow{\star}_A^{\rho'} w'$ where $\rho' = \mathbf{d}_{w'}^{-1} \circ \rho \circ \mathbf{d}_w$. By corollary 5, $\rho'(|x|) = |x|$ for $|x| \in \text{Inferable}_{\mathcal{K}}(w, A)$. Thus, $\rho(M) = \mathbf{d}_{w'} \circ \mathbf{d}_w^{-1}(M)$ if $M \in \mathbf{d}(\text{Inferable}_{\mathcal{K}}(w, A))$. (2): Assume $M_1, \dots, M_n \in \mathbf{d}(\text{Inferable}_{\mathcal{K}}(w, A))$. I.e., there are variables x_1, \dots, x_n such that $\Box_A x_i \in w$ and $x_i = M_i \in w$. Pick variable y such that $y = f(x_1, \dots, x_n) \in w$. By axiom $(\Box 1)$, $\Box_A y = f(x_1, \dots, x_n) \in w$. By corollary 4, $y = f(x_1, \dots, x_n) \in w'$. By lemma 13, $\mathbf{d}_{w'}(|y|) = f(\mathbf{d}_{w'}(|x_1|), \dots, \mathbf{d}_{w'}(|x_n|))$ and $\mathbf{d}_w(|y|) = f(\mathbf{d}_w(|x_1|), \dots, \mathbf{d}_w(|x_n|))$. By axiom $(\Box 3)$, $\Box_A y \in w$. By (3), $\rho(\mathbf{d}_w(|y|)) = \mathbf{d}_{w'}(|y|)$. But, from above, $\mathbf{d}_{w'}(|y|) = f(\mathbf{d}_{w'}(|x_1|), \dots, \mathbf{d}_{w'}(|x_n|)) = f(\rho(\mathbf{d}_w(|x_1|)), \dots, \rho(\mathbf{d}_w(|x_n|)))$. Thus, $\rho(f(M_1, \dots, M_n)) = f(\rho(M_1), \dots, \rho(M_n))$, since $\mathbf{d}_w(|y|) = f(M_1, \dots, M_n)$ from axiom (Subst) and the fact that $y = f(x_1, \dots, x_n) \in w$. \square

Lemma 15 Assume

1. $w \longrightarrow_A w'$.
2. ρ is a permutation on \mathcal{T}_{\equiv} .

3. $\rho(M) = d_{w'} \circ d_w^{-1}(M)$ if $M \in d(\text{Inferable}_{\mathcal{K}}(w, A))$.

Then, $w \xrightarrow{d}_{\rho} w'$.

Proof. Let $\rho' = d_{w'}^{-1} \circ \rho \circ d_w$. By assumptions (2) and (3) and lemma 12, ρ' is identity on $\text{Inferable}_{\mathcal{K}}(w, A)$ and permutes $\overline{\text{Inferable}_{\mathcal{K}}(w, A)}$. By assumption (1) and corollary 5, $w \xrightarrow{\rho'} w'$. By construction of \xrightarrow{d}_{ρ} , $w \xrightarrow{d}_{\rho} w'$. \square

8.5 Canonical Interpreted System

Finally, we define a world-morphism w , which morphs $d(\mathcal{K}^*)$ into a model $w(d(\mathcal{K}^*))$ induced by an interpreted system. The transformation step relies on axioms ($\square 1$) and ($\square 5$). We assume the following set of store locations:

$$LOC = \mathcal{F} \cup ((D \cup \mathcal{F}) \times \{A\})$$

(where D is the domain in \mathcal{K} and \mathcal{K}^*). Each agent observes store locations indexed by itself:

$$LOC|A = (D \cup \mathcal{F}) \times \{A\}$$

The morphism w maps W to states over LOC defined by:

1. $w(w)(\langle |x|, A \rangle) = d_w(|x|)$, if $|x| \in \text{Inferable}_{\mathcal{K}}(w, A)$.
2. $w(w)(\langle |x|, A \rangle) = \perp$, if $|x| \notin \text{Inferable}_{\mathcal{K}}(w, A)$.
3. $w(w)(\langle F, A \rangle) = \top$, if $\square_A F \in w$.
4. $w(w)(\langle F, A \rangle) = \perp$, if $\square_A F \notin w$.
5. $w(w)(F) = \top$, if $F \in w$.
6. $w(w)(F) = \perp$, if $F \notin w$.

where \perp and \top are two non-equivalent 0-arity operators from Σ . (If there is only one such operator, i.e., the single agent A , then let $\perp = A$ and $\top = h(A)$.) Requirements (3) and (4) on w encode the knowledge state $w|A$ inside the local state $w(w)|A$. Requirements (5) and (6) ensure injectivity. Requirements (1) and (2), together with (3) and (4), ensure that the same permutations ρ are possible between $w(w)$ and $w(w')$ in \sim_A as between w and w' in \xrightarrow{d}_A .

Corollary 6 w is a morphism from $d(\mathcal{K}^*)$ to S and \mathcal{T}_{\perp} .

Proof. Since w is injective. \square

Lemma 16 $\text{Inferable}(A, w(w)) = d_w(\text{Inferable}_{\mathcal{K}}(A, w))$.

Proof. $Inferable(\mathbf{w}(w), A) \supseteq \mathbf{d}_w(Inferable_{\mathcal{K}}(w, A))$: From condition (1) in the definition of \mathbf{w} . $Inferable(\mathbf{w}(w), A) \subseteq \mathbf{d}_w(Inferable_{\mathcal{K}}(w, A))$: By induction on length of the derivation that establishes $M \in Inferable(\mathbf{w}(w), A)$. Base case. Assume $M \in \text{ran}(\mathbf{w}(w)|A)$. If $M \in \{\top, \perp\}$ then $M \in \mathbf{d}_w(Inferable_{\mathcal{K}}(w, A))$, by lemma 3.6. On the other hand, if M is $\mathbf{d}_w(|x|)$ and $|x| \in Inferable_{\mathcal{K}}(w, A)$, then, trivially, $M \in \mathbf{d}_w(Inferable_{\mathcal{K}}(w, A))$. Induction step. Assume $M_1, \dots, M_n \in Inferable(\mathbf{w}(w), A)$. By induction assumption, $M_1, \dots, M_n \in \mathbf{d}_w(Inferable_{\mathcal{K}}(w, A))$. I.e., there are $|x_1|, \dots, |x_n| \in Inferable_{\mathcal{K}}(w, A)$ such that $\mathbf{d}_w(|x_i|) = M_i$. By axiom (Subst), $\Box_A x_i \in w$. Since $\vdash \exists y. y = f(x_1, \dots, x_n)$, we have $y = f(x_1, \dots, x_n) \in w$ for some $y \in VAR$. By lemma 13, $\mathbf{d}_w(|y|) = [f](\mathbf{d}_w(|x_1|), \dots, \mathbf{d}_w(|x_n|))$. By axiom ($\Box 3$), $\Box_A y \in w$, i.e., $|y| \in Inferable_{\mathcal{K}}(w, A)$, i.e., $\mathbf{d}_w(|y|) \in \mathbf{d}_w(Inferable_{\mathcal{K}}(w, A))$, i.e., $[f](\mathbf{d}_w(|x_1|), \dots, \mathbf{d}_w(|x_n|)) \in \mathbf{d}_w(Inferable_{\mathcal{K}}(w, A))$, i.e., $[f](M_1, \dots, M_n) \in \mathbf{d}_w(Inferable_{\mathcal{K}}(w, A))$. \square

Lemma 17 $w \xrightarrow{d}_A^\rho w'$, if and only if, $\mathbf{w}(w) \sim_A^\rho \mathbf{w}(w')$.

Proof. \Rightarrow -direction: Assume $w \xrightarrow{d}_A^\rho w'$. We need to show that ρ is a permutation on \mathcal{T}_{\equiv} , ρ respects Σ on $Inferable(\mathbf{w}(w), A)$ and ρ respects $LOC|A$ between $\mathbf{w}(w)$ and $\mathbf{w}(w')$, i.e., $\rho \circ \mathbf{w}(w)|A = \mathbf{w}(w')|A$. (i) ρ is a permutation on \mathcal{T}_{\equiv} : Lemma 14.1. (ii) ρ respects Σ on $Inferable(\mathbf{w}(w), A)$: Lemma 14.2 and lemma 16. (iii) ρ respects $LOC|A$ between $\mathbf{w}(w)$ and $\mathbf{w}(w')$, i.e., $\rho(\mathbf{w}(w)|A) = \mathbf{w}(w')|A$: We show that $\rho(\mathbf{w}(w)(\langle |x|, A \rangle)) = \mathbf{w}(w')(\langle |x|, A \rangle)$; Respect for other locations is shown similarly. By construction of \xrightarrow{d}_A^ρ , we have $w \longrightarrow_A w'$. Assume $|x| \in Inferable_{\mathcal{K}}(w, A)$. By corollary 4, $|x| \in Inferable_{\mathcal{K}}(w', A)$. By lemma 14.3 and condition (1) in the construction of \mathbf{w} , $\rho(\mathbf{w}(w)(\langle |x|, A \rangle)) = \mathbf{w}(w')(\langle |x|, A \rangle)$. Assume $|x| \notin Inferable_{\mathcal{K}}(w, A)$. By corollary 4, $|x| \notin Inferable_{\mathcal{K}}(w', A)$. By condition (2) in the construction of \mathbf{w} , $\mathbf{w}(w)(\langle |x|, A \rangle) = \mathbf{w}(w')(\langle |x|, A \rangle) = \perp$. But, from lemma 14.2 and lemma 16, $\rho(\perp) = \perp$.

\Leftarrow -direction: Assume $\mathbf{w}(w) \sim_A^\rho \mathbf{w}(w')$. We show conditions (1), (2) and (3) in lemma 15, from which it follows that $w \xrightarrow{d}_A^\rho w'$. Condition (1): Since ρ respects Σ on $Inferable(\mathbf{w}(w), A)$, $\rho(\perp) = \perp$ and $\rho(\top) = \top$. Thus, since ρ respects $LOC|A$ between $\mathbf{w}(w)$ and $\mathbf{w}(w')$, we have $w|A = w'|A$, from conditions (3) and (4) in the construction of \mathbf{w} . By corollary 4, $w \longrightarrow_A w'$. Condition (2): By construction of \longrightarrow_A . Condition (3): Since $w \longrightarrow_A w'$, by corollary 4, $|x| \in Inferable_{\mathcal{K}}(w, A)$ iff $|x| \in Inferable_{\mathcal{K}}(w', A)$. Let $|x| \in Inferable_{\mathcal{K}}(w, A)$. Since ρ respects Σ on $Inferable(\mathbf{w}(w), A)$, $\rho(\mathbf{d}_w(|x|)) = \mathbf{d}_{w'}(|x|)$, from condition (1) in the construction of \mathbf{w} . \square

Let the canonical interpreted system be $\mathcal{I} = \langle LOC, S, |, I \rangle$, where $S = \{\mathbf{w}(w) : w \in W\}$ and $I(p, \mathbf{w}(w)) = \{(M_1, \dots, M_n) \mid \mathbf{w}(w)(p(M_1, \dots, M_n)) = \top\}$.

Lemma 18 $I(p, \mathbf{w}(w)) = I^d(p, w)$.

Proof. From axiom (Subst). \square

Lemma 19 $w(d(\mathcal{K}^*))$ is induced by \mathcal{I} .

Proof. From lemmas 12, 13, 17 and 18. □

Lemma 20 $\overline{\text{Inferable}(w(w), A)}$ has at least two members.

Proof. By axiom ($\square 5$), $\overline{\text{Inferable}_{\mathcal{K}}(w, A)}$ has at least two members $|x|$ and $|y|$. By lemmas 12 and 16, $\overline{\text{Inferable}(w(w), A)}$ has at least the two members $d_w(|x|)$ and $d_w(|y|)$. □

Theorem 6 Every consistent statement is satisfiable in some interpreted system.

Proof. Assume $\not\models \neg F$. By lemmas 5, 6 and 8, there is saturated set w_0 containing F . Starting from w_0 , build the canonical assignment $V_{\mathcal{K}}$ and the canonical interpreted system \mathcal{I} . By lemma ??, $w(w_0), d_{w_0} \circ V_{\mathcal{K}} \models_{\mathcal{I}} F$. By lemma 20, \mathcal{I} satisfies our requirement on systems. □

From theorem 6, we get completeness theorem 5.

9 Embedding of BAN and SVO

9.1 The BAN Modality

We show how to capture the epistemic modality from BAN logic [10]. BAN statements β are propositional, built from closed atomic statements $p(M_1, \dots, M_n)$, epistemic modalities and Boolean operators.¹ Protocol derivations using BAN proceed from assumptions about what messages an agent sees (i.e., can extract from received messages) to a conclusion about what the agent knows about some inferred key, such as knowing that another agent knows the key to be fresh. The step from seeing to knowing is through the so called message meaning rule:

$$A \text{ sees } enc(M, K) \rightarrow \square_A \text{ good}_G K \rightarrow \square_A \bigvee_{B \in G} B \text{ said } enc(M, K) \quad (MMR)$$

where agent B said a message if that message is part of something B sent, and a key is $good_G$ if that key is good as a symmetric encryption key for communication between agents in group $G \subseteq \mathcal{A}$. The formulation (MMR) abstracts from the original message meaning rule, leaving out the assumption that messages contain a reliable sender field [12, 13]. In addition to the message meaning rule, BAN includes several other epistemic axioms, all similar to:

$$\text{fresh } M \rightarrow \square_A \text{ good}_G K \rightarrow \square_A \text{ fresh } enc(M, K) \quad (Fresh)$$

where a message is fresh if it is not part of something sent in an old session.

The rule of necessitation (Nec) is incompatible with the intended meaning of BAN axioms, since ground terms M refer *de re* (“directly”) in BAN [12, 13].

¹Original BAN includes “idealized” messages, but no negation. But, these differences are orthogonal to our concern here.

BAN does not include a substitute for the necessitation rule, only various predicate specific axioms, i.e., (*MMR*) and axioms similar to (*Fresh*). We propose that the following omega-weakening of necessitation is faithful to BAN:

$$\frac{\beta[\overline{M}/\overline{c}], \text{ all } \overline{M}}{\Box_A \overline{M} \rightarrow \Box_A \beta[\overline{M}/\overline{c}]} \quad (WNec)$$

where \overline{c} is all constants from *SEC* occurring in β . For instance, assume $enc(M, K)$ contains M is “valid” in BAN, for all $M, K \in \mathcal{T}$. Then, by rule (*WNec*), so is $\Box_A M, K \rightarrow \Box_A enc(M, K)$ contains M . Similar weakenings of necessitation appear in [12, 13, 23].

Define translation τ from BAN into our logic:

$$\beta(\overline{M})^\tau = \exists \overline{x}. (\beta(\overline{x}) \wedge \bigwedge_i x_i = M_i)$$

where \overline{M} is a list $\langle M_1, \dots, M_n \rangle$ of all ground terms occurring as arguments to predicates in β , $\beta(\overline{x})$ is the result of substituting x_i for M_i , and $\exists \overline{x}$ abbreviates $\exists x_1 \dots \exists x_n$. For instance, τ translates $\Box_A \Box_B A$ receives $enc(M, K)$ to $\exists x.x = enc(M, K) \wedge \Box_A \Box_B A$ receives x . Let $WNec^\tau$ be the τ -translations of *WNec*.

Proposition 4 *WNec $^\tau$ is a derived rule.*

Proof. Pick a statement $\beta(\overline{M})$ with message terms \overline{M} . Let \overline{c} be all constants from *SEC* in \overline{M} . Assume

$$\vdash (\beta(\overline{M})[\overline{N}/\overline{c}])^\tau, \text{ all } \overline{N}$$

i.e.,

$$\vdash \overline{x} = \overline{M}[\overline{N}/\overline{c}] \rightarrow \beta(\overline{x}), \text{ all } \overline{N}$$

By infinitary rule (*Gen m*),

$$\vdash \forall \overline{m}. (\overline{x} = \overline{M}[\overline{m}/\overline{c}] \rightarrow \beta(\overline{x}))$$

By rule (*Nec*) and lemma 3.2,

$$\vdash \forall \overline{m}. \Box_A (\overline{x} = \overline{M}[\overline{m}/\overline{c}] \rightarrow \beta(\overline{x}))$$

By axiom (*mx*),

$$\vdash \forall \overline{y}. \Box_A (\overline{x} = \overline{M}[\overline{y}/\overline{c}] \rightarrow \beta(\overline{x}))$$

i.e.,

$$\vdash \Box_A \overline{x} = \overline{M}[\overline{y}/\overline{c}] \rightarrow \Box_A \beta(\overline{x})$$

By lemma 3.10, since \overline{c} includes all constants from *SEC* in \overline{M} ,

$$\vdash \Box_A \overline{y} \rightarrow \overline{x} = \overline{M}[\overline{y}/\overline{c}] \rightarrow \Box_A \beta(\overline{x})$$

i.e.,

$$\vdash \overline{y} = \overline{N} \rightarrow \Box_A \overline{y} \rightarrow \overline{x} = \overline{M}[\overline{N}/\overline{c}] \rightarrow \Box_A \beta(\overline{x})$$

i.e.,

$$\vdash (\Box_A \overline{N} \rightarrow \Box_A \beta(\overline{M})[\overline{N}/\overline{c}])^\tau$$

□

We proceed to derive $(MMR)^\tau$ and $(Fresh)^\tau$ using $WNec^\tau$. Let BAN be the conjunction of the following four assumptions:

$$\begin{aligned} \forall x. A \text{ sees } x \rightarrow \Box_A A \text{ sees } x & \quad \exists x. \neg \Box_A x \wedge \neg \text{good}_G x \\ \exists x. \neg \Box_A x \wedge \neg A \text{ sees } x & \quad \exists x. \neg \Box_A x \wedge \neg \text{fresh } x \end{aligned}$$

Trivially, an interpreted system \mathcal{I} satisfies the first conjunct of BAN if, and only if, $\rho(I(A \text{ sees}, s)) \subseteq I(A \text{ sees}, s')$ whenever $s \sim_A^\rho s'$ in \mathcal{I} . Roughly following [5], we assume $\text{good}_G x$ abbreviates: $\forall y. \bigvee_{A \in \mathcal{A}} A \text{ sees } \text{enc}(y, x) \rightarrow \bigvee_{B \in \mathcal{G}} B \text{ said } \text{enc}(y, x)$.

Corollary 7 *The following are theorems:*

1. $BAN \rightarrow (A \text{ sees } M \rightarrow \Box_A A \text{ sees } M)^\tau$
2. $BAN \rightarrow (A \text{ sees } M \rightarrow \Box_A M)^\tau$
3. $BAN \rightarrow (\Box_A \text{fresh } M \rightarrow \Box_A M)^\tau$
4. $BAN \rightarrow (\Box_A \text{good}_G K \rightarrow \Box_A K)^\tau$

Proof. (1) Immediate. (2), (3) and (4) From axiom $\Box 4$. □

Proposition 5 $\vdash BAN \rightarrow (MMR)^\tau$, assuming $\text{dec}(\text{enc}(M, K), K) \equiv M$.

Proof. By proposition 4,

$$(\Box_A \text{enc}(M, K) \rightarrow \Box_A A \text{ sees } \text{enc}(M, K) \rightarrow \Box_A \bigvee_{A'} A' \text{ sees } \text{enc}(M, K))^\tau$$

is a theorem. By corollary 7.1 and corollary 7.2,

$$BAN \rightarrow (A \text{ sees } \text{enc}(M, K) \rightarrow \Box_A \bigvee_{A'} A' \text{ sees } \text{enc}(M, K))^\tau$$

is a theorem. By the definition of *good*, $BAN \rightarrow (MMR)^\tau$ is a theorem. □

Proposition 6 $\vdash BAN \rightarrow (Fresh)^\tau$, if we add an additional axiom: $\text{fresh } t \rightarrow \text{fresh enc}(t, t')$.

Proof. By assumption,

$$(\text{fresh } M \rightarrow \text{fresh enc}(M, K))^\tau$$

is a theorem, for all M, K . By proposition 4,

$$(\Box_A M, K \rightarrow \Box_A (\text{fresh } M \rightarrow \text{fresh enc}(M, K)))^\tau$$

is a theorem. By corollary 7.3 and corollary 7.4, $\vdash BAN \rightarrow (Fresh)^\tau$. □

Finally, we observe that the embedding τ induces a new truth condition:

Proposition 7 *The following are equivalent:*

- $s \models_{\mathcal{I}} (\Box_A \beta(\overline{M}))^\tau$
- $\forall s' \in S : \forall \rho : s \sim_A^\rho s' \Rightarrow s' \models_{\mathcal{I}} \beta(\rho(\overline{M}))^\tau$

Proof. $s \models_{\mathcal{I}} (\Box_A \beta(\overline{M}))^\tau$ iff $s, V[\overline{x} \mapsto \overline{M}] \models_{\mathcal{I}} \Box_A \beta(\overline{x})$ iff $\forall s' \in S : \forall \rho : s \sim_A^\rho s' \Rightarrow s', \rho \circ V[\overline{x} \mapsto \overline{M}] \models_{\mathcal{I}} \beta(\overline{x})$. But, $s', \rho \circ V[\overline{x} \mapsto \overline{M}] \models_{\mathcal{I}} \beta(\overline{x})$ iff $s' \models_{\mathcal{I}} \beta(\rho(\overline{M}))^\tau$. \square

Providing a faithful semantics for BAN's epistemic modality has been a long-standing problem. The truth condition in proposition 7 is, essentially, a generalization to an arbitrary equational theory of the semantics proposed for BAN in [12, 13].

9.2 The SVO Modality

Protocol derivations in SVO [34], a successor to BAN, uses variables (represented as stars: \star, \star_x, \star_y , etc.) to refer *de re* to possibly undecrypted content. The derivations assume that seeing implies knowledge of seeing to the extent that the seen message can be decrypted. For instance, for the equational theory in example 1,

$$\begin{aligned} A \text{ sees enc}(\text{pair}(x, x'), pk(z)), A \text{ infers } z \rightarrow \\ \Box_A A \text{ sees enc}(\text{pair}(x, x'), pk(z)) \end{aligned} \quad (16)$$

Implications from seeing to knowledge of seeing, such as (16), are not justified by the proof system in [34], but the authors remark that it would be straightforward to capture such implications in an axiom. We propose the following axiom:

$$A \text{ sees } T \rightarrow \Box_A \text{ VAR}(T) \rightarrow \Box_A A \text{ sees } T \quad (SEE)$$

where T is any term without constants from SEC . The semantics in [34] does not support (16) or SEE . More generally, the semantics there does not support *de re* reference of variables. We show, however, that our semantics fits (16) and SEE . Let SVO be the following assumption:

$$\forall x. (A \text{ sees } x \rightarrow \Box_A A \text{ sees } x) \wedge \exists x. (\neg \Box_A x \wedge \neg A \text{ sees } x)$$

Proposition 8 *The following hold:*

1. $\vdash SVO \rightarrow A \text{ sees } x \rightarrow \Box_A x$.
2. $\vdash SVO \rightarrow SEE$
3. $\vdash SVO \rightarrow (16)$

Proof. (1): By axiom ($\Box 4$). (2): From lemma 3.10. (3): By equations in example 1, and axioms (\equiv) and ($\Box 3$) and proposition 8.1,

$$SVO, A \text{ sees enc}(\text{pair}(x, x'), pk(z)), \Box_A z \rightarrow \Box_A x, x', z$$

is a theorem. By proposition 8.2, $\vdash SVO \rightarrow (16)$. \square

10 Related Work

In [13] we use a propositional variant of the semantics presented here to account for BAN, and [12] gives a completeness result. For the relationship to [13], see proposition 7 and its explanation. The completeness result of this paper is stronger and much less ad hoc: The logic is richer, we avoid restriction to finite message spaces (which does not square well with most formal accounts of cryptography), we avoid the ad hoc "internal actions" used in [12] with no clear computational meaning, and the axiomatization is much less schematic and does not rely on a specific term algebra.

In the security literature, there are several different semantics for epistemic logic. Often, the "standard" multi-agent system semantics [16] is used, with identity as the equivalence on local states (cf. [21, 29, 30, 32]). But in connection with cryptography, identity is clearly inappropriate as equivalence. This is manifested in counter-intuitive validities such as $\exists x.A \text{ receives encrypt}(M, x) \rightarrow \Box_A \exists x.A \text{ receives encrypt}(M, x)$ (cf. *local state omniscience* in appendix A). Our semantics is most closely related to the AT-semantics [4, 5]. The AT semantics, which applies only to symmetric encryption, reduces indistinguishability to identity on expressions that use a fixed symbol \Box to represent undecryptable sub-expressions, and thereby loses all information about undecryptable data, including knowledge that may have been obtained indirectly, for instance, as a result of the protocol being executed (cf. the examples in section 6, also the unsoundness of BAN's message meaning rule in AT). We emphasize that there are no completeness results for AT-semantics, or its variants (cf. [34]). Moreover, since AT-semantics, and its variants, follow basic Kripke semantics, they render agents cryptographically omniscient (1). On the other hand, approaches based on explicit knowledge avoid cryptographic omniscience but have other drawbacks, discussed briefly in section 1. The compositional protocol logic of Durgin et al [15] uses knowledge only in terms of Dolev-Yao type message deduction. The role of permutations in our semantics is slightly reminiscent of [8, 35]. A version of theorem 3 for an AT-style semantics was proposed by S. Kramer (private correspondence).

The application of epistemic logic to cryptographic protocol analysis goes back to BAN logic [10]. Our protocol examples are, more directly, inspired by anonymity specifications in [21] and specifications for the SET protocol in [7]. We refer to [23] for a comprehensive dictionary of epistemic security specifications.

Interaction axiom ($\Box 3$), and, to a lesser degree, interaction axiom ($\Box 1$), are reminiscent of BAN-style proof systems. However, BAN-style proof systems contain only ad hoc rules specific to concrete predicates.

11 Concluding Remarks

One issue left open by our work is the role of the de dicto quantifier $\forall m$. We have been unable to obtain completeness for a compact logic which does not use

this quantifier. A candidate omega-rule is:

$$\frac{x = M \rightarrow F, \text{ all } M \in \mathcal{T}}{\forall x.F}.$$

However, it is difficult to see how to obtain a lemma corresponding to lemma 7 (with a suitably adjusted definition of omega-completion).

We can show that the de dicto quantifier adds to the expressive power. Let $\Sigma = \mathcal{A} = \{A\}$, i.e., let there be only one public operator, namely the agent identifier A , and let \equiv be identity on ground terms.

Proposition 9 *No statement free of the de dicto quantifier is equivalent to $\exists m.\exists x.x \neq A \wedge \Box_A x = m$.*

Proof. In appendix B. □

Our semantics is formulated in a counterpart semantics framework, although the choice of framework is, to some extent, a matter of taste. It is possible to reformulate the semantics in the framework of first-order intensional logic [9]. In such a framework, variables denote intensions, i.e., functions from states to individuals. In our setting, individuals are messages, and intensions are terms built from store locations and operators, such as the s -terms of section 2. However, reformulating our logic as a first-order intensional logic would, it seems, make security specifications more complex. A statement $\Box_A F(x)$ in our logic translates to, it seems, something like $\exists y.x = y \wedge A\text{-term}(y) \wedge \Box_A F(y)$, where $A\text{-term}$ is a predicate which applies to an intension if that intension is built from feasibly computable operators and store locations A can observe. An additional intension y is needed, since the intension x might be built from store locations not observed by A . As a result, the translation induces extra nesting of quantifiers and modalities. To illustrate, the statement $\Box_B \Box_A F(x)$ translates to $\exists y.x = y \wedge B\text{-term}(y) \wedge \Box_B \exists z.z = y \wedge A\text{-term}(z) \wedge \Box_A F(z)$.

The proposed logic is static only: It expresses properties of states, but not of computations. In the future, we plan to extend the completeness result to include temporal modalities, and to link to concepts in information flow security, and to behavioral equivalences for applied pi.

References

- [1] M. Abadi, M. Baudet, and B. Warinschi. Guessing attacks and the computational soundness of static equivalence. In *FoSSaCS'06*, pages 398–412, 2006.
- [2] M. Abadi and V. Cortier. Deciding knowledge in security protocols under equational theories. *Theor. Comput. Sci.*, 367(1-2):2–32, 2006.
- [3] M. Abadi and A. D. Gordon. A bisimulation method for cryptographic protocols. *Nordic J. of Computing*, 5(4), 1998.
- [4] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002.
- [5] M. Abadi and M. Tuttle. A semantics for a logic of authentication. In *PODC'91*, pages 201–216. ACM Press, August 1991.

- [6] R. Accorsi, D. A. Basin, and L. Viganò. Towards an awareness-based semantics for security protocol analysis. *Electr. Notes Theor. Comput. Sci.*, 55(1), 2001.
- [7] N. Agray, W. van der Hoek, and E. P. de Vink. On BAN logics for industrial security protocols. In *CEEMAS*, pages 29–36, 2001.
- [8] P. Bieber. A logic of communication in hostile environments. In *CSFW III*, pages 14–22. IEEE Computer Society Press, 1990.
- [9] T. Brauner and S. Ghilardi. First-order modal logic. In F. W. Patrick Blackburn, Johan van Benthem, editor, *Handbook of Modal Logic: Volume III*. Elsevier, 2006.
- [10] M. Burrows, M. Abadi, and R. M. Needham. A logic of authentication. *ACM Trans. Comput. Syst.*, 8(1):18–36, 1990.
- [11] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981.
- [12] M. Cohen and M. Dam. A completeness result for BAN logic. In *M4M05*, 2005.
- [13] M. Cohen and M. Dam. Logical omniscience in the semantics of BAN logic. In *FCS05*, pages 121–132, 2005.
- [14] G. Corsi. Counterpart semantics. a foundational study on quantified modal logics. Research rept PP-2002-20, ILLC, 2002.
- [15] N. Durgin, J. Mitchell, and D. Pavlovic. A compositional logic for proving security properties of protocols. *J. Comput. Secur.*, 11(4):677–721, 2004.
- [16] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.
- [17] C. Fournet and M. Abadi. Mobile values, new names, and secure communication. In *POPL’01*, pages 104–115, 2001.
- [18] D. Gabbay, V. Shehtman, and D. Skvortsov. Quantification in nonclassical logic. 2006. Manuscript.
- [19] J. W. Garson. Quantification in modal logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic: Volume II: Extensions of Classical Logic*. Reidel, 1984.
- [20] O. Goldreich. *Foundations of Cryptography*, volume Basic Tools. Cambridge University Press, 2001.
- [21] J. Halpern and K. O’Neill. Anonymity and information hiding in multiagent systems. In *CSFW’03*, pages 75–88, 2003.
- [22] J. Y. Halpern and R. Pucella. Modeling adversaries in a logic for security protocol analysis. In *FASec*, pages 115–132, 2002.
- [23] S. Kramer. Logical concepts in cryptography. Cryptology ePrint Archive, Report 2006/262, 2006. <http://eprint.iacr.org/>.
- [24] D. Lewis. Counterpart theory and quantified modal logic. *Journal of Philosophy*, 65:113–126, 1968.
- [25] A. Lomuscio and B. Wozna. A combination of explicit and deductive knowledge with branching time: Completeness and decidability results. In *DALT*, pages 188–204, 2005.
- [26] Mastercard and VISA. SET Secure Electronic Transaction Specification. 1997.
- [27] R. Parikh and R. Ramanujam. Distributed processes and the logic of knowledge. In *Logic of Programs*, pages 256–268, 1985.
- [28] R. Parikh and R. Ramanujam. A knowledge based semantics of messages. *Journal of Logic, Language and Information*, 12(4):453–467, 2003.
- [29] R. Pucella. *Reasoning about Resource-Bounded Knowledge: Theory and Application to Security Protocol Analysis*. Ph.D. Thesis, Cornell University, 2004.
- [30] R. Ramanujam and S. P. Suresh. Deciding knowledge properties of security protocols. In R. van der Meyden, editor, *TARK*, pages 219–235. National University of Singapore, 2005.

- [31] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1), 1998.
- [32] K. S. Ron van der Meyden. Symbolic model checking the knowledge of the dining cryptographers. In *CSFW'04*, 2004.
- [33] A. Sabelfeld and A. C. Myers. Language-based information-flow security. *IEEE J. Selected Areas in Communications*, 21(1):5–19, Jan. 2003.
- [34] P. F. Syverson and P. C. van Oorschot. A unified cryptographic protocol logic. NRL Publication 5540-227, Naval Research Lab, 1996.
- [35] M.-J. Toussaint and P. Wolper. Reasoning about cryptographic protocols. In J. Feigenbaum and M. Merritt, editors, *Distributed Computing and Cryptography*, volume 2 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 245–262. American Mathematical Society, 1989.

A Abstract Correspondence Results

The relativized indistinguishability relation \sim_A^ρ was defined by certain conditions on substitutions ρ . In this section, we provide logical characterizations of these, and some other, conditions. Throughout the present section, we assume a system $\mathcal{S} = \langle LOC, S, | \rangle$ and an arbitrary accessibility $\sim_A \subseteq S \times (\mathcal{T}_\equiv \longrightarrow \mathcal{T}_\equiv) \times S$.

Proposition 10 $s \sim_A^\rho s'$ implies that

1. ρ is injective
2. ρ is surjective
3. $\rho(f(\overline{M})) = f(\overline{\rho(M)})$, if all $M_i \in \text{Inferable}(s|A)$
4. $\rho \circ s|A = s'|A$

respectively, if and only if,

1. $\models_S x \neq y \rightarrow \Box_A x \neq y$
2. $\models_S \forall x. \Box_A F \rightarrow \Box_A \forall x. F$
3. $\models_S y = f(\overline{x})$, A infers $\overline{x} \rightarrow \Box_A y = f(\overline{x})$
4. $\models_S @_l x \rightarrow \Box_A @_l x$, $l \in LOC|A$

respectively.

Refer to the schema in proposition 10.4 as *local state introspection*. The first two correspondences above are well-known in counterpart semantics (cf. [14]), and so is the following correspondence for cryptographic omniscience (1).

Proposition 11 $s \sim_A^\rho s'$ implies $\rho(M) = M$, if and only if, \mathcal{S} satisfies (1).

By proposition 10.4 and proposition 11, we obtain the standard multi-agent system semantics [16], namely basic Kripke semantics [9] with local state identity as the Kripkean accessibility relation, if, in our framework, we define the relativized accessibility relation for A as the most inclusive \sim_A which validates cryptographic omniscience (1) as well as local state introspection (i.e., the schema in proposition 10.4). The combination of cryptographic omniscience and local state introspection leads to *local state omniscience*: $\@_l M \rightarrow \Box_A \@_l M$ for $l \in LOC|A$. In multi-agent system semantics with a different notion of state than ours, local state omniscience manifests itself in counter-intuitive validities such as $\exists x. A \text{ receives } enc(M, x) \rightarrow \Box_A \exists x. A \text{ receives } enc(M, x)$.

Finally, we provide a correspondence result for $(\Box 4^*)$, by which we mean the schema that results from axiom $(\Box 4)$ if we replace $\Box_A x$ with $A \text{ infers } x$. We say that non-deducible messages are anonymous, if $s \sim_A^\pi s$ whenever π is a permutation on $X \subseteq \overline{Inferable}(A, s)$ and X is finite; Here, $s \sim_A^\pi s$ means that π can be extended to a substitution ρ , defined on all messages, such that $s \sim_A^\rho s$.

Proposition 12 *Non-deducible messages are anonymous, if and only if, \mathcal{S} satisfies axiom $(\Box 4^*)$.*

Proof. Only-if direction: Straight-forward. If direction: Assume non-deducible messages are not anonymous, i.e., there is a state $s \in \mathcal{S}$ and finite $X \subseteq \overline{Inferable}(A, s)$ and permutation π on X such that $s \not\sim_A^\pi s$. Pick $x_1, \dots, x_n, z_1, \dots, z_n \in VAR$ and assignment V such that $Domain(\pi) = \{V(x_1), \dots, V(x_n)\}$ and $V(z_i) = \pi(V(x_i))$ and $V(x_i) \neq V(x_j)$ for all $i \neq j$. Choose an interpretation I and n -ary predicate p such that every n -tuple of messages satisfies p at every state $s' \in \mathcal{S}$, except that $\langle V(z_1), \dots, V(z_n) \rangle \notin I(p, s)$. Let \mathcal{I} be the interpreted system based on \mathcal{S} and I . Since $s \not\sim_A^\pi s$, we have $s, V \models_{\mathcal{I}} \Box_A p(x_1, \dots, x_n)$. Also, $s, V \models_{\mathcal{I}} \neg A \text{ infers } \bar{x}, \bar{z}$. Since $V(x_i) \neq V(x_j)$, we have $s, V \models_{\mathcal{I}} \bigwedge_{i,j} (x_i = x_j \leftrightarrow z_i = z_j)$. But, $s, V \not\models_{\mathcal{I}} p(z_1, \dots, z_n)$. \square

B Undefinability of De Dicto Quantifier

We proceed to prove proposition 9. Assume a set Γ of statements, free of de dicto quantifiers and closed under sub-statements. Assume two multi-agent systems \mathcal{S} and \mathcal{S}' , with state spaces S and S' respectively. A Γ -morphism from \mathcal{S} to \mathcal{S}' is a pair w, d such that:

1. $w : S \rightarrow S'$ is a surjective map
2. d_s is a permutation on \mathcal{T}_{\equiv} , for each $s \in S$
3. $V(t) = V(t') \Leftrightarrow (d_s \circ V)(t) = (d_s \circ V)(t')$, for all $(t = t') \in \Gamma$ and all assignments V in \mathcal{S}
4. $w(s) \sim_A^\rho w(s')$ in \mathcal{S}' iff $s \sim_A^{\rho'} s'$ in \mathcal{S} , where $\rho' = d_{s'}^{-1} \circ \rho \circ d_s$

Morphism condition (3) might appear tautological, but is not. As explained in section 5, $V(t) = V(t')$ need not imply that $(\rho \circ V)(t) = (\rho \circ V)(t')$.

Lemma 21 $s, V \models_S F$ iff $w(s), (d_s \circ V) \models_{S'} F$, for $F \in \Gamma$.

Proof. Straightforward induction on F . □

Next, we show that lemma 21 fails if Γ contains de dicto quantifiers. Roughly, if Γ contains de dicto quantifiers, the above proof fails because the induction step for statement $\forall m. F[m/x]$ requires the induction assumption for $F[M/x]$, for each ground term M . But, $F[M/x]$ need not be in Γ .

Let $\Sigma = \mathcal{A} = \{A\}$, i.e., there is only one public operator, the agent identifier A . Let \equiv be identity on ground terms. For any distinct $c, d \in SEC$, we construct two multi-agent systems $\mathcal{S}_{cd} = \langle LOC, S, | \rangle$ and $\mathcal{S}'_{cd} = \langle LOC, S', | \rangle$, defined as follows: $LOC = LOC|A = \{l_1, l_2\}$; $S = \{s_1, s_2\}$, where $s_1(l_1) = c$, $s_1(l_2) = d$, $s_2(l_1) = d$ and $s_2(l_2) = c$; Finally, $S' = \{s_1\}$.

Lemma 22 If no statement in Γ contains c or d then there is a Γ -morphism from \mathcal{S}_{cd} to \mathcal{S}'_{cd} .

Proof. We can define a Γ -morphism w, d as follows. Define $w : S \rightarrow S'$ such that $w(s_1) = w(s_2) = s_1$. Let d_{s_1} be identity on \mathcal{T}_{\equiv} . Let d_{s_2} be the permutation on \mathcal{T}_{\equiv} which maps c to d and conversely maps d to c , but leaves all other messages unchanged. □

Corollary 8 \mathcal{S}'_{cd} but not \mathcal{S}_{cd} satisfies $\exists m. \exists x. x \neq A \wedge \Box_A x = m$.

Proposition 9 No $\forall m$ -free statement is equivalent to $\exists m. \exists x. x \neq A \wedge \Box_A x = m$.

Proof. Pick any $\forall m$ -free statement F . Pick distinct $c, d \in SEC$ which do not occur in F . Let Γ be the set of sub-statements of F . By lemma 21 and lemma 22, F does not distinguish between \mathcal{S}'_{cd} and \mathcal{S}_{cd} . But, by corollary 8, the statement $\exists m. \exists x. x \neq A \wedge \Box_A x = m$ does distinguish the two systems. □