KTH Computer Science
and Communication

Logics of Knowledge and Cryptography

Completeness and Expressiveness

MIKA COHEN

Doctoral Thesis in Teleinformatics
Stockholm, Sweden 2007

TRITA-CSC-A 2007:11
ISSN-1653-5723
ISRN-KTH/CSC/A–07/11–SE
ISBN 978-91-7178-705-7

School of Computer Science and Communication
KTH
SE-100 44 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i teleinformatik fredagen den 15 juni 2007 kl. 10.00 i E2, Lindstedtsvägen 3, Kungl Tekniska högskolan, Stockholm.

Tryck: Universitetsservice US AB

Abstract

An understanding of cryptographic protocols requires that we examine the knowledge of protocol participants and adversaries: When a participant receives a message, does she know who sent it? Does she know that the message is fresh, and not merely a replay of some old message? Does a network spy know who is talking to whom?

This thesis studies logics of knowledge and cryptography. Specifically, the thesis addresses the problem of how to make the concept of knowledge reflect feasible computability within a Kripke-style semantics. The main contributions are as follows.

- A generalized Kripke semantics for first-order epistemic logic and cryptography, where the later is modeled using private constants and arbitrary cryptographic operations, as in the Applied Pi-calculus.
- An axiomatization of first-order epistemic logic which is sound and complete relative to an underlying theory of cryptographic terms, and to an omega-rule for quantifiers. Besides standard axioms and rules from first-order epistemic logic, the axiomatization includes some novel axioms for the interaction between knowledge and cryptography.
- Epistemic characterizations of static equivalence and Dolev-Yao message deduction.
- A generalization of Kripke semantics for propositional epistemic logic and symmetric cryptography.
- Decidability, soundness and completeness for propositional BAN-like logics with respect to message passing systems. Completeness and decidability are generalised to logics induced from an arbitrary base of protocol specific assumptions.
- An epistemic definition of message deduction. The definition lies between weaker and stronger versions of Dolev-Yao deduction, and coincides with weaker Dolev-Yao regarding all atomic messages. For composite messages, the definition withstands a well-known counterexample to Dolev-Yao deduction.
- Protocol examples using mixes, a Crowds style protocol, and electronic payments.

Sammanfattning

För att kunna förstå kryptografiska protokoll behöver vi fråga oss vilken kunskap protokolldeltagare och angripare tillägnar sig under protokollets gång. När en protokolldeltagare tar emot ett meddelande, behöver vi fråga oss: Vet hon vem som har skickat det? Vet hon om meddelandet är nytt eller återanvänt? Vet en nätverksspion vilka protokolldeltagare som just nu kommunicerar med varandra?

I den här avhandlingen undersöker vi logiker för kunskap och kryptografi. Vi behandlar frågan hur man kan få kunskapsbegreppet att reflektera praktisk beräkningsbarhet inom en Kripkeliknande semantik. Vi presenterar följande bidrag:

- En generaliserad Kripkesemantik för första ordningens epistemisk logik och kryptografi, där den senare representeras av privata konstanter och godtyckliga kryptografiska operationer, liksom i tillämpad pi-kalkyl.
- En axiomatisering av första ordningens epistemisk logik som är sund och fullständig relativt dels en underliggande teori om kryptografiska termer, dels en omega-regel för kvantifikatorer. Utöver standardaxiom och regler från första ordningens epistemisk logik inkluderar axiomatiseringen några nya axiom för samspelet mellan kunskap och kryptografi.
- Epistemisk karakterisering av statisk ekvivalens och Dolev-Yao meddelandededuktion.
- En generalisering av Kripkesemantik för epistemisk satslogik och symmetrisk kryptografi.
- Avgörbarhet, sundhet och fullständighet för BAN-liknande satslogik. Fullständighet och avgörbarhet lyfts till logik härledd från en godtycklig bas av protokoll antaganden.
- En epistemisk definition av meddelandededuktion. Definitionens omfång ligger mitt emellan svagare och starkare versioner av Dolev-Yao deduktion. För atomära meddelanden, sammanfaller definitionen med den svagare varianten av Dolev-Yao, medan den för sammansatta meddelanden motstår ett välkänt motexempel mot Dolev-Yao-deduktion.
- Protokollexempel som använder mixar, ett Crowds-liknande protokoll och elektronisk betalning.

> As we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns - the ones we don't know we don't know.

Donald Rumsfeld, former United States Secretary of Defense

Acknowledgements

First of all, I would like to thank Mads Dam, who has been an outstandingly good supervisor: Dynamic and effectual, yet always flexible and open. On a personal level, Mads's genial and relaxed style and his good sense of humor has made working together a great pleasure.

I also wish to thank the whole formal methods team for providing a friendly and enjoyable atmosphere as well as many interesting discussions over lunch. More specifically I would like to thank: Dilian Gurov for stepping in for Mads during his sabbatical in Australia, and for helping me to structure my work. Dilian has also become a good friend during the years. Irem Aktug, with whom I have been fortunate enough to share an office for almost four years. A supreme roommate and a reliable unix and tex support. In between the daily toils we have had many laughs. I have also, at different times, enjoyed sharing offices with Thom Birkeland, Wen Xu and Andreas Lundblad.

I am particularly grateful to my wife, Katie Asplund Cohen, for extensive and valuable discussions on how to clarify and formulate ideas; Without her help, my thesis would certainly be (even) less comprehensible. In every way, I am lucky to have such an extraordinary wife.

I am also grateful to my undergraduate supervisor Krister Segerberg for inspiring me to continue studying after my masters and for introducing me to the world of modal logic, and to Johan van Benthem for many intriguing and lengthy discussions during my visit as an undergraduate at ILLC.

I would further like to thank Karl Meinke for critical help on first-order logic, Michael O. Rabin for asking productive questions, Joachim Parrow for helping me to plan my research at an early stage, Simon Kramer for lengthy discussions regarding epistemic logic, Torben Braüner for prompt replies to my email bombardments and Mårten Trolin for taking time to answer my questions on cryptography.

I would like to thank my supervisor Mads, my wife Katie and my father Ian Cohen for comments and feedback on earlier drafts of this thesis.

Finally, I would like to thank both my parents for their generous support and encouragement.

Contents

Chapter 1

Introduction

Communication over the internet involves many security risks. When you order an item from a web store, there is a risk that your credit card details leak to unauthorized parties. When you download a piece of software or receive an e-mail, there is a risk that the software or e-mail does not originate from the party from which it purports to be. When you post a message through an instant messaging service, there is a risk that someone can track the message back to you, even if you yourself did not disclose your identity.

Security protocols are special programs that protect us against the security threats posed by "adversaries" present on a communication network. For example, a security protocol might ensure that the submitted credit card details remain confidential, or that the downloaded software originates from the source from which it claims to originate, or that messages cannot be tracked back to their sender.

However, it is not easy to design a security protocol: The designer has no prior knowledge of the way adversaries on the network will act, and therefore must consider how the protocol functions under all possible adversary behaviours. Some possible adversary strategies can easily be overlooked.

There is a need, therefore, for mathematical tools that will assist software developers in analysing security protocols and in uncovering otherwise unforeseen attacks. In this thesis, we contribute to the foundations for such tools.

1.1 Security Protocols

Security protocols are small distributed programs that provide security services to network communication. Most security protocols rely on one-way functions, i.e., functions that are easy to compute but infeasible to invert without additional information. In other words, infeasible computational resources are required to find the input which yields a given output. For example, a symmetric encryption scheme consists of two one-way functions, encryption *enc* and decryption *dec*, such

that:

$$dec(enc(M, K), K) = M \tag{1.1}$$

The encryption function takes a plaintext message M and a parameter, called the key, K, and produces a ciphertext $enc(M, K)$. The decryption function reverses the process, and recovers the original plaintext M from $enc(M, K)$ and the key K. Thus, if you see the ciphertext $enc(M, K)$ and you know the key K, you can extract the plaintext M from the ciphertext.

Using the symmetric encryption scheme, two agents A and B can communicate over a public network in a way that prevents any eavesdropping spy from learning what is being said. Assume that A and B share a secret key K. To send a secret message M to agent B over the network, agent A first encrypts M under K, and then sends the encryption to B:

$$A \longrightarrow B : enc(M, K)$$

(The notation $A \longrightarrow B : M$ means that agent A sends message M to agent B over the public network.) Upon receiving the encryption, agent B can decrypt it using K, and recover message M. A spy, who eavesdrops on the network traffic, might observe the encryption $enc(M, K)$ on the wire. But, since the spy does not know the key K, the spy cannot (with feasible computational resources) recover M from the encryption. Thus, M remains secret:

Secrecy Goal If B receives $enc(M, K)$, the spy does not know that B sees M.

Of course, if the spy eavesdrops on the whole network route between agents A and B, the spy can track the encryption $enc(M, K)$ travel from A to B. Consequently, the spy knows that A is talking to B; Their conversation is not anonymous.

However, more high-level security services, such as anonymity, can also be achieved by the same means, namely one-way functions. But, it may require a more complex communication protocol. As an illustration, consider the following protocol for anonymous communication within a group of agents sharing a secret key K. The group includes a special forwarding server, the *mix*, that receives as input a sequence of encryptions from group members $A_1, \ldots, A_n$:

$$\begin{aligned} A_1 &\longrightarrow mix : enc(M_1\, to\, B_1, K) \\ &\vdots \\ A_n &\longrightarrow mix : enc(M_n\, to\, B_n, K) \end{aligned}$$

where the encryption content $M_i\, to\, B_i$ signifies that the message M_i is intended for agent B_i. Using the shared key K, the mix decrypts each input $enc(M_i\, to\, B_i, K)$ and recovers the content $M_i\, to\, B_i$. Once the mix has received n encryptions, it sends each M_i to its specified destination B_i. But, the messages are sent in random

order:

$$\begin{array}{rcl} mix & \longrightarrow & B_{\pi(1)} : M_{\pi(1)} \\ \vdots & & \\ mix & \longrightarrow & B_{\pi(n)} : M_{\pi(n)} \end{array}$$

for some random permutation π on $\{1, \ldots, n\}$. The eavesdropping spy observes the encryption $enc(M_i\, to\, B_i, K)$ travel from agent A_i to the mix, and later observes M_i travel from the mix to agent B_i. But, since the spy lacks the key K, the spy cannot decrypt $enc(M_i\, to\, B_i, K)$ to recover M_i. Therefore - this is the idea behind the protocol - the spy is unable to link mix input $enc(M_i\, to\, B_i, K)$ to mix output M_i. If so, the protocol allows group member A_i to send message M_i to group member B_i, without the spy knowing who sent the message M_i:

Anonymity Goal 1 If agent B received message M, the spy does not know that M originated from A.

Anonymity Goal 2 If agent B received message M, the spy does not know that M did not originate from A.

1.2 Formal Cryptography

Security protocols are notoriously error prone. Even for simple protocols, like the mix-based protocol sketched above, it is extremely difficult to foresee all possible ways in which the adversary may act in order to subvert the protocol. For example, unless you have seen a similar protocol before, you may easily overlook the fact that the above protocol fails to meet its goals if the mix accepts the same input twice: If the spy replays an input, exactly two inputs and exactly two outputs are identical, hence the spy can link the two inputs to the two outputs, and consequently the anonymity goals fail.

Over the past decades a number of mathematical techniques have been developed that help protocol designers analyze the security of their designs. In the so-called computational approach to security, protocol analysis is based on complexity and probability theory (cf. [13, 38]): A protocol is secure if an attacker, in the form of an arbitrary randomized polynomial-time Turing machine, has only negligible probability of success. Proofs in this approach are, however, often subtle and error-prone, and intuition is easily lost in mathematical details.

The *formal approach* to security protocols, also known as the *Dolev-Yao* approach was initiated in [28]. Here, one-way functions, such as encryption and decryption, are idealized in order to obtain models that are more intuitive and tractable, with potentially better support for automation. Roughly, cryptography is treated as an abstract data type: It is assumed that cryptographic objects can only be manipulated using a restricted set of operations, which are governed by

some simple algebraic laws. For instance, in the case of symmetric cryptography, it might be assumed that messages are only manipulated through the encryption function *enc* and the decryption function *dec*, which satisfy the equation (1.1). Of course, real encryption and decryption functions satisfy other equalities, besides those induced by (1.1), and real encryptions (bitstrings) can be manipulated by any number of different operations. On the other hand, many attacks on protocols do not depend on the mathematical details of the cryptographic functions employed in the protocol, but instead are due to the way these functions are used in communication between agents, i.e., due to the protocol logic, as is the case with the above replay attack (cf. [18]). A well-known example is the man-in-the-middle attack on the Needham-Schroeder Public Key Protocol, found by Gawin Lowe more than fifteen years after the protocol was introduced [61].

This thesis belongs primarily to the latter school of formal, as opposed to computational, security protocol analysis. However, recent work has begun to bridge the gap between the formal and computational approaches, with results showing that protocols that are secure in a certain formal model are also secure in a certain computational model (cf. [1, 5, 6, 8, 25, 63, 64] and section 1.7 below).

1.3 Message Deduction, Indistinguishability and Epistemic Logic

An understanding of security protocols requires that we examine the knowledge of agents: When an agent receives a message, does she know who sent it? Does she know that the message is fresh, and not merely a replay of some old message? Does a network spy know who is communicating with whom on the network? Consequently, a definition of knowledge is a central concept in several formal approaches to security protocol analysis.

A simple and frequently used notion of knowledge is *Dolev-Yao message deduction* [28]. Here, the information content is messages: An agent A deduces ("knows") a message M, if agent A on its own can obtain M through feasibly computable operations, starting from directly observed messages (such as messages that A received and keys that A generated). For example, if agent A observes both the symmetric encryption $enc(M, K)$ and the key K then A deduces the message M. Some security services can be formulated directly in terms of message deduction. For instance, the secrecy goal in section 1.1 might be approximated:

$$B \textit{ receives } enc(M, K) \rightarrow \neg \textit{spy deduces } M$$

However, message deduction is a very limited form of knowledge. Security services, besides some simple forms of secrecy, are not easily formalized in terms of message deduction. Consider, for instance, the anonymity goals in section 1.1. Clearly, anonymity does not mean that the spy cannot deduce the transmitted message M or deduce the agent name A, since by assumption, the spy sees the message on the wire and agent names are public knowledge.

A richer notion of knowledge can be obtained using indistinguishability relations. In process algebra (cf. [4, 32, 71]) and information flow analysis (cf. [73]), knowledge is commonly defined in terms of an observational equivalence of programs: A program successfully hides a condition if varying the condition has no observable effect. For example, the anonymity goals in section 1.1 may be captured by stating that, roughly speaking, an instance of the protocol where agent A sends message M to agent B and agent A' sends message M' to agent B' is observationally equivalent (from the point of view of the spy) to an instance where agent A sends M' to B' and A' sends M to B.

Indistinguishability-based knowledge is used also in opacity theory (cf. [15, 49]), where knowledge is defined in terms of indistinguishability of protocol runs: A condition F is opaque ("hidden") to an observer if for every protocol run s satisfying the condition F there is another protocol run s' which does not satisfy the condition F, yet s is indistinguishable from s' (to the observer). For instance, the first anonymity goal in section 1.1 means that the condition:

$$B \textit{ received } M \wedge A \textit{ originated } M \tag{1.2}$$

is opaque to the spy, i.e., for every protocol run s satisfying (1.2), there is a run s', indistinguishable from s to the spy, that fails (1.2).

Epistemic logic – "the logic of knowledge" – is closely related to opacity theory (cf. [31, 66]). Epistemic logic extends classical logic with a so called epistemic modality $\Box_A$ expressing knowledge of agent A: The formula $\Box_A F$ is true if agent A knows that condition F holds. Formally, $\Box_A F$ holds at a protocol run s if condition F holds at all indistinguishable runs s':

$$s \models \Box_A F \Leftrightarrow \forall s' : s \sim_A s' \Rightarrow s' \models F \tag{1.3}$$

where s and s' range over runs (computation points) of the given protocol, and $s \sim_A s'$ means that runs s and s' are indistinguishable to agent A. A protocol is said to satisfy a logical formula if every run of the protocol satisfies the formula. Thanks to the epistemic modalities, informal, high-level descriptions of security services translate directly to epistemic logic. (We refer to [55] for a comprehensive dictionary of security specifications in epistemic logic.) For instance, the two anonymity goals in section 1.1 can be formalized, respectively, as follows:

$$B \textit{ received } M \rightarrow \neg\Box_{spy} A \textit{ originated } M \tag{1.4}$$

$$B \textit{ received } M \rightarrow \neg\Box_{spy}\neg A \textit{ originated } M \tag{1.5}$$

In this thesis, we investigate the relationship between message deduction, indistinguishability and epistemic modalities in contexts that involve cryptography.

1.4 Standard Multi-Agent Semantics

The semantics (1.3), known as Kripke semantics, is the standard semantics for the epistemic modality. Clearly, if epistemic logic is to be used for describing security

protocols, i.e., programs, its Kripke semantics needs to be *grounded* [84], in the sense that when a program (for instance, a distributed JAVA program) and a logical formula are given, the semantics determines if the program satisfies the formula. In effect, this means that the indistinguishability relation $\sim_A$ needs to be defined in terms of runs (computations) of programs, rather than taken as a primitive.

Kripke semantics (1.3) has a standard form of grounded instantiation:

$$s \sim_A s' \Leftrightarrow s|A = s'|A \tag{1.6}$$

where $s|A$ is the local state of agent A at the run (computation point) s. Intuitively, the local state $s|A$ contains the evidence available to agent A at s. The precise definition of a local state may vary somewhat depending on the notion of run (computation point) used. For example, if runs s are sequences of input and output actions, the local state $s|A$ might be the sub-trace of s of those actions performed by agent A. Combining the Kripkean truth condition (1.3) with (1.6), we obtain

$$s \models \Box_A F \Leftrightarrow \forall s' : s|A = s'|A \Rightarrow s' \models F \tag{1.7}$$

where s and s' range over computation points (runs) of the underlying protocol. Thus, an agent knows a fact if its local evidence forces the fact to hold in the given set of computations.

Today, the standard multi-agent semantics (1.7) is a mature research area; There are many results and tools for model checking, i.e., for determining if (the set of runs of) a given program satisfies a logical formula (cf. [35, 50, 58, 67, 80, 82]), and there are numerous completeness results (cf. [33, 46, 81]). In fact, most model checking techniques and completeness results concern epistemic logics extended with temporal modalities, such as next-time modalities for "Next it will be the case that" and future-time modalities for "It will always be the case that". Axioms for the interaction between epistemic and temporal modalities depend on the specifics of the local state projection $|$ (for instance, whether the local state grows over time), but also on other factors, such as whether communication is synchronous or not.

Starting in the early 90's, the standard multi-agent semantics (1.7) has been applied extensively to computer security (cf. [12, 41, 42, 43, 44, 45, 82, 60, 75]). The focus has been on anonymity properties, formalized in the manner of (1.4) and (1.5) above.

1.5 The Local State Omniscience Problem

However, the standard multi-agent semantics (1.6) is problematic for security protocols that rely on one-way functions, such as encryption. Such protocols concern knowledge in the sense of *resource bounded knowledge*, i.e., knowledge which is restricted by limited computational powers for cryptographic calculations. Thus, in specifications (1.4) and (1.5), the intended meaning of the epistemic modality $\Box_{spy}$ is something like "With feasible computational resources for cryptographic calculations, the spy can infer that". The standard semantics does not reflect resource

limitations for cryptographic calculations, since in the standard semantics, agents know every property of their own local state, even those properties that require infeasible cryptographic resources to calculate; Agents in the standard semantics are *local state omniscient*. For example, assume that the local state of agent A records the messages A sends and receives. Then, agent A knows if message M is the encryption-content of some received encryption:

$$A \, received \, enc(M, K) \models \Box_A \exists x. A \, received \, enc(M, x)$$

But, if A lacks the relevant key K, it might require infeasible computational resources for A to calculate that M is part of a received encryption.

1.6 BAN Logic

Due to local state omniscience in the standard multi-agent semantics, applications of epistemic logic to cryptography have mostly been based on proof systems, rather than semantics. The first proof system combining epistemic logic and cryptography, known as BAN logic, appeared in the late 80's. Since then, BAN logic has spawned many extensions and variations (cf. [7, 9, 27, 39, 51, 52, 57, 74, 77, 78, 83]). In BAN-style analyses of a security protocol, the security goal – in most cases an authentication goal – is formulated as a statement in epistemic logic. For instance:

$$\Box_{bank} \, customer \; sent \, M$$

$$\Box_{bank} \, \Box_{merchant} customer \; sent \, M$$

The security goal is then derived in the proof system, starting from more self-evident assumptions about what happens during protocol execution, such as what messages are sent, received or generated:

$$bank \; received \, M, \; bank \; generated \; nonce \, N$$

However, a BAN-style proof system with no semantics is unsatisfactory. Without a semantics, it is unclear what is established by a derivation in the proof system: A proof system is merely a definition, and as such it needs further justification. Moreover, the restriction to proof system based protocol analysis is unfortunate. Indeed, elsewhere in epistemic logic, semantically based techniques for analysing protocols, for instance model checking (section 1.4), are preferred (cf. [47]).

1.7 Dolev-Yao Indistinguishability

There have been a few attempts at adjusting the standard multi-agent semantics (1.7) to BAN-like logics. The style of adjustment was introduced in AT semantics [7], which replaces the test for local state identity in (1.6) by a test for local state indistinguishability:

$$s \sim_A s' \Leftrightarrow s|A \sim s'|A \tag{1.8}$$

where $\sim$ is an indistinguishability relation on local states, each local state being, essentially, a sequence of messages. Approximately, two message sequences are indistinguishable if they are identical *up to* content inside encryptions for which the decryption key cannot be deduced. For instance, for symmetric cryptography, the sequences $K \cdot enc(\text{"}Yes\text{"}, K)$ and $K \cdot enc(\text{"}No\text{"}, K)$ are distinguishable, since the decryption key K can be (trivially) deduced from each sequence. On the other hand, the sequences $enc(\text{"}Yes\text{"}, K) \cdot enc(\text{"}No\text{"}, K)$ and $enc(\text{"}No\text{"}, K) \cdot enc(\text{"}Yes\text{"}, K)$ are indistinguishable, since the encryptions cannot be opened.

The original indistinguishability $\sim$ in [7] applies to symmetric cryptography only. But, some later variants extend the relation to forms of asymmetric cryptography (cf. [15, 24, 77, 83]). Collectively, these indistinguishability relations are referred to as *Dolev-Yao indistinguishability* relations, since they are all based on formal (i.e., Dolev-Yao style) cryptography.

The common intuition behind Dolev-Yao indistinguishability relations is that two message configurations are indistinguishable if every experiment — based on a restricted set of available operations – produces the same result at both message configurations. This intuition is made explicit in *static equivalence* [32], a general form of indistinguishability which has recently received special focus. Static equivalence is a relation between stores, i.e., mappings from store locations $l_1, l_2, l_3, \ldots$ to messages. Two stores σ and σ' are statically equivalent if they satisfy the same equality tests. I.e., σ and σ' are statically equivalent, $\sigma \approx \sigma'$, if

- $\sigma(l_1) = \sigma(l_2) \Leftrightarrow \sigma'(l_1) = \sigma'(l_2)$.
- $hash(\sigma(l_1)) = \sigma(l_3) \Leftrightarrow hash(\sigma'(l_1)) = \sigma'(l_3)$.
- $decrypt(\sigma(l_1), \sigma(l_2)) = \sigma(l_4) \Leftrightarrow decrypt(\sigma'(l_1), \sigma'(l_2)) = \sigma'(l_4)$.
- And similarly, for all equality tests built from store locations and feasibly computable operations.

Static equivalence is defined with respect to an arbitrary collection of feasibly computable operators – symmetric encryption and decryption, asymmetric encryption and decryption, random encryption and decryption, digital signatures, hash functions, etc. – given by an equational theory. To model random asymmetric encryption, for example, one might assume the weakest equational theory satisfying the following equation:

$$dec(enc(M, pk(K), N), K) = M$$

Informally, *pk* produces a public key from a private seed K, *enc* encrypts the first argument M using the second $pk(K)$ as encryption key and the third argument N as a random seed, and *dec* decrypts the first argument using the second argument as decryption key. Depending on the specific choice of equational theory, static equivalence can be decidable (cf. [2]).

Recently, computational soundness results linking Dolev-Yao indistinguishability relations to computational models of cryptography have received attention (cf.

[1, 5, 6, 8, 25, 63, 64]). Roughly, it is shown that, given some assumptions on the cryptographic primitives, if two message configurations are Dolev-Yao indistinguishable, then their interpretations in the computational setting are indistinguishable to a computational adversary. This line of work was initiated in [6] with a computational soundness result for the original AT-indistinguishability [7]. Recently, computational soundness results have been obtained also for static equivalence, for instance [1] for a language involving symmetric and asymmetric cryptography.

The AT-indistinguishability was first introduced in the context of Kripke semantics, to provide a semantics for a BAN-like logic; Combining the Kripkean truth condition (1.3) with (1.8), one obtains:

$$s \models \Box_A F \Leftrightarrow \forall s' : s|A \sim s'|A \Rightarrow s' \models F \tag{1.9}$$

Subsequent work on Dolev-Yao indistinguishability relations has, with a few exceptions (cf. [77, 83]), been outside the framework of epistemic logic. There are some (sketched) soundness results for BAN logic derivates ([7, 77, 83]) with respect to AT-style semantics (1.9), but no more substantial results. Most critically, there are no completeness results; Completeness for BAN-like logics has remained an open problem. Completeness results are important, even if we are not interested in proof system based protocol analysis, since completeness results constitute strong evidence that the semantics behaves as expected.

1.8 The Logical Omniscience Problem

Any Dolev-Yao indistinguishability might serve as a basis for Kripke semantics (1.3). For instance, assuming that local states $s|A$ are stores, we can consider two computation points s and s' indistinguishable to agent A if $s|A$ and $s'|A$ are statically equivalent:

$$s \sim_A s' \Leftrightarrow s|A \approx s'|A \tag{1.10}$$

However, no matter what indistinguishability relation $\sim_A$ is used in Kripke semantics – be it AT-style indistinguishability (1.8), static equivalence based indistinguishability (1.10) or whatever - Kripke semantics (1.3) is subject to the so called *logical omniscience problem*. In Kripke semantics (1.3), agents know all the logical consequences of what they know, whether or not these consequences can be computed with feasible resources for cryptographic computations:

$$F \models F' \implies \Box_A F \models \Box_A F' \tag{1.11}$$

For instance, from the validity:

$$\models enc(M, K) \mathit{contains}\, M$$

logical omniscience yields:

$$\models \Box_A enc(M, K) \mathit{contains}\, M \tag{1.12}$$

Logical omniscience is problematic for security specifications such as (1.4) and (1.5), even though they do not directly describe knowledge of cryptographic relationships. Consider an instance of the anonymity protocol where the message M_i is hidden from the spy under the shared key K:

$$\begin{array}{rcl} A_1 & \longrightarrow & mix : enc(enc(M_1, K)\, to\, B_1, K) \\ \vdots & & \\ A_n & \longrightarrow & mix : enc(enc(M_n, K)\, to\, B_n, K) \\ mix & \longrightarrow & B_{\pi(1)} : enc(M_{\pi(1)}, K) \\ \vdots & & \\ mix & \longrightarrow & B_{\pi(n)} : enc(M_{\pi(n)}, K) \end{array}$$

for some random permutation π on $\{1, \ldots, n\}$. Anonymity goal (1.4) now becomes:

$$B\ received\, enc(M, K) \rightarrow \neg \Box_{spy} A\ originated\, enc(M, K) \tag{1.13}$$

As the replay attack on the protocol illustrates, even if the protocol implementation achieves secrecy:

$$B\ received\, enc(M, K) \rightarrow \neg \Box_{spy} exists\, M$$

the protocol implementation may still fail to provide anonymity, i.e., specification (1.13) could fail. However, logical omniscience contradicts these intuitions, since logical omniscience produces:

$$\Box_{spy} A\ originated\, enc(M, K) \models \Box_{spy} exists\, M$$

from the validity:

$$A\ originated\, enc(M, K) \models exists\, M$$

As the logical omniscience problem highlights, accounting for the epistemic modality in cryptographic contexts is not merely a question of finding an appropriate indistinguishability relation; Logical omniscience follows in Kripke semantics, no matter which indistinguishability relation is chosen.

1.9 Syntactic Approach to Knowledge

The most common response to logical omniscience in epistemic logic is to abandon Kripke-style semantics for a more syntactic account of knowledge (cf. [31]):

$$s \models \Box_A F \Leftrightarrow F \in \chi(s|A) \tag{1.14}$$

where the function χ associates a set $\chi(s|A)$ of statements to each local state $s|A$. The knowledge function χ is left open, to be adjusted for each specific protocol under consideration.

There are a few different intuitions motivating (1.14) in the literature. In some cases, the intuition is simply that $\chi(s|A)$ is the "knowledge base" available at $s|A$, given as an explicit list of statements (cf. [29]). In other instances, the intuition is that $(s|A)$ is the set of statements that A is "aware" of at s, perhaps generated from a base of primitive statements of which the agent is aware (cf. [30]). In these instances, an agent is considered to explicitly know a statement if the agent is aware of the statement and the agent knows the statement in the sense of standard multi-agent semantics (1.7). In [59], completeness is shown for a logic combining the awareness-modality (1.14), interpreted by arbitrary χ, the standard multi-agent modality (1.7) and temporal modalities. In yet other cases, the intuition is that $\chi(s|A)$ reflects the knowledge algorithm available at $s|A$: An agent knows a fact if the agent can compute the fact using the available knowledge algorithm (cf. [40]).

Often, the knowledge function χ is defined by way of an inference relation (cf. [54, 69]): $\chi(s|A)$ is the set of statements inferable from the base $s|A$. For certain statements F that are about the local state $s|A$ itself, it seems possible to provide an inference relation for $F \in \chi(s|A)$ which is, at least approximately, intuitively complete. In [43, 60], for instance, the knowledge function χ lifts the Dolev-Yao message deduction relation (section 1.3) to statements approximately along the following lines:

$$\begin{aligned} s \models A \mathit{\ received} M &\implies A \mathit{\ has} M \in \chi(s|A) \\ A \mathit{\ has\ pair}(M, M') \in \chi(s|A) &\implies A \mathit{\ has} M \in \chi(s|A) \\ A \mathit{\ has\ pair}(M, M') \in \chi(s|A) &\implies A \mathit{\ has} M' \in \chi(s|A) \\ A \mathit{\ has\ enc}(M, K),\ A \mathit{\ has} K \in \chi(s|A) &\implies A \mathit{\ has} M \in \chi(s|A) \end{aligned}$$

where A *has* M means that M occurs (as a sub-message) inside the local state of agent A.

However, as soon as we are interested in what one part (agent) of a system knows about another part (agent), the knowledge function χ has no generally applicable definition (which even approximately is intuitively complete). We argue that leaving the interpretation of χ open, begs to some extent the verification question the logic is supposed to help us with, namely to determine what facts agents (protocol participants and adversaries) are able to infer during the execution of the protocol. Consider, for example, the anonymity specification (1.4) for the mix-based protocol in section 1.1. To apply the semantics (1.14), we need to first to lay down the conditions (for the given protocol) under which A *originated* $M \in \chi(s|spy)$. In other words, we need to know the truth of the specification (1.4) itself.

1.10 Knowledge *De Re* and Knowledge *De Dicto*

In philosophical logic, a distinction is made between two ways in which terms can refer inside the scope of an epistemic modality. To illustrate the distinction, say that agent A receives the value $enc(c, c')$, where either of c, c' may be unknown

to A. Is it then true that "A knows that A received $enc(c, c')$"? Under the *de re* interpretation (cf. [14]), the answer is yes: The value ("bitstring") denoted by $enc(c, c')$ is known by A to be received. On the other hand, under the *de dicto* interpretation, the statement is about the term "$enc(c, c')$" itself. In this case, the statement might be false: Agent A need not know that the term used, "$enc(c, c')$", applies to the value received. In previous sections, we have assumed that all terms refer *de re*. Thus, $A \mathit{\ received\ } M \rightarrow \Box_A A \mathit{\ received\ } M$ has been considered intuitively valid for all terms M. This assumption that all terms refer *de re* is common to most combinations of epistemic logic and cryptography.

However, logical omniscience (1.11) contradicts resource-bounded knowledge only if complex terms are assumed to refer *de re*. For instance, (1.12) ascribes unlimited decryption power only if term $enc(M, K)$ refers *de re*. Thus, if we instead let complex terms refer *de dicto*, we regain logical omniscience as an acceptable rule. This is attractive, since logical omniscience (also known as the *rule of normality*) is fundamental to many results in modal logic.

Some mechanism to refer *de re*, however, is needed, since security goals may concern knowledge of partly undecryptable messages (cf. anonymity goals 1 and 2 in section 1.1, where the spy might be unable to decrypt M). In philosophical logic (cf. [14]), it is customary to let variables $x, y, z, \ldots$ refer *de re* while letting closed terms (terms built from constants and function symbols, but with no variables) refer *de dicto*. Following this custom, the *de re* statement:

$$A \mathit{\ received\ } x \rightarrow \Box_A A \mathit{\ received\ } x$$

is intuitively valid, while the corresponding *de dicto* schema:

$$A \mathit{\ received\ } M \rightarrow \Box_A A \mathit{\ received\ } M, \quad \text{all terms } M$$

is intuitively invalid.

1.11 First-Order Epistemic Logic

It can be argued that quantifiers are so natural and convenient for program specifications that they should be brought explicitly into specification languages based on epistemic logic (cf. [10]). In a security protocol setting, the combination of quantifiers and epistemic modalities allows nuanced descriptions about knowledge of cryptographic structure. Indeed, understanding what agents know of cryptographic structure is sometimes essential for understanding a security protocol. For instance, in the mix-based protocol in section 1.1, we need to determine if the spy can link an encryption x which the mix inputs to an output y, i.e., if the spy can know that the input x *contains* the output y. As another example (to be developed in more detail in section 9.3), consider a protocol for secure electronic payments involving three parties: A customer, a merchant, and a bank. To place an order, the customer sends a message x_M containing two sections:

- An order section containing a list x_O of the products to be purchased.
- A payment section containing payment details x_P (credit card number, etc.).

The message x_M is intended to be asymmetrically opaque: The merchant should be able to determine only the order instruction x_O, and the bank should be able to determine only the payment instruction x_P. Thus, we might wish to check if:

- The merchant knows that x_M *contains* order details x_O.
- The merchant does not know that x_M *contains* payment details x_P.
- The bank knows that *there exists* some order details y_O such that the merchant knows that x_M *contains* order details y_O.

and inversely for the banks knowledge.

Moreover, quantifiers allow an embedding of the propositional language where complex cryptographic terms, such as $enc(M, K)$, refer *de re* into the first-order language where only variables x refer *de re*. To illustrate the embedding, the propositional statement

$$\Box_A \Box_B A \, received \, enc(M, K)$$

where the term $enc(M, K)$ refers *de re* can be translated to:

$$\exists x. x = enc(M, K) \wedge \Box_A \Box_B A \, received \, x$$

where only x refers *de re*.

While completeness for first-order modal logic is an active research area in philosophical logic (cf. [14, 23, 34, 36]), completeness for first-order epistemic logic with respect to semantics that are grounded, i.e., semantics without abstract epistemic primitives, has not received much attention in the literature. In [10], completeness with respect to standard multi-agent indistinguishability (1.6) is shown for a first-order epistemic logic, but we are not aware of any other grounded completeness results.

1.12 The Cryptographic Omniscience Problem

If we let variables refer *de re* and closed terms refer *de dicto*, logical omniscience is intuitively valid. However, an aspect of the logical omniscience problem remains. For languages with variables, the basic Kripke semantics generalizes (1.3) in the straightforward way ([14]):

$$s, V \models \Box_A F \Leftrightarrow \forall s' : s \longrightarrow_A s' \Rightarrow s', V \models F \tag{1.15}$$

where V is an assignment of messages M to variables x. If the semantics is grounded, mathematical operations do not depend on the current run (computation state) s, and so term equalities depend only on the assignment:

$$s, V \models t = t' \implies s', V \models t = t'$$

for any open terms t and t' built from one-way operations and variables x. For instance,

$$s, V \models x = dec(y, z) \implies s', V \models x = dec(y, z)$$

Consequently, in basic Kripke semantics (1.15), agents know all cryptographic equalities, which makes them *cryptographically omniscient*:

$$t = t' \models \Box_A\, t = t' \tag{1.16}$$

For example,

$$x = decrypt(y, z) \models \Box_A x = decrypt(y, z)$$

Thus, knowledge of an equality does not reflect that the equality is feasible to compute. Instead, the epistemic modality is vacuous on cryptographic equations. In fact, all counterexamples to logical omniscience (for languages with *de re* reference of complex cryptographic terms) translate directly into counter examples to cryptographic omniscience (for languages with *de re* reference of variables and *de dicto* reference of complex terms).

1.13 Contributions

In this thesis, we study the combination of epistemic logic and formal cryptography. We address the problem of how to reflect feasible computability within a Kripke-style framework. The contributions are as follows.

1. A generalized Kripke semantics for first-order epistemic logic and cryptography, the latter modeled using private constants and arbitrary cryptographic operations, as in the Applied Pi-calculus [32]. First-order Kripke semantics is generalized by updating the assignment (of data to logical variables) as we follow the epistemic accessibility relation from a system state to an indistinguishable system state. As a result, cryptographic omniscience is avoided. The epistemic accessibility relation and the update to assignments are determined by static equivalence [32], as reformulated in a manner reminiscent of framed bisimulation [3].

2. An axiomatization of first-order epistemic logic which is sound and complete relative to an underlying theory of cryptographic terms, and to an omega-rule for quantifiers. Besides standard axioms and rules from first-order epistemic logic, the axiomatization includes some novel axioms for the interaction between knowledge and cryptography. The axiomatization is illustrated by an embedding of BAN-like [16] proof rules.

3. Epistemic characterizations of static equivalence and Dolev-Yao message deduction [28].

4. A generalization of propositional Kripke semantics for symmetric cryptography. While the above first-order semantics updates the assignment, the propositional semantics updates the predicated term M inside the evaluated statement $F(M)$. As a result, logical omniscience is avoided. The epistemic accessibility relation used is in the tradition of AT-indistinguishability [7].

5. Decidability, soundness and completeness for propositional BAN-like [16] logics with respect to message passing systems. Completeness and decidability are generalized to logics induced from an arbitrary base of protocol specific assumptions.

6. A novel epistemic definition of message deduction. The definition lies between weaker and stronger versions of Dolev-Yao deduction, and coincides with weaker Dolev-Yao regarding all atomic messages. For composite messages, the definition withstands the well-known Duck-Duck-Goose counterexample [43] to Dolev-Yao deduction.

7. Protocol examples using mixes [17], a Crowds [70] style protocol, and electronic payments [62].

The completeness result (2) above is the main technical result in the thesis. Result (5) (excluding soundness) depends on a restriction to a finite message space, on a somewhat artificial definition of message passing system and on a quasi-semantic proof rule. Still, the completeness result (5) is the first attempt in the literature at completeness for BAN-like logics. In contrast to result (5), the completeness result (2) has no such ad hoc limitations.

The thesis is divided into two parts, which can be read independently. The first part includes results (4), (5) and (6) above, while the second part includes results (1), (2) and (3). The protocol examples (result (7)) are shared between the two parts.

1.14 Publications

The thesis is based on the results originally presented in the following publications (The numbers are from the bibliography at the end of the thesis):

[20] Mika Cohen and Mads Dam. Logical Omniscience in the Semantics of *BAN* Logic. In *Foundations of Computer Security Workshop (FCS)*, 2005, 121-132.

Chapters 2 - 5 are based on the above paper. In addition, these chapters include the following results and examples which are not to be found in the above paper: Lemma 4.1.6, example 4.1.2, lemma 4.1.8, example 4.1.6, example 4.2.3, proposition 4.2.5, proposition 4.2.7, lemma 4.2.8, proposition 5.1.3, corollary 5.1.4, proposition 5.1.6, proposition 5.2.1, proposition 5.2.2, proposition 5.2.3, proposition 5.3.6, lemma 5.4.1, corollary 5.4.2, lemma 5.4.4, lemma 5.4.5, theorem 5.4.6, corollary 5.4.7.

[**19**] Mika Cohen and Mads Dam. A Completeness Result for *BAN* Logic. In *Methods for Modalities Workshop (M4M)*, 2005, 202-219.

Chapter 6 is based on the above paper. The completeness result in chapter 6 adjusts the axiomatization and completeness construction from the above paper: Message passing systems no longer involve a special agent, the environment, which is not part of the logical language.

[**21**] Mika Cohen and Mads Dam. A Complete Axiomatization of Knowledge and Cryptography. To appear in *Logic in Computer Science (LICS)*, 2007.

Part II of this thesis is based on this paper. Part II includes the omitted proofs from this paper, some results for the mix based example (section 9.1) and some correspondence results (section 10.4).

The above papers are jointly authored with my supervisor, Mads Dam. Mads's role has mostly been that of an active supervisor: Mads has suggested results to pursue and participated in developing proof strategies. The details of definitions and proofs have been worked out by the present author.

Part I

Propositional Epistemic Logic and Symmetric Encryption

Chapter 2

Language and System

In this chapter, we define the language and the systems to be used in part I of the thesis.

2.1 Language

The set $\mathcal{T}$ of messages (terms) is defined by:

$$M, K ::= \ c \mid M \cdot K \mid \{M\}_K$$

where c ranges over a countable set $\mathcal{C}$ of message atoms ("constants"), $\cdot$ represents pairing and $\{-\}_-$ represents symmetric encryption. Assume a finite subset $\mathcal{A} \subseteq \mathcal{C}$ of agent names $A, B, C, \ldots$ The sub-message relation $\leq$ is the smallest reflexive and transitive relation on messages such that $M \leq \{M\}_K$, $K \leq \{M\}_K$, $M \leq M \cdot M'$ and $M' \leq M \cdot M'$.

Let p range over a countable set $\mathcal{P}$ of predicates with arities. We assume that $\mathcal{P}$ includes the special unary predicates *exists* and A *infers* for each $A \in \mathcal{A}$. Informally, A *infers* M if agent A deduces ("knows") the message M and can use it as decryption key, and M *exists* if M is a sub-message of some message some agent or other acted upon (for instance, sent, received or generated). The set $\mathcal{F}$ of statements F is generated by:

$$F ::= \ p(M_1, ..., M_n) \mid \Box_A F \mid F \wedge F \mid \neg F$$

where p has arity n. For practical reasons, we assume $n \geq 1$. Epistemic possibility $\Diamond_A$, read "Agent A considers it possible that", abbreviates $\neg\Box_A\neg$. Define disjunction ($\vee$), implication ($\rightarrow$), equivalence ($\leftrightarrow$) and truth ($\top$) in the usual way. Write $\bigwedge_{1 \leq i \leq n} F_i$ for the nested conjunction $F_1 \wedge \cdots \wedge F_n$, and let $\bigwedge_{1 \leq i \leq 0} F_i$ be $\top$.

2.2 System

We assume a standard form of multi-agent system [31, 66].[1] A system is a set of execution histories, intuitively the set of executions of some underlying program. Each execution history is a finite sequence of actions, such as actions for sending to and receiving from a common network. An agent observes some actions, but not others. For instance, an agent might observe its own sending and receiving actions, but not the sending and receiving of other agents. On the other hand, if the agent is a spy who eavesdrops on the network, the agent might observe also the communication actions of other agents.

The details are as follows. An execution history is a sequences h of the form:

$$h ::= i \mid h \cdot \pi(M)$$

where π ranges over a primitive, non-empty set Π of actions, $i : \mathcal{A} \longrightarrow 2^{\mathcal{T}}$ and $\cdot$ is sequence concatenation. The initialization i assigns a finite set $i(A)$ of messages to agent A, the messages A possesses when execution begins. The expression $\pi(M)$ represents the action π applied to message M. For instance, if π represents the action "Agent A outputs" then the expression $\pi(M)$ represents that "agent A outputs message M". A system is a triple $\mathcal{S} = \langle \Pi, H, | \rangle$ of an action vocabulary Π, a non-empty set H of execution histories over Π and an observation function $| : \mathcal{A} \longrightarrow 2^{\Pi}$. Informally, H is the set of executions of some underlying program. Since H need not be closed under prefixing, H may consist of only completed program executions. The value of A under $|$, written $\Pi|A$, is the set of actions observed directly by agent A. Observation functions lift naturally to execution histories. The local history of A in h, written $h|A$, is defined by:

$$\begin{aligned} i|A &= \mathtt{init}\, i(A) \\ (h \cdot \pi(M))|A &= (h|A) \cdot \pi(M),\ \pi \in \Pi|A \\ (h \cdot \pi(M))|A &= (h|A),\ \pi \notin \Pi|A \end{aligned}$$

where $\mathtt{init}\,\kappa$ represents a local initialization which generates the set κ of messages.

Example 2.2.1 (Message Passing System) *In a message passing system, the agents take turns to send and receive messages on a common network. We say that system $\mathcal{S}$ is a message passing system if* $\Pi = \{A\,\mathtt{sends}, A\,\mathtt{receives} \mid A \in \mathcal{A}\}$, *and* $\Pi|A = \{A\,\mathtt{sends}, A\,\mathtt{receives}\}$. *In message passing systems, thus, the observation*

[1] The definitions and results in chapters 3, 4 and 5 are easily transferred to other variants of multi-agent systems (cf. [31]).

function lifts to histories as follows:

$$\begin{aligned} i|A &= \mathtt{init}\, i(A) \\ (h \cdot A\ \mathtt{sends}\, M)|A &= (h|A) \cdot A\, \mathtt{sends}\, M \\ (h \cdot B\, \mathtt{sends}\, M)|A &= (h|A),\ B \neq A \\ (h \cdot A\, \mathtt{receives}\, M)|A &= (h|A) \cdot A\, \mathtt{receives}\, M \\ (h \cdot B\, \mathtt{receives}\, M)|A &= (h|A),\ B \neq A \end{aligned}$$

Example 2.2.2 (Message Passing System with Spying) *Assume a function* $realm : \mathcal{A} \longrightarrow 2^{\mathcal{A}}$ *assigning a set* $realm(A)$ *of agents that* A *observes ("spies on"). Assume that* $A \in realm(A)$ *for each* $A \in \mathcal{A}$*. System* $\mathcal{S}$ *is a message passing system with spying based on realm, if* $\Pi = \{A\, \mathtt{sends},\ A\, \mathtt{receives} \mid A \in \mathcal{A}\}$*, and*

$$\Pi|A \quad = \quad \{B\, \mathtt{sends},\ B\, \mathtt{receives} \mid B \in realm(A)\}$$

Thus, for message passing systems with spying, we have:

$$\begin{aligned} i|A &= \mathtt{init}\, i(A) \\ (h \cdot B\, \mathtt{sends}\, M)|A &= (h|A) \cdot B\, \mathtt{sends}\, M,\ B \in realm(A) \\ (h \cdot B\, \mathtt{sends}\, M)|A &= (h|A),\ B \notin realm(A) \\ (h \cdot B\, \mathtt{receives}\, M)|A &= (h|A) \cdot B\, \mathtt{receives}\, M,\ B \in realm(A) \\ (h \cdot B\, \mathtt{receives}\, M)|A &= (h|A),\ B \notin realm(A) \end{aligned}$$

If $realm(A) = \{A\}$ *then* $\mathcal{S}$ *is simply a message passing system. Write* $A \longrightarrow B : M$ *to abbreviate the sequence:* $(A\, \mathtt{sends}\, M) \cdot (B\, \mathtt{receives}\, M)$*.*

We introduce the auxiliary notion of action trace. An action trace is a finite, possibly empty sequence θ of initializations, local initializations and actions:

$$\theta ::= \epsilon \mid \theta \cdot i \mid \theta \cdot \mathtt{init}\, \kappa \mid \theta \cdot \pi(M)$$

where ϵ is the empty sequence and $\kappa \subseteq \mathcal{T}$. Thus, histories h and local histories $h|A$ are action traces. Write $messages(\theta)$ for the set of the messages initially possessed or acted upon in θ:

$$\begin{aligned} messages(\epsilon) &= \emptyset \\ messages(\theta \cdot i) &= messages(\theta) \cup \bigcup ran(i) \\ messages(\theta \cdot \mathtt{init}\, \kappa) &= messages(\theta) \cup \kappa \\ messages(\theta \cdot \pi(M)) &= messages(h) \cup \{M\} \end{aligned}$$

where $ran(i)$ is the range of i. Write $actions(\theta)$ for the set of actions occurring in action trace θ:

$$\begin{aligned} actions(\epsilon) &= \emptyset \\ actions(\theta \cdot i) &= actions(\theta) \cup \{i\} \\ actions(\theta \cdot \mathtt{init}\,\kappa) &= actions(\theta) \cup \{\mathtt{init}\,\kappa\} \\ actions(\theta \cdot \pi(M)) &= actions(\theta) \cup \{\pi(M)\} \end{aligned}$$

Interpretation of Predicates A predicate interpretation I on a system $\mathcal{S} = \langle \Pi, H, | \rangle$ assigns, to each predicate p and history $h \in H$, a relation $I(p, h)$ in $\mathcal{T}$ (matching the arity of p). An interpreted system based on $\mathcal{S}$ is a pair $\mathcal{I} = \langle \mathcal{S}, I \rangle$, where I is an interpretation on $\mathcal{S}$. For predicate *exists*, we assume the following fixed interpretation:

$$I(exists, h) = \{M \mid \exists M' \geq M.\, M' \in messages(h)\}$$

The interpretation of special predicate A *infers* is left open until chapter 5, where various choices are considered.

Example 2.2.3 *Assume an interpreted system $\mathcal{I}$ based on a message passing system (example 2.2.1) or a message passing system with spying (example 2.2.2). If $\mathcal{P}$ includes any of the unary predicates A received, A sent, A rec or A sen, we assume the following fixed interpretation:*

$$\begin{aligned} I(A\ sent, h) &= \{M \mid (A\,\mathtt{sends}\,M) \in actions(h)\} \\ I(A\ received, h) &= \{M \mid (A\,\mathtt{receives}\,M) \in actions(h)\} \\ I(A\ rec, h) &= \{M \mid \exists M' \geq M.\, A\,\mathtt{receives}\,M' \in actions(h)\} \\ I(A\ sen, h) &= \{M \mid \exists M' \geq M.\, A\,\mathtt{sends}\,M' \in actions(h)\} \end{aligned}$$

Thus, $A\,rec\,M$ holds if M is part of something A received, and $A\,sen\,M$ holds if M is part of something A sent.

2.3 Anonymity Example

Prima facie, anonymity is an epistemic notion: An action is anonymous if an observer cannot know who performed the action. Indeed, several recent papers analyse anonymity in terms of epistemic logic (cf. [42, 49, 76, 82]).

Specification Template

In [42], a simple template for anonymity specifications is proposed. Adapted to our language, the template looks as follows. Assume an anonymity set $X \subseteq \mathcal{A}$ of agents, and assume an n-ary (primitive or defined) predicate p_A for each agent

$A \in X$. Informally, $p_A(M_1, ..., M_n)$ expresses that agent A has performed action p on message arguments $M_1, ..., M_n$. We say that p is anonymous with respect to an observer $spy \in \mathcal{A}$ and anonymity set X if:

$$p_A(M_1, ..., M_n) \rightarrow \neg\Box_{spy} p_A(M_1, ..., M_n) \tag{2.1}$$

$$p_A(M_1, ..., M_n) \rightarrow \Diamond_{spy} p_B(M_1, ..., M_n) \tag{2.2}$$

for all $A, B \in X$ and all messages M_i. For instance, to express anonymity in a voting protocol, let p_A be the unary predicate $A\,voted$, expressing that agent A voted for the argument:

$$A\,voted\,M \rightarrow \neg\Box_{spy} A\,voted\,M$$

$$A\,voted\,M \rightarrow \Diamond_{spy} B\,voted\,M$$

for all A, B in the anonymity set X of voters.

Crowds-Style Protocol

We illustrate specification template (2.1) - (2.2) in a protocol for anonymized message delivery in the style of Crowds [70]. The protocol 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 $\{to\,B : M\}_K$ to some random crowd member C_1, where $to\,B : M$ abbreviates, say, $B \cdot M$. Agent C_1, in turn, sends the received ciphertext to B or to a random forwarder $C_2 \in Crowd$, and so on, until the message reaches its intended destination B:

$$\begin{aligned} A &\longrightarrow C_1 : \{to\,B : M\}_K \\ C_1 &\longrightarrow C_2 : \{to\,B : M\}_K \\ &\vdots \\ C_n &\longrightarrow B : \{to\,B : M\}_K \end{aligned}$$

In addition to crowd members, there are some spies, each spy eavesdropping on part of the network. Assume that $Crowd \subseteq \mathcal{A}$ and assume a set $Spies \subseteq \mathcal{A}$, disjoint from $Crowd$. Assume a function realm : $\mathcal{A} \longrightarrow 2^{\mathcal{A}}$ such that:

$$\begin{aligned} realm(A) &= \{A\},\ A \in Crowd \\ spy &\in realm(spy),\ spy \in Spies \end{aligned}$$

Informally, $realm(A)$ is the set of agents that A observes; Crowd members observe only their own actions, while a spy might observe the actions of some crowd members. Let $X_{spy} = \{A \in Crowds \mid A \notin realm(spy)\}$ be the set of all crowd members outside the observation domain of $spy \in Spies$.

Sender anonymity means that a spy cannot tell the originator of a given message:

$$A \textit{ originated } M \rightarrow \neg\Box_{spy} A \textit{ originated } M,\ A \in X_{spy} \tag{2.3}$$

$$A \textit{ originated } M \rightarrow \Diamond_{spy} B \textit{ originated } M,\ A, B \in X_{spy} \tag{2.4}$$

Receiver anonymity, on the other hand, means that the spy cannot tell the intended destination of a given message:

$$M \textit{ is for } A \rightarrow \neg\Box_{spy} M \textit{ is for } A,\ A \in X_{spy} \tag{2.5}$$

$$M \textit{ is for } A \rightarrow \Diamond_{spy} M \textit{ is for } B,\ A, B \in X_{spy} \tag{2.6}$$

where M *is for* A holds if the intended final destination of M is agent A. Note that (2.3) and (2.4) instantiate templates (2.1) and (2.2), with p_A set to the predicate A *originated* and X set to X_{spy}. Similarly, (2.5) and (2.6) instantiate templates (2.1) and (2.2), with p_A set to the predicate *is for* A, although, here, the predicate p_A does not express that A performed some specific action p.

Protocol Implementation

We implement the protocol in a message passing system with spying (example 2.2.2). Assume that *Crowd* contains at least three members. Assume also that for each $spy \in Spies$, there are at least two crowd members $A, B \in Crowd$ unobserved by spy, i.e., $A, B \notin realm(spy)$. Let $\mathcal{S} = \langle \Pi, H, | \rangle$ be the message passing system with spying based on *realm* and where H consists of all histories of the form:

$$i \cdot (A_1 \longrightarrow A_2 : \{A_n \cdot M\}_K) \cdots (A_{n-1} \longrightarrow A_n : \{A_n \cdot M\}_K)$$

for any initialization i, any natural number n, any agents $A_1, ..., A_n$ and any messages M and K such that

- $n > 1$
- $A_1, ..., A_n \in Crowd$
- $M, K \in \mathcal{C} - \mathcal{A}$
- $i(A_1) = \{K, M\}$, $i(A) = \{K\}$ for $A \in Crowd - \{A_1\}$, $i(spy) = \emptyset$ for $spy \in Spies$

In initialization i, each crowd member obtains the shared key K, and the protocol initiator, A_1, obtains, in addition, the message payload M. The ciphertext $\{A_n \cdot M\}_K$ travels from A_1 to A_2, from A_2 to A_3, and so on until it reaches its intended destination A_n.

Let $\mathcal{I} = \langle \mathcal{S}, I \rangle$ be an interpreted system, based on the above implementation $\mathcal{S}$, such that:

$$\begin{aligned} M \in I(A \textit{ originated}, h) &\Leftrightarrow \exists i. \exists \theta. h = i \cdot (A \,\texttt{sends}\, M) \cdot \theta \\ M \in I(\textit{is for } A, h) &\Leftrightarrow \exists \theta. h = \theta \cdot (A \,\texttt{receives}\, M) \end{aligned}$$

where θ ranges over action traces and i over initializations. Thus, agent A originated a message if the first action, after initialization, was A sending that message. A message is for agent A if the last action is agent A receiving that message. Clearly, these definitions are specific to system $\mathcal{S}$. If messages could be lost – say spies were active and sometimes blocked messages – then the predicate *is for A* would have to be interpreted in terms of message structure, and not in terms of where M eventually ends up: M *is for* A if someone sent M and M contains destination field A.[2] In the current system $\mathcal{S}$, however, messages are not lost. For the predicate A *originated*, a more generally applicable definition is possible, but our simple interpretation suffices for the specifications here.

[2]I.e., $M = \{to\, A : M'\}_K$ for some M' and K.

Chapter 3

Kripke Semantics and Cryptography

Epistemic logic has a standard semantical framework, Kripke semantics [48]. In this chapter, we review some existing combinations of Kripke semantics and formal cryptography.

3.1 The Logical Omniscience Problem

In Kripke semantics, the epistemic modality $\Box_A$ is interpreted through an epistemic possibility relation $\sim_A$ between states, in our case histories. The agent knows a fact F, if F holds at all epistemically possible histories.

Definition 3.1.1 (Kripke Semantics)

$$h \models_{\mathcal{I}} \Box_A F \quad \Leftrightarrow \quad \forall h' \in H : h \sim_A h' \Rightarrow h' \models_{\mathcal{I}} F$$

Informally, $h \sim_A h'$ means that at history h agent A, given all it knows, could just as well be at h'. In computer science applications of epistemic logic, $\sim_A$ is typically an equivalence relation, the intuition being that $h \sim_A h'$ if h and h' are indistinguishable to A. Truth conditions for Boolean operators and atomic statements are the usual:

$$\begin{aligned} h \models_{\mathcal{I}} p(M_1, ..., M_n) &\quad \Leftrightarrow \quad \langle M_1, ..., M_n \rangle \in I(p, h) \\ h \models_{\mathcal{I}} F \wedge F' &\quad \Leftrightarrow \quad h \models_{\mathcal{I}} \text{ and } h \models_{\mathcal{I}} F' \\ h \models_{\mathcal{I}} \neg F &\quad \Leftrightarrow \quad h \not\models_{\mathcal{I}} F \end{aligned}$$

Entailment and validity are also defined as usual. For a set Δ of statements, write $h \models_{\mathcal{I}} \Delta$ if $h \models_{\mathcal{I}} F$ for all $F \in \Delta$. A set Δ entails a statement F in interpreted system $\mathcal{I}$, written $\Delta \models_{\mathcal{I}} F$, if for all histories $h \in H$, if $h \models_{\mathcal{I}} \Delta$ then $h \models_{\mathcal{I}} F$. The set Δ entails F in system $\mathcal{S}$, written $\Delta \models_{\mathcal{S}} F$, if Δ entails F in all interpreted

systems based on $\mathcal{S}$. The set Δ entails F, in symbols $\Delta \models F$, if Δ entails F in all interpreted systems. A statement F is valid in $\mathcal{I}/\mathcal{S}$ if the empty set entails F in $\mathcal{I}/\mathcal{S}$. The statement F is valid if the empty set entails F.

In Kripke semantics, no matter what epistemic possibility relation $\sim_A$ is chosen, agents know all the logical consequences of what they know, they are logically omniscient. Writing $\Box_A\Delta$ for the set $\{\Box_A F \mid F \in \Delta\}$, we have:

Fact 3.1.2 (Logical Omniscience) $\Delta \models_{\mathcal{I}} F \Rightarrow \Box_A\Delta \models_{\mathcal{I}} \Box_A F$

Elsewhere in modal logic, i.e., outside epistemic logic, logical omniscience is known as the *rule of normality*, and we shall use the terms interchangeably.

Logical omniscience does not agree with our use of the epistemic modality. In particular, it goes against the intended meaning of anonymity templates (2.1) and (2.2) for cryptographic terms M. Consider the implementation $\mathcal{I}$ of the Crowds-style protocol in section 2.3. For any two crowd members $A \neq B$, we have:

$$\models_{\mathcal{I}} \neg\{to\, B : M\}_K \; is\; for\, A \tag{3.1}$$

By logical omniscience, we obtain:

$$\models_{\mathcal{I}} \Box_{spy}\neg\{to\, B : M\}_K \; is\; for\, A$$

for any spy. Consequently, receiver anonymity specification (2.6) fails in $\mathcal{I}$. But, intuitively, receiver anonymity should not fail merely due to (3.1). As another example, sender anonymity specification (2.3) is also problematic under logical omniscience. Assume a global eavesdropper *spy*, observing all network traffic. Intuitively, the global eavesdropper can trace a message from its origination to its final destination. Thus, sender anonymity (2.3) should fail for *spy*:

$$A\; originated\, \{to\, B : M\}_K \models_{\mathcal{I}} \Box_{spy} A\; originated\, \{to\, B : M\}_K \tag{3.2}$$

However, from the triviality:

$$A\; originated\, \{to\, B : M\}_K \models_{\mathcal{I}} exists\, M$$

logical omniscience yields:

$$\Box_{spy} A\; originated\, \{to\, B : M\}_K \models_{\mathcal{I}} \Box_{spy} exists\, M \tag{3.3}$$

Combining (3.2) and (3.3), we get:

$$A\; originated\, \{to\, B : M\}_K \models_{\mathcal{I}} \Box_{spy} exists\, M$$

stating that the message payload M is leaked to the global eavesdropper. Again, this is counterintuitive; We expect that M remains confidential.

More generally, logical omniscience is incompatible with the combination of two assumptions: On the one hand, the assumption of feasible computability, or, more

precisely, the assumption that the epistemic modality reflects that agents can only perform feasibly computable cryptographic calculations. On the other hand, the assumption that message terms refer *de re*, i.e., "by value". *De re* reference means, for instance, that the statement

$$\Box_{spy} A \, originated \, \{to \, B : M\}_K \tag{3.4}$$

expresses that the spy knows that A originated a given value ("bitstring"). The statement (3.4) leaves it open to what extent the spy can decrypt that value and determine its message content and encryption key. Therefore, if the spy has only limited decryption power and term $\{to\, B : M\}_K$ refers *de re*, then statement (3.4) should not entail

$$\Box_{spy} exists \, M$$

although it does so under logical omniscience.

Kripke semantics is quite frequently used for languages where the modality is intended to reflect feasible computability and cryptographic terms M are intended to refer *de re* (cf. [7, 45, 75, 76]), even if, as we have seen, it can lead to unreasonable conclusions about the knowledge of agents.

3.2 Classical Multi-Agent Knowledge

The Kripkean accessibility relation has a default definition in multi-agent systems [31, 66]: $h \sim_A h'$ if A's local observations are the same in h and h'. In our setting, this translates to the following.

Definition 3.2.1 (Classical Indistinguishability)

$$h \sim_A h' \Leftrightarrow h|A = h'|A$$

As it happens, the classical semantics is problematic even if we drop the assumption that terms refer *de re*: In the classical semantics, agents are *local state omniscient*, i.e., agents know every property of their own local state, including properties that require infeasible cryptographic resources to calculate. We say that F is about A in $\mathcal{I}$, if F only depends on the local history of A, i.e., if

$$h \models_{\mathcal{I}} F, \; h|A = h'|A \implies h' \models_{\mathcal{I}} F$$

Corollary 3.2.2 (Local State Omniscience) *The following is valid in $\mathcal{I}$, assuming that F is about A:*

$$F \rightarrow \Box_A F$$

Example 3.2.3 *Continuing example 2.2.3, the statements $A \, rec \, M$ and $A \, sen \, M$ are about agent A in any interpreted system $\mathcal{I}$ based on a message passing system. By corollary 3.2.2, $\mathcal{I}$ validates:*

$$A \, rec \, M \rightarrow \Box_A A \, rec \, M$$
$$A \, sen \, M \rightarrow \Box_A A \, sen \, M$$

which goes against the assumption of limited decryption power of agents, even if terms do not refer de re.

3.3 AT-Style Semantics

The so called AT semantics [7], named after Abadi and Tuttle, weakens the classical indistinguishability relation (definition 3.2.1) so as to avoid local state omniscience.[1] The intuition is that, due to the limited decryption power of agents, data $h|A$ and $h'|A$ can carry the same information, even if $h|A \neq h'|A$. Approximately, two local states have the same information content if they are identical except for content inside unopened encryptions. For instance, the local states $\mathtt{init}\,\emptyset \cdot \pi(\{A\}_K)$ and $\mathtt{init}\,\emptyset \cdot \pi(\{B\}_K)$ have the same information content. On the other hand, assuming that $A \neq B$, the local states $\mathtt{init}\,\{K\} \cdot \pi(\{A\}_K)$ and $\mathtt{init}\,\{K\} \cdot \pi(\{B\}_K)$ have different information content, since the decryption key K is known at each local state. AT semantics is defined using a Dolev-Yao definition of inferred ("known") messages (cf. sections 1.3 and 5.1), but here we leave the definition of $I(A\ \mathit{infers}, h)$ open. However, throughout section 3.3, we assume that the set of known keys depends just on the local history, i.e., $h|A = h'|A'$ implies that $I(A\ \mathit{infers}, h) = I(A'\ \mathit{infers}, h')$.

The details are as follows. Write $\mathit{struct}_\kappa(M)$ for the structure of message M discernable through a set $\kappa \subseteq \mathcal{T}$ of decryption keys:

$$\begin{array}{rcl} \mathit{struct}_\kappa(\{M\}_K) & = & \bot,\ K \notin \kappa \\ \mathit{struct}_\kappa(\{M\}_K) & = & \{\mathit{struct}_\kappa(M)\}_{\mathit{struct}_\kappa(K)},\ K \in \kappa \\ \mathit{struct}_\kappa(M \cdot M') & = & \mathit{struct}_\kappa(M) \cdot \mathit{struct}_\kappa(M') \\ \mathit{struct}_\kappa(c) & = & c,\ c \in \mathcal{C} \end{array}$$

where $\bot$ is a fixed dummy symbol. For instance, $\mathit{struct}_{\{c\}}(\{c' \cdot \{M\}_K\}_c) = \{c' \cdot \bot\}_c$ if $K \neq c$. Discernable structure lifts to local histories by pointwise application:

$$\begin{array}{rcl} \mathit{struct}_\kappa(\mathtt{init}\,\kappa) & = & \mathtt{init}\,\{\mathit{struct}_\kappa(M) \mid M \in \kappa\} \\ \mathit{struct}_\kappa(h|A \cdot \pi(M)) & = & \mathit{struct}_\kappa(h|A) \cdot \pi(\mathit{struct}_\kappa(M)) \end{array}$$

Definition 3.3.1 (Information Content) *The information content in local state $h|A$, written content$(h|A)$, is $\mathit{struct}_{I(A\ \mathit{infers}, h)}(h|A)$*

Definition 3.3.2 (AT Indistinguishability)

$$h \sim_A h' \Leftrightarrow \mathit{content}(h|A) = \mathit{content}(h'|A)$$

Arguably, the AT semantics respects an assumption of limited decryption power. In particular, local state omniscience fails if the interpretation of predicate $A\ \mathit{infers}$ is reasonable, as the following example illustrates.

[1] Avoiding local state omniscience was not explicitly stated as a goal in [7], but we speculate that this was a motivation for the semantics.

Example 3.3.3 *Assume an interpreted message passing system $\mathcal{I}$ containing two histories h and h' such that:*

$$\begin{aligned} h|A &= \mathtt{init}\,\emptyset \cdot A\,\mathtt{receives}\,\{c\}_K \\ h'|A &= \mathtt{init}\,\emptyset \cdot A\,\mathtt{receives}\,\{c'\}_{K'} \end{aligned}$$

Assuming that $K \notin I(A\,infers, h)$ and $K' \notin I(A\,infers, h')$, we obtain that

$$content(h|A) = \mathtt{init}\,\emptyset \cdot (A\,\mathtt{receives}\,\bot) = content(h'|A)$$

i.e., $h \sim_A h'$. Assuming that $c \not\leq \{c'\}_{K'}$, it follows that $h \not\models_{\mathcal{I}} \Box_A A\,rec\,c$, i.e., $A\,rec\,c \not\models_{\mathcal{I}} \Box_A A\,rec\,c$, even though statement $A\,rec\,c$ is about agent A in $\mathcal{I}$.

The AT semantics has some successors (cf. [77, 83]). Most notably, SVO [77] adjusts AT so that the identity of an unopened ciphertext is discernable.[2] In AT, where all unopened messages reduce to a single dummy $\bot$, agents are unable to track unopened ciphertexts.

Example 3.3.4 *Consider the following message passing system $\mathcal{I}$ with spies. There are four different agents A, B, C and spy. The latter is a global eavesdropper: $realm(spy) = \{A, B, C, spy\}$. The set H of execution histories contains all histories of the form:*

$$i \cdot A\,\mathtt{sends}\,M_A \cdot B\,\mathtt{sends}\,M_B \cdot C\,\mathtt{receives}\,M$$

for any initialization i and any messages M_A, M_B and M such that $M = M_A$ or $M = M_B$. Thus, agent A talks to agent C if $M = M_A$. Intuitively, since the spy observes all network traffic and can track messages as they travel from one agent to the next, the spy knows if A is talking to C:

$$A\,talks\,to\,C \rightarrow \Box_{spy} A\,talks\,to\,C \tag{3.5}$$

However, in AT semantics (3.5) might fail in $\mathcal{I}$. Pick $h, h' \in H$ such that:

$$h = i \cdot A\,\mathtt{sends}\,\{M_A\}_{K_A} \cdot B\,\mathtt{sends}\,\{M_B\}_{K_B} \cdot C\,\mathtt{receives}\,\{M_A\}_{K_A}$$

$$h' = i \cdot A\,\mathtt{sends}\,\{M_A\}_{K_A} \cdot B\,\mathtt{sends}\,\{M_B\}_{K_B} \cdot C\,\mathtt{receives}\,\{M_B\}_{K_B}$$

Assume that $K_A, K_B \notin I(spy\,infers, h)$ and $K_A, K_B \notin I(spy\,infers, h')$. We obtain:

$$content(h|spy) = \mathtt{init}\,\kappa \cdot A\,\mathtt{sends}\,\bot \cdot B\,\mathtt{sends}\,\bot \cdot C\,\mathtt{receives}\,\bot = content(h'|spy)$$

for some set $\kappa \subseteq \mathcal{T}$. I.e., in AT semantics, $h \sim_{spy} h'$. Assuming $\{M_A\}_{K_A} \neq \{M_B\}_{K_B}$, we have $h' \not\models_{\mathcal{I}} A\,talks\,to\,C$, i.e., $h \not\models_{\mathcal{I}} \Box_{spy} A\,talks\,to\,C$. Thus, (3.5) fails in $\mathcal{I}$.

[2] SVO also extends the crypto algebra to asymmetric encryption. Here, we restrict ourselves to symmetric encryption.

In SVO, each unopened ciphertext M is replaced by a distinct dummy $\perp_M$; Two histories h and h' are considered indistinguishable to agent A if, after unopened ciphertexts have been replaced by dummies, there is a substitution of dummies that transforms the local history $h|A$ into the local history $h'|A$. In effect, therefore, agents can compare undecrypted messages for identity.

Since AT, and its successors, follow Kripke semantics, they do not support *de re* reference of message terms (see section 3.1), although at least the original AT semantics was intended to do so.[3] We illustrate with a simple example.

Example 3.3.5 *Consider any interpreted system $\mathcal{I}$ based on a message passing system. Under AT semantics, the following implications need not be valid in $\mathcal{I}$:*

$$\begin{aligned} A\,\mathit{received}\,M &\rightarrow \Box_A A\,\mathit{received}\,M \\ A\,\mathit{sent}\,M &\rightarrow \Box_A A\,\mathit{sent}\,M \end{aligned}$$

although, intuitively, they should be valid in $\mathcal{I}$ if term M refers de re: If an agent received/sent a value ("bitstring"), the agent knows it received/sent that value. As a counterexample in AT, assume that H contains, at least, execution histories h and h' such that:

$$\begin{aligned} h|A &= \mathtt{init}\,\emptyset \cdot (A\,\mathtt{receives}\,\{M\}_K) \cdot (A\,\mathtt{sends}\,\{M\}_K) \\ h'|A &= \mathtt{init}\,\emptyset \cdot (A\,\mathtt{sends}\,\{M'\}_{K'}) \cdot (A\,\mathtt{sends}\,\{M'\}_{K'}) \end{aligned}$$

where $M \neq M'$ and $K \neq K'$. Assume that $K \notin I(A\,\mathit{infers}, h)$ and assume also that $K' \notin I(A\,\mathit{infers}, h')$. Then,

$$\mathit{content}(h|A) = \mathtt{init}\,\emptyset \cdot (A\,\mathtt{receives}\,\perp) \cdot (A\,\mathtt{sends}\,\perp) = \mathit{content}(h'|A)$$

I.e., $h \sim_A h'$. Since $h' \not\models_{\mathcal{I}} A\,\mathit{received}\,\{M\}_K$, it follows $h \not\models_{\mathcal{I}} \Box_A A\,\mathit{received}\,\{M\}_K$. But, $h \models_{\mathcal{I}} A\,\mathit{received}\,\{M\}_K$. Similarly, we obtain $h \not\models_{\mathcal{I}} \Box_A A\,\mathit{sent}\,\{M\}_K$ and $h \models_{\mathcal{I}} A\,\mathit{sent}\,\{M\}_K$.

[3] See section 6.1.

Chapter 4

Permutation-Based Semantics

In this chapter, we generalize AT-style semantics by updating the predicated data as we follow the indistinguishability relation.

4.1 Relativized AT-style Indistinguishability

In AT-style semantics, data (i.e., messages) is substituted for other data as we follow $\sim_A$ from a history h to an epistemically possible history h': If $h \sim_A h'$, and

$$h|A = \texttt{init}\,\{M_0\} \cdot \pi_1(M_1) \cdots \pi_n(M_n),$$

then, for some possibly different messages $M'_0, ..., M'_n$, we have:

$$h'|A = \texttt{init}\,\{M'_0\} \cdot \pi_1(M'_1) \cdots \pi_n(M'_n)$$

Intuitively, message M_i at history h corresponds to ("is a counterpart of") M'_i at h', in the sense that everything that agent A observes of M_i at h, agent A also observes of M'_i at h'. Agent A observes, in particular, feasibly computable properties and relationships. For instance, if $M_1 = \{M_2\}_{M_3}$ then agent A can compute this relationship (since A is given the decryption key M_3), and so $M'_1 = \{M'_2\}_{M'_3}$.

Also intuitively, message correspondences extend to messages besides those the agent acted upon, in other words, besides those in *messages*$(h|A)$ and *messages*$(h'|A)$. For instance, if $h \sim_A h'$, and

$$h|A = \texttt{init}\,\{K\} \cdot \pi_1(\{M\}_K) \cdots$$

then for some K', M', etc.,

$$h'|A = \texttt{init}\,\{K'\} \cdot \pi_1(\{M'\}_{K'}) \cdots$$

and M corresponds to M', even if $M \notin$ *messages*$(h|A)$ and $M' \notin$ *messages*$(h'|A)$.

To make $\sim_A$ keep track of message correspondences, we relativize $\sim_A$ to a permutation ρ on $\mathcal{T}$. Informally, if $h \sim_A^\rho h'$ then for agent A, any message M at h corresponds to $\rho(M)$ at h'. For $h \sim_A^\rho h'$ to hold, we require that ρ respects the actions of A in h, i.e., we require that

$$\rho(h|A) = h'|A$$

where ρ is extended to local histories by pointwise application:

$$\begin{aligned} \rho(\mathtt{init}\,\kappa) &= \mathtt{init}\,\{\rho(M) \mid M \in \kappa\} \\ \rho(h|A \cdot \pi(M)) &= \rho(h|A) \cdot \pi(\rho(M))) \end{aligned}$$

Moreover, for $h \sim_A^\rho h'$ to hold, we require that ρ is consistent with the keys available to agent A at h, i.e., $I(A\,\mathit{infers}, h)$. Informally, ρ is consistent with a set of keys if ρ respects all the message structure accessible through the keys. Formally, permutation consistency is defined as follows.

Definition 4.1.1 (Consistent Permutation) *Permutation ρ is consistent with $\kappa \subseteq \mathcal{T}$, in symbols $\rho \lhd \kappa$, if and only if,*

1. $K \in \kappa \Rightarrow \rho(\{M\}_K) = \{\rho(M)\}_{\rho(K)}$
2. $\rho(M \cdot M') = \rho(M) \cdot \rho(M')$
3. $\rho(c) = c$, *for* $c \in \mathcal{C}$

For $M \not\geq M'$ and $M' \not\geq M$, we write $[M - M']$ for the substitution on messages that exchanges M and M': $[M - M'](M'')$ is the result of exchanging M and M' in M''.

Lemma 4.1.2 $K, K' \notin \kappa \implies [\{M\}_K - \{M'\}_{K'}] \lhd \kappa$

Proof Let $\rho = [\{M\}_K - \{M'\}_{K'}]$. (1) Trivially, ρ is a permutation. (2) If $N' \in \kappa$ then $\{N\}_{N'}$ is neither $\{M\}_K$ nor $\{M'\}_{K'}$, and so, $\rho(\{N\}_{N'}) = \{\rho(N)\}_{\rho(N')}$. (3) $\rho(N \cdot N') = \rho(N) \cdot \rho(N')$, since $N \cdot N'$ is neither $\{M\}_K$ nor $\{M'\}_{K'}$. (4) $\rho(c) = c$, since atomic c is neither $\{M\}_K$ nor $\{M'\}_{K'}$. □

We lift permutations to sets $\kappa \subseteq \mathcal{T}$ in the expected way ($\rho(\kappa) = \{\rho(M) \mid M \in \kappa\}$):

Lemma 4.1.3 *The following hold:*

- $\rho \lhd \kappa,\ \kappa \supseteq \kappa' \implies \rho \lhd \kappa'$ *(Monotonicity)*
- $id \lhd \kappa$ *(Reflexivity)*
- $\rho \lhd \kappa,\ \rho' \lhd \rho(\kappa) \implies (\rho' \circ \rho) \lhd \kappa$ *(Transitivity)*
- $\rho \lhd \kappa \implies \rho^{-1} \lhd \rho(\kappa)$ *(Symmetry)*

Proof Monotonicity and reflexivity: Immediate. Transitivity: Assume that $\rho \lhd \kappa$ and $\rho' \lhd \rho(\kappa)$. We show that $\rho' \circ \rho$ respects encryption with keys in κ (i.e., condition 1 in definition 4.1.1), showing that $r' \circ r$ respects clear text (i.e., conditions 2 and 3 in definition 4.1.1) is trivial. Assume that $K \in \kappa$. By the assumptions, $\rho(\{M\}_K) = \{\rho(M)\}_{\rho(K)}$ and $\rho'(\{\rho(M)\}_{\rho(K)}) = \{\rho'(r(M))\}_{\rho'(\rho(K))}$. Thus, $(\rho' \circ \rho)(\{M\}_K) = \rho'(\rho(\{M\}_K)) = \rho'(\{\rho(M)\}_{\rho(K)}) = \{\rho'(r(M))\}_{\rho'(\rho(K))} = \{(\rho' \circ \rho)(M)\}_{(\rho' \circ \rho)(K)}$. Symmetry: Assume that $\rho \lhd \kappa$. We show that ρ^{-1} respects encryption with keys in $\rho(\kappa)$ (i.e., condition 1 in definition 4.1.1), showing that ρ^{-1} respects clear text (i.e., conditions 2 and 3 in definition 4.1.1) is analogous. Assume that $K \in \rho(\kappa)$, i.e., $\rho^{-1}(K) \in \kappa$. By the assumption, $\rho(\{\rho^{-1}(M)\}_{\rho^{-1}(K)}) = \{\rho \circ \rho^{-1}(M)\}_{\rho \circ \rho^{-1}(K)} = \{M\}_K$. Thus, $\rho^{-1}(\{M\}_K) = \rho^{-1} \circ \rho(\{r^{-1}(M)\}_{\rho^{-1}(K)}) = \{\rho^{-1}(M)\}_{\rho^{-1}(K)}$. □

Conjoining the two requirements on $\sim_A^\rho$, we stipulate that $h \sim_A^\rho h'$ if ρ carries $h|A$ to $h'|A$ and ρ is consistent with the keys available to A at h.

Definition 4.1.4 (Relativized Indistinguishability) *$h \sim_A^\rho h'$ in $\mathcal{I}$, if and only if,*

- $\rho(h|A) = h'|A$
- $\rho \lhd I(A\ infers, h)$

Lemma 4.1.5 *$h \sim_A^{id} h$ (Reflexivity)*

Proof From reflexivity of $\lhd$ (lemma 4.1.3). □

Example 4.1.6 *Consider the implementation $\mathcal{I}$ of the Crowds-style protocol in section 2.3. Pick two execution histories $h, h' \in \mathcal{H}$ such that:*

$$\begin{aligned} h &= i \cdot (A \longrightarrow B : \{B \cdot M\}_K) \\ h' &= i' \cdot (A \longrightarrow C : \{C \cdot M'\}_{K'}) \end{aligned}$$

for three distinct crowd members A, B and C and some initializations i and i'. In h, agent A sends a message directly to B, while in h', agent A sends a message directly to C. Assume that $spy_A \in Spies$ eavesdrops on A but not on B or C, i.e., $A \in realm(spy_A)$ but $B, C \notin realm(spy_A)$. Assume that the interpretation I is such that:

$$K, K' \notin I(spy_A\ infers, h) \tag{4.1}$$

We proceed to show that:

$$h \sim_{spy_A}^\rho h' \tag{4.2}$$

where ρ is the permutation exchanging $\{B \cdot M\}_K$ and $\{C \cdot M'\}_{K'}$, in other words, $\rho = [\{B \cdot M\}_K - \{C \cdot M'\}_{K'}]$. First,

$$\begin{aligned} h|spy_A &= \texttt{init}\,\emptyset \cdot A\,\texttt{sends}\,\{B \cdot M\}_K \\ h'|spy_A &= \texttt{init}\,\emptyset \cdot A\,\texttt{sends}\,\{C \cdot M'\}_{K'} \end{aligned}$$

Thus,

$$\rho(h|spy_A) = h'|spy_A \tag{4.3}$$

From (4.1) and lemma 4.1.2, we get:

$$\rho \lhd I(spy_A \; infers, h) \tag{4.4}$$

since, by construction of $\mathcal{I}$, we have $M, M' \in \mathcal{C}$, i.e. $\{B \cdot M\}_K \not\geq \{C \cdot M'\}_{K'}$ and $\{C \cdot M'\}_{K'} \not\geq \{B \cdot M\}_K$. But, (4.2) follows from (4.3) and (4.4).

Under certain interpretations of the predicate *A infers*, reflexivity, transitivity and symmetry of $\lhd$ transfer to the relativized indistinguishability relation.

Definition 4.1.7 (Introspective Interpreted System) *Interpreted system $\mathcal{I}$ is introspective if, and only if,*

$$h \sim_A^{\rho} h' \implies \rho(I(A \; infers, h)) = I(A \; infers, h')$$

Lemma 4.1.8 *Assume that $\mathcal{I}$ is introspective.*

- $h \sim_A^{\rho} h', h' \sim_A^{\rho'} h'' \implies h \sim_A^{\rho' \circ \rho} h''$ *(Transitivity)*
- $h \sim_A^{\rho} h' \implies h' \sim_A^{\rho^{-1}} h$ *(Symmetry)*

Proof Symmetry: Assume that $h \sim_A^{\rho} h'$, i.e. $\rho(h|A) = h'|A$ and $\rho \lhd I(A \; infers, h)$. By symmetry of $\lhd$ (lemma 4.1.3), $\rho^{-1} \lhd \rho(I(A \; infers, h))$. Since $\mathcal{I}$ is introspective, $\rho^{-1} \lhd I(A \; infers, h')$. But, $\rho^{-1}(h'|A) = h|A$. Thus, $h' \sim_A^{\rho^{-1}} h$. Transitivity follows similarly from transitivity of $\lhd$ (lemma 4.1.3). □

The relativized indistinguishability relation implicitly defines an AT-like indistinguishability relation:

$$h \sim_A h' \Leftrightarrow \exists \rho : h \sim_A^{\rho} h' \tag{4.5}$$

With the existential quantification over permutations ρ, we loose the information about how cipher texts at h may correspond for A to cipher texts at h'.

The consistency relation $\lhd$ is related to the states of knowledge and belief of [11, 79]. The definition 4.1.1 of $\lhd$ is not intended to be canonical: There are alternative, equally reasonable, definitions. Most obviously, requirement (3), which reflects the assumption that atoms are "plain text", could be restricted to atoms in $\mathcal{A}$. As another example, it might, perhaps, be reasonable to restrict requirement (2) to messages in the given set κ:

$$\begin{aligned} M \in \kappa, \; M' \in \kappa \quad &\Rightarrow \quad \rho(M \cdot M') = \rho(M) \cdot \rho(M') \\ M \cdot M' \in \kappa \quad &\Rightarrow \quad \rho(M \cdot M') = \rho(M) \cdot \rho(M') \end{aligned}$$

However, with this restriction, soundness for classical BAN (chapter 6) would fail. (Specifically, BAN rules *R7* and *R8* would be unsound.) As regards the requirement that ρ must be a permutation, we note that symmetry of $\lhd$ in lemma 4.1.3, and indirectly symmetry of $\sim_A^{\rho}$ in lemma 4.1.8, depend on this requirement.

4.2 Permutation-Based Truth Condition

In AT-style semantics, data inside $h|A$ is transformed into other data as we follow $\sim_A$ from a history h to an indistinguishable history h'. However, in the Kripkean truth condition 3.1.1, predicated data, i.e., data inside the evaluated statement F, is left unchanged by the move from h to h'. Thus, the history and the statement are not "syncronized". Here, by contrast, we depart from AT-style semantics, and Kripke semantics in general, by updating the evaluated statement F to the corresponding statement $\rho(F)$ for each transition $h \sim_A^\rho h'$.

First, permutations are lifted to statements in the obvious way as follows:

$$\begin{aligned} \rho(p(\overline{M})) &= p(\rho(\overline{M})) \\ \rho(F \wedge F') &= \rho(F) \wedge \rho(F') \\ \rho(\neg F) &= \neg\rho(F) \\ \rho(\Box_A F) &= \Box_A \rho(F) \end{aligned}$$

Intuitively, if $h \sim_A^\rho h'$ then, for agent A, F at h corresponds to $\rho(F)$ at h'.

Example 4.2.1 *Continuing example 4.1.6, permutation ρ maps statement*

$$A\ sent\{B \cdot M\}_K \tag{4.6}$$

to statement

$$A\ sent\{C \cdot M'\}_{K'} \tag{4.7}$$

Thus, for spy_A, statement (4.6) at h corresponds to statement (4.7) at h'.

We stipulate that an agent knows a statement if *corresponding* statements hold at indistinguishable histories.

Definition 4.2.2 (Truth Condition for Knowledge)

$$h \models_{\mathcal{I}} \Box_A F \quad \Leftrightarrow \quad \forall \rho : \forall h' \in H : h \sim_A^\rho h' \Rightarrow h' \models_{\mathcal{I}} \rho(F) \tag{4.8}$$

Thus, we check a corresponding statement $\rho(F)$ at h', and not the original statement F, as in Kripke semantics (definition 3.1.1). Remaining truth conditions, as well as the notion of validity, are preserved from section 3.1.

Example 4.2.3 *Consider the history h in example 4.1.6. For all that spy_A knows, the value $\{B \cdot M\}_K$ goes to agent C:*

$$h \models_{\mathcal{I}} \Diamond_{spy_A} C\ received\{B \cdot M\}_K$$

This follows from (4.2) in example 4.1.6 and from $h' \models_{\mathcal{I}} C\ received \rho(\{B \cdot M\}_K)$.

Proposition 4.2.4 (Modal Axioms K and T) *The following are valid:*

1. $\Box_A(F \rightarrow F') \rightarrow \Box_A F \rightarrow \Box_A F'$

2. $\Box_A F \rightarrow F$

Proof (1): Straightforward. (2): Follows directly from lemma 4.1.5. □

Proposition 4.2.5 (Modal Axioms 4 and 5) *The following are valid in introspective interpreted systems:*

1. $\Box_A F \rightarrow \Box_A \Box_A F$
2. $\neg \Box_A F \rightarrow \Box_A \neg \Box_A F$

Proof From lemma 4.1.8. □

Proposition 4.2.6 (Receive and Send Introspection) *The following are valid in any interpreted system based on a message passing system:*

1. A *received* $M \rightarrow \Box_A A$ *received* M
2. A *sent* $M \rightarrow \Box_A A$ *sent* M

Proof Receive introspection: Assume that $h \models_{\mathcal{I}} A$ *received* M and $h \sim_A^{\rho} h'$ in $\mathcal{I}$. From the first assumption, $A \,\texttt{receives}\, M \in \mathit{Actions}(h|A)$, so by the second assumption, $A \,\texttt{receives}\, \rho(M) \in \mathit{Actions}(h'|A)$, i.e., $h' \models_{\mathcal{I}} A$ *received* $\rho(M)$. Since h' and ρ are arbitrary, $h \models_{\mathcal{I}} \Box_A A$ *received* M. Send introspection: Analogous. □

Proposition 4.2.6 can be generalized to arbitrary systems. If an operation $\pi \in \Pi$ is observable to an agent, the agent knows when π is applied to a message:

Proposition 4.2.7 (Action Introspection) *Assume that* $\Pi \subseteq \mathcal{P}$. *Assume that an interpreted system* $\mathcal{I}$ *such that:* $I(\pi, h) = \{M \mid \pi(M) \in \mathit{actions}(h)\}$. *The following is valid in* $\mathcal{I}$*:*

$$\pi(M) \rightarrow \Box_A \pi(M),\ \pi \in \Pi|A$$

Proof Assume that $\pi \in \Pi|A$ and $h \models_{\mathcal{I}} \pi(M)$ and $h \sim_A^{\rho} h'$ in $\mathcal{I}$. From the first and second assumption, $\pi(M) \in \mathit{Actions}(h|A)$, so by the third assumption, $\pi(\rho(M)) \in \mathit{Actions}(h'|A)$, i.e., $h' \models_{\mathcal{I}} \pi(\rho(M))$. Since h' and ρ are arbitrary, $h \models_{\mathcal{I}} \Box_A \pi(M)$. □

We recall the implementation of the Crowds-style protocol in section 2.3. It depends on the interpretation of the predicate *spy infers*, of course, whether or not the model satisfies its specifications.

Lemma 4.2.8 (Crowds-Style Protocol) *Let* $\mathcal{I}$ *be the protocol implementation in section 2.3. Specifications (2.3), (2.4), (2.5) and (2.6) are valid in* $\mathcal{I}$*, assuming:*

$$\mathcal{C} \cap I(\mathit{spy\ infers}, h) = \emptyset$$

for all protocol executions $h \in H$.

Proof For specification (2.6): Assume that $spy \in Spies$. Assume that $A, B \notin realm(spy)$. Assume also that $h \models_{\mathcal{I}} M'$ *is for* A. By construction of H, we have i, $A_1, \ldots, A_n, M$ and K such that h is:

$$i \cdot (A_1 \longrightarrow A_2 : \{A_n \cdot M\}_K) \cdots (A_{n-1} \longrightarrow A_n : \{A_n \cdot M\}_K)$$

where $A_n = A$ and $M' = \{A_n \cdot M\}_K$. By construction of H, there is $h' \in H$ such that h' is:

$$i \cdot (A_1 \longrightarrow A_2 : \{B \cdot M\}_K) \cdots (A_{n-1} \longrightarrow A_n : \{B \cdot M\}_K) \cdot (A_n \longrightarrow B : \{B \cdot M\}_K)$$

Let $\rho = [\{A \cdot M\}_K - \{B \cdot M\}_K]$ be the substitution that exchanges $\{A \cdot M\}_K$ and $\{B \cdot M\}_K$. Since $i(spy) = \emptyset$ and since $A = A_n, B \notin realm(spy)$, we have:

$$\rho(h|spy) = h'|spy \tag{4.9}$$

By construction of H, $K \in \mathcal{C}$. Thus, by assumption, $K \notin I(spy\, infers, h)$. By lemma 4.1.2:

$$\rho \lhd I(spy\, infers, h) \tag{4.10}$$

From (4.9) and (4.10):

$$h \sim^{\rho}_{spy} h' \text{ in } \mathcal{I} \tag{4.11}$$

Since $\rho(M') = \{B \cdot M\}_K$:

$$h' \models_{\mathcal{I}} \rho(M') \textit{ is for } B \tag{4.12}$$

From (4.11) and (4.12), $h \models_{\mathcal{I}} \Diamond_{spy} M'$ *is for* B. This completes the proof of specification (2.6).

For specification (2.5): Assume that $spy \in Spies$. Assume that $A \notin realm(spy)$. By construction of H, there exists $B \notin realm(spy)$ such that $A \neq B$. Assume that $h \models_{\mathcal{I}} M'$ *is for* A. By the same reasoning as for (2.6), we obtain $h \sim^{\rho}_{spy} h'$ in $\mathcal{I}$ and $h' \models_{\mathcal{I}} \rho(M')$ *is for* B. Since $A \neq B$, $h' \not\models_{\mathcal{I}} \rho(M')$ *is for* A. Consequently, $h \not\models_{\mathcal{I}} \Box_{spy} M'$ *is for* A.

For specification (2.4): Assume that $spy \in Spies$. Assume that $A, B \notin realm(spy)$. Assume that $h \models_{\mathcal{I}} A$ *originated* M'. By construction of H, there are i, $A_1, \ldots, A_n, M$ and K such that h is:

$$i \cdot (A_1 \longrightarrow A_2 : \{A_n \cdot M\}_K) \cdots (A_{n-1} \longrightarrow A_n : \{A_n \cdot M\}_K)$$

where $A_1 = A$ and $M' = \{A_n \cdot M\}_K$. By construction of H, there is $h' \in H$ and initialization i' such that h' is:

$$i' \cdot (B \longrightarrow A_1 : \{A_n \cdot M\}_K) \cdot (A_1 \longrightarrow A_2 : \{A_n \cdot M\}_K) \cdots \\ \cdots (A_{n-1} \longrightarrow A_n : \{A_n \cdot M\}_K)$$

Since $A, B \notin realm(spy)$ and $i(spy) = i'(spy) = \emptyset$, we have $h|spy = h'|spy$. So, by lemma 4.1.3, $h \sim^{id}_{spy} h'$ in $\mathcal{I}$. But, $h' \models_{\mathcal{I}} B$ *originated* $id(M')$. Consequently, $h \models_{\mathcal{I}} \Diamond_{spy} B$ *originated* M'. This completes the proof of specification (2.4).

Specification (2.3) is obtained from the proof of (2.4) in the same way that (2.5) is obtained above from the proof of (2.6). □

4.3 Weak Normality

As the following example illustrates, the permutation-based semantics avoids logical omniscience (the rule of normality).

Example 4.3.1 *Continuing example 4.1.6, assume a binary predicate, contains, interpreted in $\mathcal{I}$ as follows: $I(contains, h) = \{\langle M, M'\rangle \mid M \geq M'\}$. Trivially, $\models_{\mathcal{I}} \{B \cdot M\}_K$ contains B. However, $\not\models_{\mathcal{I}} \Box_{spy_A}\{B \cdot M\}_K$ contains B, since by (4.2), we have $h \sim^{\rho}_{spy_A} h'$ and $h' \not\models_{\mathcal{I}} \rho(\{B \cdot M\}_K)$ contains $\rho(B)$. (We assume that $M' \not\geq B$ and $K' \not\geq B$.)*

Thus, knowledge is not closed under all entailments. Still, knowledge is closed under entailments that depend only on accessible structure. Let $A \textit{ infers } \kappa = \{A \textit{ infers } K \mid K \in \kappa\}$.

Lemma 4.3.2 (Permutation Normality)

$$\rho(F) \models_{\mathcal{I}} \rho(F'), \forall \rho \lhd \kappa \implies A \textit{ infers } \kappa, \Box_A F \models_{\mathcal{I}} \Box_A F'$$

Proof From monotonicity of $\lhd$ (lemma 4.1.3). Assume that the left hand side of the implication. Pick any $h \in H$ such that $h \models_{\mathcal{I}} A \textit{ infers } \kappa$ and $h \models_{\mathcal{I}} \Box_A F$. Then, $\kappa \subseteq I(A \textit{ infers}, h)$. Pick any ρ and $h' \in H$ such that $h \sim^{\rho}_A h'$. Then, $\rho \lhd I(A \textit{ infers}, h)$. By monotonicity of $\lhd$, $\rho \lhd \kappa$. By assumption, $\rho(F) \models_{\mathcal{I}} \rho(F')$. Since, $h' \models_{\mathcal{I}} \rho(F)$, it follows that $h' \models_{\mathcal{I}} \rho(F')$. Since ρ and h' were chosen arbitrarily, $h \models_{\mathcal{I}} \Box_A F'$. $\Box$

Obviously, lemma 4.3.2 generalizes to a set of statements in place of F.

The weakening of normality in lemma 4.3.2 quantifies over the domain of $\lhd$. However, we can weaken lemma 4.3.2 by substituting the left-hand side of the implication ($\implies$) by a statement schema. To this end, we introduce some notation. To begin with, terms are extended with variables. Let open terms t be generated by:

$$t, t' ::= x \mid c \mid t \cdot t' \mid \{t\}_{t'}$$

where x ranges over a countable set of variables, and as before, $c \in \mathcal{C}$. Let $Keys(t)$ be the set of open terms applied as keys in t. For example, $Keys(\{x \cdot \{c, A\}_y\}_{c'}) = \{y, c'\}$. In detail:

$$\begin{aligned} Keys(\{t\}_{t'}) &= \{t'\} \cup Keys(t) \cup Keys(t') \\ Keys(t \cdot t') &= Keys(t) \cup Keys(t') \\ Keys(c) &= \emptyset \\ Keys(x) &= \emptyset \end{aligned}$$

If X is a set of open terms, let $Keys(X) = \bigcup_{t \in X} Keys(t)$. An assignment is a function V from variables to messages. Write $|t|_V$ for the result of replacing each variable x in t with its assigned image $V(x)$. Write $|\{t_1, \ldots, t_n\}|_V$ for the set $\{|t_1|_V, \ldots, |t_n|_V\}$.

Lemma 4.3.3 *If* $\rho \lhd |Keys(t)|_V$, *then* $\rho(|t|_V) = |t|_{\rho \circ V}$

Proof By induction over the structure of t. For the base step, t is a variable or an atom. The case when t is a variable is immediate. If t is an atom c, then $|t|_V = c$, i.e., $\rho(|t|_V) = \rho(c) = c$ by requirement (3) in the definition 4.1.1. But, $c = |c|_{\rho \circ V}$. For the induction step, assume that the property holds for open terms t_1 and t_2, i.e. $\rho \lhd |Keys(t_1)|_V \Rightarrow |t_1|_{\rho \circ V} = \rho(|t_1|_V)$ and $\rho \lhd |Keys(t_2)|_V \Rightarrow |t_2|_{\rho \circ V} = \rho(|t_2|_V)$. Assume that $\rho \lhd |Keys(\{t_1\}_{t_2})|_V$. Then, $\rho \lhd |Keys(t_1)|_V \cup |Keys(t_2)|_V \cup \{|t_2|_V\}$. By monotonicity of $\lhd$ (lemma 4.1.3),

$$\rho \lhd |Keys(t_1)|_V \text{ and } \rho \lhd |Keys(t_2)|_V \text{ and } \rho \lhd \{|t_2|_V\} \tag{4.13}$$

From (4.13) and requirement (1) in the definition 4.1.1, we get $\rho(\{|t_1|_V\}_{|t_2|_V}) = \{\rho(|t_1|_V)\}_{\rho(|t_2|_V)}$, i.e., $\rho(|\{t_1\}_{t_2}|_V) = \{\rho(|t_1|_V)\}_{\rho(|t_2|_V)}$. Also from (4.13), by the induction assumption, $\rho(|t_1|_V) = |t_1|_{\rho \circ V}$ and $\rho(|t_2|_V) = |t_2|_{\rho \circ V}$. Thus, $\rho(|\{t_1\}_{t_2}|_V) = \{|t_1|_{\rho \circ V}\}_{|t_2|_{\rho \circ V}} = |\{t_1\}_{t_2}|_{\rho \circ V}$. In a similar way, we can show that also pairing preserves the property. □

Open statements α are statements with messages M replaced by open terms t:

$$\alpha, \alpha' ::= \; p(t_1, ..., t_n) \mid \Box_A \alpha \mid \alpha \wedge \alpha' \mid \neg \alpha$$

The function *Keys* is lifted to open statements in the expected way: $Keys(\alpha) = Keys(Terms(\alpha))$, where $Terms(\alpha)$ is the set of open terms occurring in α. Assignments are also extended to open statements in the expected way: $|\alpha|_V$ is the result of replacing each variable x in α with its assigned image $V(x)$. Write $\alpha \models_{\mathcal{I}} \beta$ if $|\alpha|_V \models_{\mathcal{I}} |\beta|_V$, for all assignments V. Finally, by combining lemma 4.3.2 with lemma 4.3.3, we get the following weakening of normality.

Theorem 4.3.4 (Weak Normality)

$$\alpha \models_{\mathcal{I}} \alpha' \implies A \textit{ infers } Keys(\alpha, \alpha'), \Box_A \alpha \models_{\mathcal{I}} \Box_A \alpha'$$

Proof Assume that $\alpha \models_{\mathcal{I}} \alpha'$. Pick any assignment V and any permutation $\rho \lhd |Keys(\alpha, \alpha')|_V$. By monotonicity of $\lhd$ (lemma 4.1.3), $\rho \lhd |Keys(\alpha)|_V$ and $\rho \lhd |Keys(\alpha')|_V$. By lemma 4.3.3, $\rho(|\alpha|_V) = |\alpha|_{\rho \circ V}$ and $\rho(|\alpha'|_V) = |\alpha'|_{\rho \circ V}$. Thus, $\rho(|\alpha|_V) \models_{\mathcal{I}} \rho(|\alpha'|_V)$ is an instance of the assumption that $\alpha \models_{\mathcal{I}} \alpha'$. Since ρ was chosen arbitrarily, lemma 4.3.2 implies $A \textit{ infers } |Keys(\alpha, \alpha')|_V, \Box_A |\alpha|_V \models_{\mathcal{I}} \Box_A |\alpha'|_V$. Since V is arbitrary, $A \textit{ infers } Keys(\alpha, \alpha'), \Box_A \alpha \models_{\mathcal{I}} \Box_A \alpha'$. □

Again, weak normality generalizes to a set of open statements in the place of α. Like permutation normality (lemma 4.3.2), weak normality formalizes the intuition that knowledge is closed under feasibly computable entailments.

Example 4.3.5 *In any interpreted system* $\mathcal{I}$, *we have:* $exists\{x\}_y \models_{\mathcal{I}} exists\, x$. *Since* $Keys(\{\{x\}_y, x\}) = \{y\}$, *weak normality (theorem 4.3.4) yields:*

$$A \textit{ infers } y, \Box_A exists\{x\}_y \models_{\mathcal{I}} \Box_A exists\, x$$

I.e., $A \textit{ infers } K, \Box_A exists\{M\}_K \models_{\mathcal{I}} \Box_A exists\, M$, *for any* M *and* K.

Example 4.3.6 *Continuing example 4.3.1, we have:* $\models_{\mathcal{I}} \{x\}_y$ *contains* x*. By weak normality (theorem 4.3.4),*

$$A \textit{ infers } y \models_{\mathcal{I}} \Box_A \{x\}_y \textit{ contains } x$$

since $\mathit{Keys}(\{\{x\}_y, x\}) = \{y\}$.

As example 4.3.6 illustrates, an agent knows what is inside an encryption if the agent knows the key to the encryption. However, as suggested by the replay attack on the mix in section 1.2, sometimes an agent knows what is inside an encryption even though the agent cannot decrypt it. The following example illustrates this point.

Example 4.3.7 *Consider an interpreted system* $\mathcal{I}$ *based on a message passing system that implements the Needham Schröder Shared Key Protocol [65] between two principals* A *and* B *and a key server.*[1] *If principal* A *receives a message of the form* $\{N \cdot B \cdot K \cdot x\}_{K_A}$*, where* K_A *is A:s server key, then the message must originate from the server and* x *must be the ticket intended for principal* B*:*

$$A \textit{ receives } \{N \cdot B \cdot K \cdot x\}_{K_A},\ K_A \textit{ server key of } A \models_{\mathcal{I}} x \textit{ contains } K \cdot A$$

(We leave the interpretation $I(\textit{server key of}\, A, h)$ *unspecified.) By weak normality (theorem 4.3.4),*

$$\Box_A A \textit{ receives } \{N \cdot B \cdot K \cdot \{K \cdot A\}_{K_B}\}_{K_A}, A \textit{ infers } K_A,\ \Box_A K_A \textit{ server key of } A$$

$$\models_{\mathcal{I}} \Box_A \{K \cdot A\}_{K_B} \textit{ contains } K \cdot A$$

By receive introspection (proposition 4.2.6),

$$A \textit{ receives } \{N \cdot B \cdot K \cdot \{K \cdot A\}_{K_B}\}_{K_A}, A \textit{ infers } K_A,\ \Box_A K_A \textit{ server key of } A$$

$$\models_{\mathcal{I}} \Box_A \{K \cdot A\}_{K_B} \textit{ contains } K \cdot A$$

Thus, if principal A *receives* $\{N \cdot B \cdot K \cdot \{K \cdot A\}_{K_B}\}_{K_A}$ *from the server then* A *knows that* $K \cdot A$ *is the content of the ticket* $\{K \cdot A\}_{K_B}$*, even though* A *does not know the decryption key* K_B*.*

[1] *The details of the protocol are not important.*

Chapter 5

Message Deduction

The notion of *deduced messages* ("known messages") plays a central role in formal analysis of security protocols. Some simple forms of secrecy goals can be formulated directly in terms of message deduction: A value is secret if an unauthorized party cannot deduce the value. Moreover, message deduction is used to define indistinguishability relations, for instance in AT-style semantics (section 3.3), but also outside epistemic logic (cf. [6, 15]). In section 4.1, the relativized indistinguishability $\sim_A^\rho$ was defined in terms of messages deduced by agent A. In this chapter, we examine alternative definitions of message deduction and their effect on the epistemic modality.

5.1 Dolev-Yao Deduction

The set of known messages (in our setting: the interpretation of predicate $A\ infers$) is customarily defined through a Dolev-Yao style message inference relation [28]: An agent knows a message if the agent has observed the message (typically: received it), or if the message can be obtained from already known messages through a set of feasible computable operations. For message spaces based on pairing and symmetric cryptography, there are two versions of Dolev-Yao style message inference, one weaker than the other. According to the weaker definition, an agent knows a message if the agent can obtain the message by un-pairing and decryption, starting from directly observed messages.

Definition 5.1.1 (Weak Dolev-Yao) *An interpreted system $\mathcal{I}$ is weak Dolev-Yao, if and only if, $I(A\ infers, h)$ is the least set of messages such that:*

1. $messages(h|A) \subseteq I(A\ infers, h)$
2. $M \cdot M' \in I(A\ infers, h) \implies M \in I(A\ infers, h)$
3. $M \cdot M' \in I(A\ infers, h) \implies M' \in I(A\ infers, h)$

4. $\{M\}_K, K \in I(A\ infers, h) \implies M \in I(A\ infers, h)$

According to the stronger definition, an agent knows a message if the agent can obtain the message by un-pairing, decryption, pairing and encryption, starting from directly observed messages.

Definition 5.1.2 (Strong Dolev-Yao) *An interpreted system $\mathcal{I}$ is strong Dolev-Yao, if and only if, $I(A\ infers, h)$ is the least set of messages such that:*

1. $messages(h|A) \subseteq I(A\ infers, h)$
2. $M \cdot M' \in I(A\ infers, h) \implies M \in I(A\ infers, h)$
3. $M \cdot M' \in I(A\ infers, h) \implies M' \in I(A\ infers, h)$
4. $\{M\}_K, K \in I(A\ infers, h) \implies M \in I(A\ infers, h)$
5. $M, M' \in I(A\ infers, h) \implies M \cdot M' \in I(A\ infers, h)$
6. $M, K \in I(A\ infers, h) \implies \{M\}_K \in I(A\ infers, h)$

Thus, in weak Dolev-Yao, only de-constructing ("analysing") operations – un-pairing and decryption – are used in the message inference, while in strong Dolev-Yao, also "synthesizing" operations - pair forming and encryption – are used. We say that $\mathcal{I}$ is Dolev-Yao, if $\mathcal{I}$ is either weak Dolev-Yao or $\mathcal{I}$ is strong Dolev-Yao.

Proposition 5.1.3 *If $\mathcal{I}$ is Dolev-Yao then $\mathcal{I}$ is introspective.*

Proof Assume that $\mathcal{I}$ is weak Dolev-Yao and $h \sim_A^\rho h'$. We show $\rho(I(A\ infers, h)) \subseteq I(A\ infers, h')$ by induction on the inference length to reach $M \in I(A\ infers, h)$. Base case, M is inferred in one step, i.e., $M \in messages(h|A)$. By assumption, $\rho(h|A) = h'|A$, and so $\rho(M) \in messages(h'|A)$. Induction step: Assume that $M \in I(A\ infers, h)$ is inferred in n steps. Case (1), the last inference step in the derivation to M was left un-pairing, i.e., (2) in definition 5.1.1. Then, $M \cdot M' \in I(A\ infers, h)$ is derived in less than n steps, for some M'. By induction assumption $\rho(M \cdot M') \in I(A\ infers, h')$. By assumption, $\rho \lhd I(A\ infers, h)$. By condition (2) in definition 4.1.1, $\rho(M \cdot M') = \rho(M) \cdot \rho(M')$. So, $\rho(M) \cdot \rho(M') \in I(A\ infers, h')$. By left-un-pairing, i.e., (2) in definition 5.1.1, $\rho(M) \in I(A\ infers, h')$. Case (2), the last inference step in the derivation to M was right-un-pairing: Similar to case (1). Case (3), the last inference step in the derivation to M was decryption, i.e., (4) in definition 5.1.1: Similar to case (1), but using condition (1) in the definition 4.1.1. This completes the proof that $\rho(I(A\ infers, h)) \subseteq I(A\ infers, h')$. Continuing, we show that $\rho^{-1}(I(A\ infers, h')) \subseteq I(A\ infers, h)$ by induction on the inference length to reach $M \in I(A\ infers, h')$. Base case, M is inferred in one step, i.e., $M \in messages(h'|A)$. By assumption, $\rho^{-1}(h'|A) = h|A$, and so $\rho^{-1}(M) \in messages(h|A)$. Induction step: Assume that $M \in I(A\ infers, h')$ is inferred in n steps. Case (1), the last inference step in the derivation to M was left-un-pairing, i.e., (2) in definition 5.1.1. Then, $M \cdot M' \in I(A\ infers, h')$ is derived in less than n steps, for some M'. By induction

assumption, $\rho^{-1}(M \cdot M') \in I(A\,infers, h)$. By assumption, $\rho \lhd I(A\,infers, h)$. By symmetry of $\lhd$ (lemma 4.1.3), $\rho^{-1} \lhd \rho(I(A\,infers, h))$. By condition (2) in definition 4.1.1, $\rho^{-1}(M \cdot M') = \rho^{-1}(M) \cdot \rho^{-1}(M')$. So, $\rho^{-1}(M) \cdot \rho^{-1}(M') \in I(A\,infers, h)$. By left unpairing, i.e., (2) in definition 5.1.1, $\rho^{-1}(M) \in I(A\,infers, h)$. Case (2), the last inference step in the derivation to M was right un-pairing: Similar to case (1). Case (3), the last inference step in the derivation to M was decryption, i.e., (4) in definition 5.1.1: Then, for some K, both $\{M\}_K \in I(A\,infers, h')$ and $K \in I(A\,infers, h')$ are derived in less than n steps. By induction assumption, $\rho^{-1}(\{M\}_K), \rho^{-1}(K) \in I(A\,infers, h)$, i.e.,

$$K \in \rho(I(A\,infers, h)) \tag{5.1}$$

By assumption, $\rho \lhd I(A\,infers, h)$. By symmetry of $\lhd$ (lemma 4.1.3), we obtain that $\rho^{-1} \lhd \rho(I(A\,infers, h))$. So, by (5.1) and monotonicity of $\lhd$ (lemma 4.1.3), $\rho^{-1} \lhd \{K\}$. By condition (1) in definition 4.1.1, $\rho^{-1}(\{M\}_K) = \{\rho^{-1}(M)\}_{\rho^{-1}(K)}$. Thus, $\{\rho^{-1}(M)\}_{\rho^{-1}(K)} \in I(A\,infers, h)$. By decryption, i.e., (4) in definition 5.1.1, $\rho^{-1}(M) \in I(A\,infers, h)$. The proposition is shown for strong Dolev-Yao $\mathcal{I}$ in the same way. □

Corollary 5.1.4 *The following are valid in Dolev-Yao systems $\mathcal{I}$:*

1. $\Box_A F \rightarrow \Box_A \Box_A F$
2. $\neg\Box_A F \rightarrow \Box_A \neg\Box_A F$

Proof From propositions 4.2.5 and 5.1.3. □

Lemma 5.1.5 *Let $\mathcal{I}$ be the protocol implementation in section 2.3. Assume that $\mathcal{I}$ is Dolev-Yao. For all protocol executions $h \in H$:*

$$\mathcal{C} \cap I(spy\,infers, h) = \emptyset$$

Proof By routine induction. Pick $h \in H$. Then, $messages(h|spy) = \{\{M_0\}_{K_0}\}$ for some $M_0 \in \mathcal{T}$ and some $K_0 \in \mathcal{C}$. Case (A), $\mathcal{I}$ is weak Dolev-Yao: We show that $I(spy\,infers, h) \subseteq \{\{M_0\}_{K_0}\}$, by induction on the derivation length to reach $M \in I(A\,infers, h)$. Base case: Immediate. Induction step: Assume that $M \in I(spy_A\,infers, h)$ is inferred in n steps. Sub-case (1), the last inference step in the derivation to reach $M \in I(spy\,infers, h)$ was decryption, i.e., (4) in definition 5.1.1. Then, for some K, $\{M\}_K \in I(spy\,infers, h)$ and $K \in I(spy\,infers, h)$ are derived in less than n steps. By induction assumption, $\{M\}_K, K \in \{\{M_0\}_{K_0}\}$. This is impossible, since $K \neq \{M\}_K$. Thus, the last derivation step was not decryption. Similarly, we obtain that the last inference step cannot have been un-pairing. This completes the induction step. Case (B), $\mathcal{I}$ is strong Dolev-Yao: We show that $I(spy\,infers, h)$ is the set of messages generated by:

$$M, M' ::= \{M_0\}_{K_0} \mid M \cdot M' \mid \{M\}_{M'} \tag{5.2}$$

The proof is by induction on the derivation length to reach $M \in I(A\,infers, h)$. Base case: immediate. Induction step: Assume that $M \in I(spy_A\,infers, h)$ is inferred in n steps. Case (1), the last inference step in the derivation to reach $M \in I(spy\,infers, h)$ was decryption, i.e., (4) in definition 5.1.2. Then, for some K, $\{M\}_K \in I(spy\,infers, h)$ and $K \in I(spy\,infers, h)$ are derived in less than n steps. By induction assumption, $\{M\}_K$ and K are generated by (5.2). Since $K_0 \in \mathcal{C}$, K_0 is not generated by (5.2), i.e., M is generated by (5.2). Case (2), the last inference step in the derivation to reach $M \in I(spy\,infers, h)$ was left-un-pairing, i.e., (2) in definition 5.1.2. Then, for some M', $M \cdot M' \in I(spy\,infers, h)$ is derived in less than n steps. By induction assumption, $M \cdot M'$ is generated by (5.2), i.e., M is generated by (5.2). Case (3), right-un-pairing: Similar to case (2). Case (4), the last inference step in the derivation to reach $M \in I(spy\,infers, h)$ was pairing, i.e., (4) in definition 5.1.2. Then, $M = M' \cdot M''$ for some $M', M'' \in I(spy\,infers, h)$ derived in less than n steps. By induction assumption, M' and M'' are generated by (5.2), i.e., $M' \cdot M''$ is generated by (5.2). Case (5), the last inference step in the derivation to reach $M \in I(spy\,infers, h)$ was encryption, i.e., (5) in definition 5.1.2: Similar to case (4). □

Corollary 5.1.6 (Crowds-Style Protocol) *Let $\mathcal{I}$ be the protocol implementation in section 2.3. If $\mathcal{I}$ is Dolev-Yao, it satisfies specifications (2.3), (2.4), (2.5) and (2.6).*

Proof From lemma 5.1.5 and lemma 4.2.8. □

5.2 Duck-Duck-Goose Counterexample

It has been argued that Dolev-Yao style message inferences can yield counter intuitive results (cf. [43]). The counterexample is the following (artificial) style of protocol, which, following [43], we refer to as the Duck-Duck-Goose Protocol. An agent A generates a random bit sequence $bit_1, \cdots, bit_n$, and sends the sequence, bit by bit, to another agent B. When B has received all n bits, B sends a fresh nonce N to a third agent C, encrypted using the bit sequence $bit_1 \cdots bit_n$ as encryption key. Agent C merely forwards the message to A:

$$\begin{array}{lcl} A & \longrightarrow & B : bit_1 \\ & \vdots & \\ A & \longrightarrow & B : bit_n \\ B & \longrightarrow & C : \{N\}_{(bit_1 \cdots bit_n)} \\ C & \longrightarrow & A : \{N\}_{(bit_1 \cdots bit_n)} \end{array}$$

where the encryption key $bit_1 \cdots bit_n$ abbreviates the iterated pairing construction $pair(bit_1, pair(bit_2, \ldots bit_n) \ldots)$ We assume two local spies, spy_A and spy_C, who

observe the network traffic in and out of agent A and agent C respectively. Intuitively, spy_A deduces the key $(bit_1 \cdots bit_n)$, since spy_A observes each bit being sent from agent A to agent B. On the other hand, spy_B does not deduce the key $(bit_1 \cdots bit_n)$, since spy_C observes only the encryption sent via agent B. As we show next, Dolev-Yao deduction does not respect these intuitions.

We implement the protocol in a message passing system with spying (see example 2.2.2). Fix two different bits $0, 1 \in \mathcal{C}$. Fix a key-length $n > 1$. Assume that $\mathcal{A}$ contains, at least, five agents: A, B, C, spy_A and spy_C. Assume a function $realm : \mathcal{A} \longrightarrow 2^{\mathcal{A}}$ such that:

$$\begin{aligned} realm(spy_A) &= \{A, spy_A\}, \\ realm(spy_C) &= \{C, spy_C\} \end{aligned}$$

(Assume also that $A' \in realm(A')$ for every $A' \in \mathcal{A}$.) According to *realm*, spy_A observes the sending and receiving of agent A, while spy_C observes the actions of C. Let $\mathcal{S}_{DDG} = \langle \Pi, H, | \rangle$ be the message passing system with spying based on *realm* and where the set H consists of all histories of the form:

$$i \cdot (A \longrightarrow B : bit_1) \cdots$$
$$\cdots (A \longrightarrow B : bit_n) \cdot (B \longrightarrow C : \{N\}_{(bit_1 \cdots bit_n)}) \cdot (C \longrightarrow A : \{N\}_{(bit_1 \cdots bit_n)})$$

for any initialization i and any messages $bit_1, \ldots, bit_n, N$ such that:

- $bit_1, \ldots, bit_n \in \{0, 1\}$
- $N \in \mathcal{C} - \{0, 1\}$
- $i(A) = \{bit_1 \cdots bit_n\}$, $i(B) = \{N, 0, 1\}$, $i(C) = i(spy_A) = i(spy_C) = \{0, 1\}$

In initialization i, agent A creates the "secret" key $bit_1 \cdots bit_n$. The initialization provides other agents with the two bits 0 and 1, but not the specific bit sequence that A creates.[1] Let $\mathcal{I}_{DDG}$ be an interpreted system based on $\mathcal{S}_{DDG}$. As the following two propositions illustrate, we obtain counter intuitive results if $\mathcal{I}_{DDG}$ is Dolev-Yao.

Proposition 5.2.1 *The following is valid in $\mathcal{I}_{DDG}$, if it is weak Dolev-Yao:*

$$A\, received \{N\}_{bit_1 \cdots bit_n} \rightarrow \neg spy_A\, infers\, (bit_1 \cdots bit_n)$$

Proof Pick $h \in H$. Then, $messages(h|spy_A) = \{0, 1, \{N\}_{(bit_1 \cdots bit_n)}\}$ for some $bit_1, ..., bit_n \in \{0, 1\}$ and $N \in \mathcal{C}$. We show that $(spy_A\, infers, h) \subseteq messages(h|spy_A)$, by induction on the derivation length to reach $M \in I(A\, infers, h)$. Base case, immediate. Induction step: Assume that $M \in I(spy_A\, infers, h)$ is inferred in n steps. Case (1), the last inference step in the derivation to reach $M \in I(spy_A\, infers, h)$ was

[1] In part I of this thesis, there are no "public" constants, atomic messages that are known to all agents by default. To make the bits 0 and 1 "public", the initialization provides each agent with the values 0 and 1.

decryption, i.e., (4) in definition 5.1.1. Then, for some K, $\{M\}_K \in I(spy_A\ infers, h)$ and $K \in I(spy_A\ infers, h)$ are derived in less than n steps. By induction assumption, $\{M\}_K, K \in messages(h|spy_A)$. This is impossible, since $n > 1$. Thus, the last derivation step was not decryption. Similarly, we obtain that the last inference step cannot have been un-pairing. □

Proposition 5.2.1 is counter intuitive, since the spy on A should be able to infer the key $bit_1 \cdots bit_n$ used; The spy on A observes as each bit is sent by agent A and the spy knows that agents A, B and C are following the protocol.

Proposition 5.2.2 *The following is valid in $\mathcal{I}_{DDG}$, if it is strong Dolev-Yao:*

$$C\ received\ \{N\}_{bit_1 \cdots bit_n} \rightarrow spy_C\ infers\ (bit_1 \cdots bit_n)$$

Proof Pick $h \in H$. By construction, $\{0,1\} \subseteq messages(h|spy_C)$. By (1) in definition 5.1.2 and successive applications of (5) in definition 5.1.2, we obtain $bit_1 \cdots bit_n \in I(spy_C\ infers, h)$, since $bit_i \in \{0,1\}$. □

Proposition 5.2.2 is counter intuitive: The spy on C only observes the encryption C receives, and so should not be able to infer the key $bit_1 \cdots bit_n$. Unintended results for the predicate $A\ infers$ transfer to unintended results for the modality $\Box_A$, as the following two propositions illustrate.

Proposition 5.2.3 *The following is valid in $\mathcal{I}_{DDG}$, if it is weak Dolev-Yao:*

$$A\ received \{N\}_{bit_1 \cdots bit_n} \rightarrow \neg \Box_{spy_A} A\ rec\ N$$

Proof Assume that $h \models_{\mathcal{I}_{DDG}} A\ received \{N\}_{bit_1 \cdots bit_n}$. Pick $N' \in \mathcal{C}$ such that $N' \notin \{N, 0, 1\}$. Let $h' = [N - N'](h)$. (See section 4.1 for the notation $[N - N']$.) By construction of H, we have $h' \in H$. Let $\rho = [\{N\}_{bit_1 \cdots bit_n} - \{N'\}_{bit_1 \cdots bit_n}]$. We have $h'|spy_A = \rho(h|spy_A)$. From lemma 4.1.2 and proposition 5.2.1, $\rho \triangleleft I(spy_A\ infers, h)$. Thus, $h \sim^{\rho}_{spy_A} h'$. Since $\rho(N) = N \notin \{N', 0, 1\}$, $h' \not\models_{\mathcal{I}_{DDG}} A\ rec\ \rho(N)$. Thus, $h \not\models_{\mathcal{I}_{DDG}} \Box_{spy_A} A\ rec\ N$. □

Proposition 5.2.4 *The following is valid in $\mathcal{I}_{DDG}$, if it is strong Dolev-Yao:*

$$C\ received \{N\}_{bit_1 \cdots bit_n} \rightarrow \Box_{spy_C} C\ rec\ N$$

Proof Assume that $h \models_{\mathcal{I}_{DDG}} C\ received \{N\}_{bit_1 \cdots bit_n}$. Assume that $h \sim^{\rho}_{spy_C} h'$. Then, $C\ \texttt{receives}\ \rho(\{N\}_{bit_1 \cdots bit_n}) \in actions(h'|spy_C)$ and $\rho \triangleleft I(spy_C\ infers, h)$. By proposition 5.2.2, $\rho(\{N\}_{bit_1 \cdots bit_n}) = \{N\}_{bit_1 \cdots bit_n}$. Thus, $h' \models_{\mathcal{I}_{DDG}} C\ rec\ N$, i.e., $h' \models_{\mathcal{I}_{DDG}} C\ rec\ \rho(N)$. Since h' and ρ are arbitrary, $h \models_{\mathcal{I}_{DDG}} \Box_{spy_C} C\ rec\ N$. □

The Duck-Duck-Goose Protocol shows that weak Dolev-Yao deduction can be too restrictive, since it excludes messages that, intuitively, can be inferred by protocol specific dependencies. On the other hand, the protocol suggests that strong Dolev-Yao deduction can be too inclusive: As long as an adversary knows bits 0

and 1, the adversary knows every finite bit-sequence (of 0 and 1). This seems to contradict the assumption that agents have only limited computational powers.

The Duck-Duck-Goose protocol is artificial, and by itself, perhaps, not sufficient reason to abandon Dolev-Yao style interpretations of the predicate *A infers*. Indeed, in part II of this thesis, we shall use a Dolev-Yao style interpretation.[2] On the other hand, the Duck-Duck-Goose counterexample is theoretically interesting.

5.3 Message Deduction Reduced to Modality

Intuitively, an agent knows a message if the agent knows about the message, i.e., knows some relevant facts about the message. This suggests the following definition:

$$A \mathit{\ infers\ } K \leftrightarrow \Box_A \bigvee_p p(K) \tag{5.3}$$

where p ranges over a selected set of relevant predicates. For instance, if *exists* is the only relevant predicate:

$$A \mathit{\ infers\ } K \leftrightarrow \Box_A \mathit{exists}\, K \tag{5.4}$$

For simplicity, we only consider requirement (5.4); The results in this section generalize to requirement (5.3) and a set of relevant predicates.

However, the stipulation (5.4) requires a recursive definition, since the epistemic modality $\Box_A$ is defined through the interpretation of *A infers* (section 4.2).

Definition 5.3.1 (Fixed Point Interpretation) *An interpretation function I is a fixed point on a system $\mathcal{S}$, if condition (5.4) holds in the interpreted system $\mathcal{I} = \langle \mathcal{S}, I \rangle$.*

An inductive, rather than a co-inductive interpretation of *A infers* is appropriate, since $I(A \mathit{\ infers}, h)$ should assign the set of keys that agent A has gathered some positive information about at history h. We introduce some terminology. Two interpretation functions I and I' on system $\mathcal{S}$ are variants of each other if they agree on all predicates except *infers*, i.e., if $I(p, h) = I'(p, h)$ for all predicates $p \in \mathcal{P} - \{A \mathit{\ infers} \mid A \in \mathcal{A}\}$ and all $h \in H$. Variant I is smaller than I', $I \leq I'$ if $I(A \mathit{\ infers}, h) \subseteq I'(A \mathit{\ infers}, h)$ for all $A \in \mathcal{A}$ and all $h \in H$. I is strictly smaller than I' if I is smaller than I' and I' is not smaller than I.

Definition 5.3.2 (Inductive Interpretation) *Interpretation I is inductive on system $\mathcal{S}$, if I is a fixed point on $\mathcal{S}$ and there is no strictly smaller variant of I which is a fixed point on $\mathcal{S}$.*

Theorem 5.3.3 *There is a unique inductive interpretation on every system, i.e., every interpretation I on a system $\mathcal{S}$ has exactly one inductive variant.*

[2] Although, generalized to arbitrary one-way functions.

Proof From monotonicity of $\lhd$ (lemma 4.1.3). Assume an interpretation I_{orig} on a system $\mathcal{S}$. Let f be a function in the set of variants of I_{orig}, such that for each variant I of I_{orig}, $f(I)$ is the variant of I_{orig} such that:

$$f(I)(A\ infers, h) = \{K \mid\ h \models_{\langle \mathcal{S}, I\rangle} \Box_A\, exists\, K\}$$

An inductive variant of I_{orig} is, by definition, a least fixed point of the function f. To see that f is monotone, assume that I is smaller than I', i.e., $I(A\ infers, h) \subseteq I'(A\ infers, h)$ for all $A \in \mathcal{A}$ and $h \in H$. Assume that $K \in f(I)(A\ infers, h)$, i.e., $h \models_{\langle \mathcal{S}, I\rangle} \Box_A\, exists\, K$. We proceed to show that $h \models_{\langle \mathcal{S}, I'\rangle} \Box_A\, exists\, K$. Pick any $h' \in H$ and permutation ρ such that $h \sim_A^\rho h'$ in $\langle \mathcal{S}, I'\rangle$, i.e., such that $\rho(h|A) = h'|A$ and $\rho \lhd I'(A\ infers, h)$. By monotonicity of $\lhd$ (lemma 4.1.3), $\rho \lhd I(A\ infers, h)$. Thus, $h \sim_A^\rho h'$ in $\langle \mathcal{S}, I\rangle$. By assumption, $h' \models_{\langle \mathcal{S}, I\rangle} exists\, \rho(K)$, i.e., $h' \models_{\langle \mathcal{S}, I'\rangle} exists\, \rho(K)$. Since h' and ρ were chosen arbitrary, it follows that $h \models_{\langle \mathcal{S}, I'\rangle} \Box_A K$, i.e., $K \in f(I')(A\ infers, h)$. This establishes that f is monotone, and therefore has a unique least fixed point. □

Theorem 5.3.4 *If $\mathcal{I}$ is inductive then $\mathcal{I}$ is introspective.*

Proof Assume an inductive interpreted system $\mathcal{I}$ based on system $\mathcal{S}$. We prove, using fixed point induction, that:

$$h \sim_A^\rho h' \Rightarrow \rho(I(A\ infers, h)) \supseteq I(A\ infers, h')$$

Subset inclusion, i.e., $\rho(I(A\ infers, h)) \subseteq I(A\ infers, h')$, is shown analogously. Let I_j be the interpretation function at step j in the fixed point construction of the proof of theorem 5.3.3, such that $I_0(A\ infers, h) = \emptyset$, $I_{j+1} = f(I_j)$, and $I_\lambda(A\ infers, h) = \bigcup_{j<\lambda} I_j(A\ infers, h)$, if λ is a limit ordinal. We show for all j that

$$\rho \lhd I_j(A\ infers, h) \wedge \rho(h|A) = h'|A \Rightarrow \rho(I_j(A\ infers, h)) \supseteq I_j(A\ infers, h') \quad (5.5)$$

The property holds for I_0, since $I_0(A\ infers, h') = \emptyset$. For successor ordinals, assume that (5.5) holds for j. Assume that $\rho(h|A) = h'|A$ and $\rho \lhd I_{j+1}(A\ infers, h)$. By monotonicity of $\lhd$ (lemma 4.1.3), since $I_j(A\ infers, h) \subseteq I_{j+1}(A\ infers, h)$:

$$\rho \lhd I_j(A\ infers, h) \quad (5.6)$$

From (5.6), by induction assumption, $\rho(I_j(A\ infers, h)) \supseteq I_j(A\ infers, h')$. Thus, since ρ is 1-1,

$$\rho^{-1}(I_j(A\ infers, h')) \subseteq I_j(A\ infers, h) \quad (5.7)$$

Also from (5.6), $\rho^{-1} \lhd \rho(I_j(A\ infers, h))$, by symmetry of $\lhd$ (lemma 4.1.3). Thus, by monotonicity of $\lhd$ (lemma 4.1.3):

$$\rho^{-1} \lhd I_j(A\ infers, h') \quad (5.8)$$

Pick any $K' \in I_{j+1}(A\ infers, h')$. Since ρ is 1-1, there is some K such that $K' = \rho(K)$. Thus, $\rho(K) \in I_{j+1}(A\ infers, h')$, i.e.,

$$h' \models_{\langle \mathcal{S}, I_j \rangle} \Box_A\, exists\, \rho(K) \tag{5.9}$$

We proceed to show that $K \in I_{j+1}(A\ infers, h)$, i.e., $h \models_{\langle \mathcal{S}, I_j \rangle} \Box_A\, exists\, K$. Pick any permutation ρ' and any $h'' \in H$ such that $\rho'(h|A) = h''|A$ and $\rho' \lhd I_j(A\ infers, h)$. Thus, by (5.7) and monotonicity of $\lhd$ (lemma 4.1.3), $\rho' \lhd \rho^{-1}(I_j(A\ infers, h'))$. From this and (5.8) and transitivity of $\lhd$ (lemma 4.1.3), $\rho' \circ \rho^{-1} \lhd I_j(A\ infers, h')$. But $\rho' \circ \rho^{-1}(h'|A) = \rho'(\rho^{-1}(h'|A)) = \rho'(h|A) = h''|A$. By (5.9), we obtain that $h'' \models_{\langle \mathcal{S}, I_j \rangle} exists\, \rho'(K)$. Since ρ' and h'' are arbitrary, it follows that $h \models_{\langle \mathcal{S}, I_j \rangle} \Box_A\, exists\, K$, which completes the successor part of the induction argument. The limit case is routine. □

Corollary 5.3.5 *The following are valid in inductive interpreted systems $\mathcal{I}$:*

1. $\Box_A F \rightarrow \Box_A \Box_A F$
2. $\neg \Box_A F \rightarrow \Box_A \neg \Box_A F$

Proof From proposition 4.2.5 and theorem 5.3.4. □

In contrast to Dolev-Yao interpretations, the inductive interpretation behaves as intended for the Duck-Duck-Goose protocol. Let $\mathcal{I}_{DDG}$ be an interpreted system based on the implementation $\mathcal{S}_{DDG}$ of the Duck-Duck-Goose protocol (in section 5.2).

Proposition 5.3.6 *The following are valid in $\mathcal{I}_{DDG}$, if it is inductive:*

1. $A\ received \{N\}_{bit_1 \cdots bit_n} \rightarrow spy_A\ infers\ (bit_1 \cdots bit_n)$
2. $C\ received \{N\}_{bit_1 \cdots bit_n} \rightarrow \neg spy_C\ infers\ (bit_1 \cdots bit_n)$

Proof (1): Assume that $h \models_{\mathcal{I}_{DDG}} A\ received \{N\}_{bit_1 \cdots bit_n}$. By construction of H, we have initialization i such that history h is:

$$i \cdot (A \longrightarrow B : bit_1) \cdots$$
$$\cdots (A \longrightarrow B : bit_n) \cdot (B \longrightarrow C : \{N\}_{(bit_1 \cdots bit_n)}) \cdot (C \longrightarrow A : \{N\}_{(bit_1 \cdots bit_n)})$$

Since $realm(spy_A) = \{A, spy_A\}$, local history $h|spy_A$ is:

$$\mathtt{init}\, i(spy_A) \cdot (A\, \mathtt{sends}\, bit_1) \cdots$$
$$\cdots (A\, \mathtt{sends}\, bit_n) \cdot (A\, \mathtt{receives}\, \{N\}_{(bit_1 \cdots bit_n)})$$

Pick any $h' \in H$ and any permutation ρ such that $h \sim^{\rho}_{spy_A} h'$. Then, local history $h'|spy_A$ is:

$$\mathtt{init}\, \rho(i(spy_A)) \cdot (A\, \mathtt{sends}\, \rho(bit_1)) \cdots$$
$$\cdots (A\, \mathtt{sends}\, \rho(bit_n)) \cdot (A\, \mathtt{receives}\, \rho(\{N\}_{(bit_1 \cdots bit_n)}))$$

i.e., since $bit_i \in \mathcal{C}$, $h'|spy_A$ is:

$$\begin{aligned}&\texttt{init}\,\rho(i(spy_A)) \cdot (A\,\texttt{sends}\,bit_1)\cdots\\ \cdots (A\,\texttt{sends}\,bit_n) \cdot (A\,&\texttt{receives}\,\rho(\{N\}_{(bit_1\cdots bit_n)}))\end{aligned}$$

i.e., by construction of H, $h'|spy_A$ is:

$$\begin{aligned}&\texttt{init}\,\rho(i(spy_A)) \cdot (A\,\texttt{sends}\,bit_1)\cdots\\ \cdots (A\,\texttt{sends}\,bit_n) \cdot (A\,&\texttt{receives}\,\{N'\}_{(bit_1\cdots bit_n)})\end{aligned}$$

for some $N' \in \mathcal{A}$. Thus, $h' \models_{\mathcal{I}_{DDG}} exists\, bit_1 \cdots bit_n$, i.e., since $\rho(bit_1 \cdots bit_n) = \rho(bit_1)\cdots\rho(bit_n) = bit_1 \cdots bit_n$, we have $h' \models_{\mathcal{I}_{DDG}} exists\, \rho(bit_1 \cdots bit_n)$. Since h' and ρ are arbitrarily chosen, we obtain $h \models_{\mathcal{I}_{DDG}} \Box_{spy_A} exists\, bit_1 \cdots bit_n$, i.e., since $\mathcal{I}_{DDG}$ is inductive, we have $h \models_{\mathcal{I}_{DDG}} spy_A\ infers\, bit_1 \cdots bit_n$. (2): Assume that

$$h \models_{\mathcal{I}_{DDG}} C\ received\, \{N\}_{bit_1\cdots bit_n} \tag{5.10}$$

Let I_i be the interpretation function at step i in the fixed point construction of the proof of theorem 5.3.3, such that $I_0(A\ infers, h) = \emptyset$, $I_{i+1} = f(I_i)$, and $I_\lambda(A\ infers, h) = \bigcup_{i<\lambda} I_i(A\ infers, h)$, if λ is a limit ordinal. We show that for each ordinal i: $(bit_1 \cdots bit_n) \notin I_i(spy_C\ infers, h)$. Base case: $I_0 = \emptyset$. Induction step, for successor ordinals: Assume that $(bit_1 \cdots bit_n) \notin I_i(spy_C\ infers, h)$. From (5.10), by construction of H, there is initialization i such that h is:

$$\begin{aligned}&i \cdot (A \longrightarrow B : bit_1)\cdots\\ \cdots (A \longrightarrow B : bit_n) \cdot (B \longrightarrow C : \{N\}_{(bit_1\cdots bit_n)}) &\cdot (C \longrightarrow A : \{N\}_{(bit_1\cdots bit_n)})\end{aligned}$$

and $i(spy_C) = \emptyset$. Since $realm(spy_C) = \{C, spy_C\}$, $h|spy_C$ is:

$$\texttt{init}\, i(spy_C) \cdot (C\,\texttt{receives}\,\{N\}_{(bit_1\cdots bit_n)}) \cdot (C\,\texttt{sends}\,\{N\}_{(bit_1\cdots bit_n)})$$

Pick any $bit'_1, \ldots, bit'_n \in \{0,1\}$ such that $(bit_1 \cdots bit_n) \neq (bit'_1 \cdots bit'_n)$. By construction of H, $h \not\models exists\, (bit'_1 \cdots bit'_n)$, i.e., by proposition 4.2.4, $(bit'_1 \cdots bit'_n) \notin I_i(spy_C\ infers, h)$. Let $\rho = [\{N\}_{(bit_1\cdots bit_n)} - \{N\}_{(bit'_1\cdots bit'_n)}]$. By lemma 4.1.2,

$$\rho \triangleleft I_i(spy_C\ infers, h) \tag{5.11}$$

By construction of H, there is $h' \in H$ and initialization i' such that h' is:

$$\begin{aligned}&i' \cdot (A \longrightarrow B : bit'_1)\cdots\\ \cdots (A \longrightarrow B : bit'_n) \cdot (B \longrightarrow C : \{N\}_{(bit'_1\cdots bit'_n)}) &\cdot (C \longrightarrow A : \{N\}_{(bit'_1\cdots bit'_n)})\end{aligned}$$

I.e.,

$$h'|spy_C = \texttt{init}\, i'(spy_C) \cdot (C\,\texttt{receives}\,\{N\}_{(bit'_1\cdots bit'_n)} \cdot (C\,\texttt{sends}\,\{N\}_{(bit'_1\cdots bit'_n)}))$$

I.e., since $i(spy_C) = i'(spy_C) = \emptyset$:

$$h'|spy_C = \rho(h|spy_C) \tag{5.12}$$

From (5.11) and (5.12):

$$h \sim^{\rho}_{spy_C} h' \text{ in } \langle \mathcal{S}_{DDG}, I_i \rangle \tag{5.13}$$

By construction of h', we have $h' \not\models exists\,(bit_1 \cdots bit_n)$, i.e., since $\rho(bit_1 \cdots bit_n) = \rho(bit_1) \cdots \rho(bit_n) = bit_1 \cdots bit_n$,

$$h' \not\models_{\langle \mathcal{S}_{DDG}, I_i \rangle} exists\,\rho(bit_1 \cdots bit_n) \tag{5.14}$$

From (5.13) and (5.14):

$$h \not\models_{\langle \mathcal{S}_{DDG}, I_i \rangle} \Box_{spy_C}\, exists\, bit_1 \cdots bit_n$$

I.e., $(bit_1 \cdots bit_n) \notin I_{i+1}(spy_C\ infers, h)$. The induction step for limit ordinals is immediate. □

5.4 Relationship to Weak Dolev-Yao

Any fixed point interpretation is at least as inclusive as weak Dolev-Yao.

Proposition 5.4.1 *Assume that I_{DY} is a weak Dolev-Yao interpretation and I is a fixed-point interpretation on system $\mathcal{S}$. Then, $I_{DY}(A\ infers, h) \subseteq I(A\ infers, h)$.*

Proof By induction on the derivation length to reach $M \in I_{DY}(A\ infers, h)$. Base case, $M \in messages(h|A)$: Assume that $h \sim^{\rho} h'$ then $\rho(M) \in messages(h'|A)$, i.e., $h' \models_{\langle \mathcal{S}, I \rangle} exists\,\rho(M)$. Since ρ and h' are arbitrary, $h \models_{\langle \mathcal{S}, I \rangle} \Box_A exists\, M$, i.e., $M \in I(A\ infers, h)$ since I is fixed point. Induction step: Assume that $M \in I_{DY}(A\ infers, h')$ is inferred in n steps. Case (1), the last inference step in the derivation to M was left-un-pairing, i.e., (2) in definition 5.1.1. Then, $M \cdot M' \in I_{DY}(A\ infers, h)$ is derived in less than n steps, for some M'. By the induction assumption, $M \cdot M' \in I(A\ infers, h)$, i.e., $h \models_{\langle \mathcal{S}, I \rangle} \Box_A exists\, M \cdot M'$. By theorem 4.3.4 (since $exists\, M \cdot M' \models exists\, M$), $h \models_{\langle \mathcal{S}, I \rangle} \Box_A exists\, M$, i.e, $M \in I(A\ infers, h)$. Case (2), the last derivation step to M was right-un-pairing: Similar to case (1). Case (3), the last derivation step to M was decryption: Then, $\{M\}_K \in I_{DY}(A\ infers, h)$ and $K \in I_{DY}(A\ infers, h)$ are derived in less than n steps, for some K. By the induction assumption, $\{M\}_K, K \in I(A\ infers, h)$, i.e., $h \models_{\langle \mathcal{S}, I \rangle} \Box_A exists\, \{M\}_K$ and $h \models_{\langle \mathcal{S}, I \rangle} A\ infers\, K$. By theorem 4.3.4, $h \models_{\langle \mathcal{S}, I \rangle} \Box_A exists\, M$, i.e., $M \in I(A\ infers, h)$. □

Corollary 5.4.2 *Assume that I_{DY} is a weak Dolev-Yao interpretation and I is an inductive interpretation on system $\mathcal{S}$. Then, $I_{DY}(A\ infers, h) \subseteq I(A\ infers, h)$.*

Proof From proposition 5.4.1. □

By propositions 5.2.1 and 5.3.6, the converse of proposition 5.4.1 fails, i.e., weak Dolev-Yao interpretations need not be inductive. However, weak Dolev-Yao and

inductive interpretations coincide on *atomic messages* in systems where atoms are interchangeable, as we show below. The following example illustrates why the agreement fails if atoms are not interchangeable.

Example 5.4.3 *Assume two distinct agents A and B. Fix an atom $K \in \mathcal{C}$, and define an initialization action i such that:*

$$\begin{aligned} i(A) &= \emptyset \\ i(B) &= \{K\} \end{aligned}$$

Let the set of histories be $H = \{i\}$. For a weak Dolev-Yao interpretation I on H, we have: $K \notin I(A\,infers, i)$. By contrast, for any fixed point interpretation I on H, we have: $K \in I(A\,infers, i)$, since:

$$i \models_{\mathcal{I}} \Box_A\, exists\, K \tag{5.15}$$

for any interpreted system $\mathcal{I}$ based on H. (5.15) follows from the fact that $i \sim_A^{\rho} i$ implies that $\rho(K) = K$.

Thus, an agreement between the weak Dolev-Yao and the inductive interpretation can only be obtained in systems where atoms are interchangeable. A set H of execution histories is *parametric*, if it is closed under any swapping of atoms from $(\mathcal{C} - \mathcal{A})$, i.e., if $h \in H$ then $[c_0 - c_1](h) \in H$ for any $c_1, c_1 \in (\mathcal{C} - \mathcal{A})$. (See section 4.1 for the notation $[c_0 - c_1]$.) An interpreted system is parametric if it is based on a parametric set H of histories.

Write $[c_0 - c_1/\kappa](M)$ for the result of swapping c_0 and c_1 in those parts of M which are hidden from κ:

- If $K \notin \kappa$ then $[c_0 - c_1/\kappa](\{M\}_K) = \{[c_0 - c_1](M)\}_{[c_0 - c_1](K)}$
- If $K \in \kappa$ then $[c_0 - c_1/\kappa](\{M\}_K) = \{[c_0 - c_1/\kappa](M)\}_{[c_0 - c_1/\kappa](K)}$
- $[c_0 - c_1/\kappa](M \cdot M') = [c_0 - c_1/\kappa](M) \cdot [c_0 - c_1/\kappa](M')$
- $[c_0 - c_1/\kappa](c) = c$, for $c \in \mathcal{C}$

Lemma 5.4.4 *If $c_0, c_1 \notin \kappa$, then $[c_0 - c_1/\kappa] \lhd \kappa$.*

Proof (1) $[c_0 - c_1/\kappa]$ is a message permutation: By induction on the structure of messages, we get $[c_0 - c_1/\kappa]([c_0 - c_1/\kappa](M)) = M$ if $c_0, c_1 \notin \kappa$. (2) $[c_0 - c_1/\kappa]$ satisfies conditions 1, 2 and 3 in definition 4.1.1: Immediate. □

Let $Keys(h) = Keys(messages(h))$ and $Keys(H) = \bigcup_{h \in H} Keys(h)$. (The set $Keys(\kappa)$, for $\kappa \subseteq \mathcal{T}$, is defined in section 4.3.)

Lemma 5.4.5 *Let I be a weak Dolev-Yao interpretation. Let $\kappa \cap Keys(h) \subseteq I(A\,infers, h)$. Assume that $c_0, c_1 \notin I(A\,infers, h)$. Then, $[c_0 - c_1/\kappa](h|A) = [c_0 - c_1](h|A)$.*

Proof Let $h|A = \{M_1, \cdots, M_m\} \cdot \sigma_1(M_{m+1}) \cdots \sigma_n(M_n)$. Then, for each $1 \leq i \leq n$, every occurrence of c_0 or c_1 in M_i is hidden from κ, since $\kappa \cap \mathit{Keys}(h) \subseteq I(A\ \mathit{infers}, h)$. Therefore, $[c_0 - c_1/\kappa](M_i) = [c_0 - c_1](M_i)$. □

Theorem 5.4.6 (Agreement on Atoms) *Assume a system $\mathcal{S}$ based on parametric H. Assume that $\mathit{Keys}(H) \subseteq (\mathcal{C} - \mathcal{A})$. Let I_{Ind} be an inductive interpretation on $\mathcal{S}$, and let I_{DY} be a weak Dolev-Yao interpretation on $\mathcal{S}$. For any $K \in (\mathcal{C} - \mathcal{A})$:*

$$K \in I_{Ind}(A\ \mathit{infers}, h) \Rightarrow K \in I_{DY}(A\ \mathit{infers}, h)$$

Proof By corollary 5.4.2, $K \in I_{DY}(A\ \mathit{infers}, h) \Rightarrow K \in I_{Ind}(A\ \mathit{infers}, h)$. For the converse, let I_i be the interpretation function at step i in the fixed point construction of the proof of theorem 5.3.3, such that $I_0(A\ \mathit{infers}, h) = \emptyset$, $I_{i+1} = f(I_i)$, and $I_\lambda(A\ \mathit{infers}, h) = \bigcup_{i<\lambda} I_i(A\ \mathit{infers}, h)$, if λ is a limit ordinal. We show that for each ordinal i and each $c_0 \in (\mathcal{C} - \mathcal{A})$:

$$c_0 \in I_i(A\ \mathit{infers}, h) \Rightarrow c_0 \in I_{DY}(A\ \mathit{infers}, h) \tag{5.16}$$

Base case: $I_0 = \emptyset$. Induction step, for successor ordinals: Assume that $c_0 \notin I_{DY}(A\ \mathit{infers}, h)$. Since $\mathit{messages}(h)$ is finite[3], so is $I(\mathit{exists}, h)$. Therefore, there is $c_1 \in (\mathcal{C} - \mathcal{A})$ such that $c_1 \notin I(\mathit{exists}, h)$. Let $h' = [c_0 - c_1](h)$. Since H is parametric, $h' \in H$. Since $I_{DY}(A\ \mathit{infers}, h) \subseteq I(\mathit{exists}, h)$, $c_1 \notin I_{DY}(A\ \mathit{infers}, h)$. By the induction assumption, $I_i(A\ \mathit{infers}, h) \cap \mathit{Keys}(h) \subseteq I_{DY}(A\ \mathit{infers}, h)$. Let $\rho = [c_0 - c_1/I_i(A\ \mathit{infers}, h)]$. By lemma 5.4.5,

$$\rho(h|A) = [c_0 - c_1](h|A) = h'|A \tag{5.17}$$

By lemma 5.4.4,

$$\rho \lhd I_i(A\ \mathit{infers}, h) \tag{5.18}$$

From (5.17) and (5.18),

$$h \sim_A^\rho h' \text{ in } \langle \mathcal{S}, I_i \rangle \tag{5.19}$$

By construction of h', $c_0 \notin I(\mathit{exists}, h')$, i.e.,

$$h' \not\models_{\langle \mathcal{S}, I_i \rangle} \mathit{exists}\, \rho(c_0) \tag{5.20}$$

since $\rho(c_0) = c_0$, as $c_0 \in \mathcal{A}$. From (5.19) and (5.20):

$$h \not\models_{\langle \mathcal{S}, I_i \rangle} \Box_A \mathit{exists}\, c_0$$

I.e., $c_0 \notin I_{i+1}(A\ \mathit{infers}\, h)$. The induction step for limit ordinals is immediate. □

The agreement on atomic messages (theorem 5.4.6) can be used to evaluate the epistemic modality in inductive systems. The following corollary illustrates this.

Corollary 5.4.7 (Crowds-Style Protocol) *Let $\mathcal{I}$ be the protocol implementation in section 2.3. If $\mathcal{I}$ is inductive, it satisfies specifications (2.3), (2.4), (2.5) and (2.6).*

Proof From lemma 4.2.8, theorem 5.4.6 and lemma 5.1.5. □

[3]An initialization action assigns a finite set of messages to each agent.

Chapter 6

Completeness for BAN-Like Theories

Ever since the inception of BAN logic [16], there has been much confusion concerning the semantics for its epistemic modality. While several semantics have been proposed, only soundness results – not completeness results – have been attempted. In this chapter, we use the semantics from previous sections to interpret BAN's modality on message passing systems, and we prove soundness, completeness and decidability for BAN-like logics. Completeness and decidability are generalized to logics induced by an arbitrary theory base. The theory base may express how participants in a specific protocol are expected to behave, or state general assumptions about the network.

6.1 Classical BAN Logic

BAN logic, named after Burrows, Abadi and Needham, is the first, and, perhaps, the most practically succesful, proof system combining epistemic logic and formal cryptography. BAN appeared in the late 80's, and soon spawned many extensions and variations (cf. [7, 9, 27, 39, 51, 52, 57, 74, 77, 78, 83]). In BAN-style analyses of a security protocol, the security goal – in most cases an authentication goal – is formulated as an epistemic logic statement. For instance:

$$\Box_A \, B \mathit{sent} M$$

$$\Box_B \, \Box_A B \, \mathit{sent} M$$

The authentication goal is then derived in the proof system, starting from more self-evident assumptions about what happens during protocol execution, such as what messages are sent, received or generated.

We introduce rules of original BAN logic [16] as requirements on theories. A *theory* is a set L of statements such that L contains all Boolean tautologies and

R1	$A \textit{ sees } \{\textit{from } B : M\}_K, \Box_A K \textit{ secret of } G \vdash \Box_A B \textit{ said } M,\ B \in G$
R2	$A \textit{ sees } M \cdot M' \vdash A \textit{ sees } M$
R3	$A \textit{ sees } M \cdot M' \vdash A \textit{ sees } M'$
R4	$A \textit{ sees } \{M\}_K,\ \Box_A K \textit{ secret of } G \vdash A \textit{ sees } M$
R5	$\Box_A B \textit{ said } M \cdot M' \vdash \Box_A B \textit{ said } M$
R6	$\Box_A B \textit{ said } M \cdot M' \vdash \Box_A B \textit{ said } M'$
R7	$\Box_A \textit{fresh } M \vdash \Box_A \textit{fresh } M \cdot M'$
R8	$\Box_A \textit{fresh } M' \vdash \Box_A \textit{fresh } M \cdot M'$
R9	$\Box_A \textit{fresh } M,\ \Box_A K \textit{ secret of } G \vdash \Box_A \textit{fresh } \{M\}_K$
R10	$\Box_A F \vdash F$

Table 6.1: Classical BAN

L is closed under modus ponens, i.e., if $F \to F' \in L$ and $F \in L$ then $F' \in L$. A statement F is derivable from a set Δ of statements in theory L, $\Delta \vdash_L F$, if there is a finite number of statements $F_1, ..., F_n \in \Delta$ such that $(\bigwedge_{1 \le i \le n} F_i) \to F \in L$. As usual, we write $\vdash_L F$ for $\emptyset \vdash_L F$, and we omit the subscript L whenever L is clear from the context.

Assume that $\mathcal{P}$ contains unary predicates $A \textit{ sees}$, $A \textit{ said}$, *fresh* and *secret of* G, for $A \in \mathcal{A}$ and $\emptyset \subset G \subseteq \mathcal{A}$. The intended meaning of each predicate is as follows: Agent A *sees* a message if A can infer that message from something A received, and A *said* a message if A can infer that message from something A sent. A message is *fresh* if it is not a sub-message of some message sent long ago. Finally, a message is a *secret* of a non-empty group G of agents if the message is known only to members of that group. Let $\textit{from } B : M$ abbreviate, say, $B \cdot M$.

Definition 6.1.1 (Classical BAN) *A theory is a classical BAN logic if it satisfies all conditions in table 6.1.*

Note that rule *R1*, the well-known *message meaning rule*, assumes that agents are honest, in the sense that the first component inside a cipher text, if locked with a secret key, is a reliable sender field.

In definition 6.1.1, we define a class of logics, rather than a single logic, since the original BAN logic is open ended and leaves out rules that are intuitively valid.

For instance, *seeing introspection*:

$$A \textit{ sees } M \vdash \Box_A A \textit{ sees } M \tag{6.1}$$

is not part of the original BAN logic, even though it is clearly implicit in requirement *R1*. As another illustration, all requirements may be generalized to iterated modalities. For instance, requirement *R2* may be generalized to $\Box_A \Box_B C \textit{ sees } M \cdot M' \vdash \Box_A \Box_B C \textit{ sees } M$.

While the definition 6.1.1 keeps close to the original definition of BAN logic in [16], it nonetheless simplifies the original definition. Firstly, original BAN has language constructs and proof rules for asymmetric cryptography. Secondly, the original BAN paper [16] reads the epistemic modality as "Agent A believes that", rather than as "Agent A knows that". As in [68, 75], we adopt the latter interpretation, adding the rule *R10*. The BAN predicate *jurisdiction* thereby becomes superfluous, and is removed. Thirdly, we drop the BAN predicate *good*, since it is analogous to *secret*. Finally, original BAN includes so called "idealized" messages, messages with logical statements inside, and a rule to the effect that agents only say (send) statements they know to be true. We refer to [78] for a comprehensive presentation of original BAN.

It is clear that BAN logic intends complex terms to refer *de re* ("directly") and the epistemic modality to reflect the extent to which cryptographic calculations are feasible. If either of these two assumptions are dropped, then rule *R9*, for instance, is unnecessarily weak: The premiss $\Box_A K \textit{ secret of } G$ could be removed. Note also that if terms are not intended to refer *de re*, but the modality is intended to reflect limited decryptability, rule *R1* would be intuitively invalid when M is an encryption $\{M'\}_{K'}$.

The semantics for the epistemic modality in BAN have long been a source of confusion. The AT semantics (section 3.3) and some instantiations (cf. [68, 75]) of classical multi-agent semantics (section 3.2) have been proposed for BAN's modality. Since reception introspection

$$A \textit{ received } M \vdash \Box_A A \textit{ received } M \tag{6.2}$$

is invalid in AT-style semantics (see example 3.3.5), seeing introspection (6.1) also fails, and consequently, rule *R1*. On the other hand, BAN logic is sound in the proposed classical multi-agent semantics. However, in section 3.1, we found that when complex terms refer *de re* and the modality reflects limited decryptability (as in BAN logic), Kripke semantics yields unintended validities, due to the logical omniscience problem. As a result, no BAN-like logic is complete with respect to classical multi-agent semantics, or any other Kripke semantics. Indeed, in the literature on BAN-like logics, the question of completeness is largely ignored.

6.2 BAN Theories

Since the definition 6.1.1 of classical BAN logics leaves out intuitively valid rules (as does original BAN logic itself), we should not expect completeness for an arbitrary

classical BAN logic; We need stronger proof rules.

In the remainder of chapter 6, we shall restrict messages to a finite message space, i.e., a finite, non-empty set of messages closed under $\geq$ (in other words, if $M \geq M'$ then the space contains M' if it contains M); A message M is, from now on, a message in the fixed message space. Also, from now on, we assume that $\mathcal{P}$ contains exactly the following unary predicates:

$$p ::= A \textit{ received} \mid A \textit{ rec} \mid A \textit{ sent} \mid A \textit{ sen} \mid A \textit{ infers} \mid \textit{unfresh} \mid \textit{exists}$$

Recall that $A \textit{ rec } M$ if M is a sub-message of some message A received, and analogously for $A \textit{ sen}$ and $A \textit{ sent}$. Write $\exists M' \geq M.F(M)$ for the *finite*[1] disjunction $\bigvee_{M' \geq M} F(M')$.

In a sense, the language isolates epistemic content to the epistemic modality: None of the primitive predicates involve the notion of "feasible cryptographic computation" – except, of course, the predicate $A \textit{ infers}$. But, this predicate will be eliminable in the theories we consider, and is kept for presentation purposes only. By contrast, predicates in original BAN (and its successors), for instance *sees*, *said* and *secret*, do depend on a model of "feasible decryptability". Instead, "epistemic" predicates from classical BAN (section 6.1) are introduced as abbreviations, similar to [68]:[2]

- $A \textit{ sees } M =_{df} \Box_A A \textit{ rec } M$
- $A \textit{ said } M =_{df} \Box_A A \textit{ sen } M$
- $M \textit{ secret of } G =_{df} (\bigvee_{A \in G} A \textit{ infers } M) \wedge (\bigwedge_{A \notin G} \neg A \textit{ infers } M)$

Definition 6.2.1 (BAN Theory) *A theory L is a BAN theory, if and only if, L contains the axioms and is closed under the rules in table 6.2.*

The *permutation necessitation rule PNec*, which weakens the standard rule of necessitation, formalizes the intuition that an agent knows all "feasibly computable" theorems. The rule *PNec* is quasi-semantic in that it uses the consistency relation $\lhd$. But, since there are finitely many permutations, rule *PNec* is finitary, i.e., involves a finite set of premises. When combined with axiom K, *PNec* yields a weakening of normality, according to which an agent knows "feasibly computable" logical implications of what the agent knows:

Lemma 6.2.2 (Permutation Normality) *Assume that L is a BAN theory and assume that $\rho(\Delta) \vdash_L \rho(F)$ for all $\rho \lhd \kappa$. Then, $A \textit{ infers } \kappa$, $\Box_A \Delta \vdash_L \Box_A F$.*

[1] The message space is finite.

[2] The similar abbreviations found in [68] use a syntactically defined modality (cf. section 1.9).

Weakening of S5	
PNec	$\dfrac{\rho(F),\ \ \forall \rho \lhd \kappa}{A\ infers\ \kappa \rightarrow \Box_A F}$
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	$\neg\Box_A F \rightarrow \Box_A \neg\Box_A F$
Introspection	
I	$p_A(M) \rightarrow \Box_A p_A(M),\ p_A \in \{A\ received,\ A\ sent\}$
Infers Reduction	
Red	$A\ infers\ K \leftrightarrow \Box_A\ exists\ K$
Global Clock	
GC	$unfresh\ M \rightarrow \exists M' \geq M.\ \bigvee_{A \in \mathcal{A}} (A\ sent\ M' \wedge \Box_A\ unfresh\ M')$
Monotonicity	
Mono	$p(M) \rightarrow p(M'),\ M \geq M',\ p \in \{exists, A\ rec, A\ sen, unfresh\}$
Predicates Mix	
M1	$A\ received\ M \rightarrow A\ rec\ M$
M2	$A\ sent\ M \rightarrow A\ sen\ M$
M3	$A\ received\ M \rightarrow exists\ M$
M4	$A\ sent\ M \rightarrow exists\ M$
M5	$A\ \ rec\ M \rightarrow \exists M' \geq M.A\ \ received\ \ M'$
M6	$A\ \ sen\ \ M \rightarrow \exists M' \geq M.A\ \ sent\ \ M'$
M7	$exists\ M \rightarrow \exists M' \geq M.\ \bigvee_{A \in \mathcal{A}} A\ \ infers\ \ M'$

Table 6.2: BAN Theory

Proof Assume that $\rho(\Delta) \vdash_L \rho(F)$, $\forall \rho \lhd \kappa$. Since the message space is finite, there are only finitely many permutations. Let $\rho_1, ..., \rho_n$ be all permutations ρ such that $\rho \lhd \kappa$. For each $i \in \{1, ..., n\}$ there is a finite $\Delta_i \subseteq \Delta$ such that $\rho_i(\Delta_i) \vdash_L \rho_i(F)$. Thus for each $i \in \{1, ..., n\}$: $\rho_i(\Delta_1, ..., \Delta_n) \vdash_L \rho_i(F)$. Since $\Delta_1, ..., \Delta_n$ is finite, by rule *PNec* and axiom K: A *infers* κ, $\Box_A(\Delta_1, ..., \Delta_n) \vdash_L \Box_A F$. Since $\Delta_i \subseteq \Delta$: A *infers* κ, $\Box_A \Delta \vdash_L \Box_A F$. □

As its proof shows, lemma 6.2.2 depends on the restriction to a finite message space.

Axioms K, T, *4* and *5* are standard for introspective knowledge. The introspection axiom I says that an agent knows if it sent or received a message. Axiom *Red* states that an agent infers a message precisely if the agent knows it exists. According to axiom *GC*, any unfresh message M is part of some message M' some agent A sent long ago. The axiom reflects the assumption that the time is, to some extent, common knowledge: If agent A sent message M' long ago, then agent A knows it sent M' long ago, and so knows that M' is unfresh. In a temporal logic extension, axiom *GC* would reduce to sending introspection (axiom I), general epistemic-temporal interaction axioms and non-epistemic axioms for predicates. The remaining axioms are non-epistemic and straightforward. Axiom *Mono* says that A *rec*, A *sen*, *exists* and *unfresh* are monotone with respect to the sub-message relation $\geq$.

At first sight, it might appear as if predicates $A\,rec$ and $A\,sen$ are superfluous: By axioms *Mono*, *M1*, *M2*, *M5* and *M6* it follows that every BAN theory contains:

$$A\ rec\ M \leftrightarrow \exists M' \geq M.A\ received\ M'$$
$$A\ sen\ M \leftrightarrow \exists M' \geq M.A\ sent\ M'$$

Nonetheless, the predicates are not eliminable. For instance, BAN theories need not contain any of the following:[3]

$$\Box_A \exists M' \geq M.A\ received\ M' \rightarrow \Box_A A\ rec\ M$$
$$\Box_A \exists M' \geq M.A\ sent\ M' \rightarrow \Box_A A\ sen\ M$$

(Recall that $\exists M' \geq M.F(M')$ is just an abbreviation of $\bigvee_{M' \geq M} F(M')$.)

6.3 Embedding of Classical BAN Logic

By way of the definitions in section 6.2 of classical BAN predicates *sees*, *said* and *secret*, as well as the obvious abbreviation $fresh\,M =_{df} \neg unfresh\,M$, the conditions of classical BAN can be derived using the following lemma.

Lemma 6.3.1 *Assume that L is a BAN theory. Assume that $\rho(\Delta) \vdash_L \rho(F)$ for all $\rho \lhd \{K\}$.*

[3] Soundness theorem 11.2.1 can be used to show this.

1. $\Box_A\, K\, secret\, of\, G,\ \Box_A \Delta \vdash_L \Box_A F$.

2. $A\, sees\, K\ \Box_A \Delta \vdash_L \Box_A F$.

Proof (1): From axiom *Red* and axiom *T*, $K\, secret\, of\, G \vdash exists\, K$, i.e., by lemma 6.2.2, $\Box_A K\, secret\, of\, G \vdash \Box_A exists\, K$, i.e., by axiom *Red*, $\Box_A K\, secret\, of\, G \vdash A\, infers\, K$. By assumption and lemma 6.2.2, we reach (1). (2): From axioms *Mono*, *M3* and *M5*, $A\, rec\, K \vdash exists\, K$, i.e., by lemma 6.2.2, $\Box_A A\, rec\, K \vdash \Box_A exists\, K$, i.e., by axiom *Red*, $A\, sees\, K \vdash A\, infers\, K$. By assumption and lemma 6.2.2, we reach (2). □

Theorem 6.3.2 *BAN theories satisfy classical BAN conditions R2 - R10.*

Proof From axiom *Mono* and lemma 6.3.1. □

In fact, through successive application of lemma 6.3.1, theorem 6.3.2 can be generalized to classical BAN conditions with iterated modalities. For instance, BAN theories satisfy the following generalization of condition *R9*:

$$\Box_A \Box_B fresh\, M,\ \Box_A \Box_B K\, secret\, of\, G \vdash \Box_A \Box_B fresh\, \{M\}_K$$

To obtain classical BAN condition *R1*, we add an origination axiom:

$$K\ secret\ of\ G \rightarrow A\ rec\ \{from\, B : M\}_K \rightarrow B\ said\ \{from\, B : M\}_K \wedge B\ sees\ K \tag{6.3}$$

Theorem 6.3.3 *Any BAN theory that contains the origination axiom (6.3) satisfies classical BAN condition* R1.

Proof From axiom *Mono*, $B\, sen\, \{from\, B : M\}_K \vdash B\, sen\, M$. By lemma 6.3.1.2, we obtain $\Box_B B\, sen\, \{from\, B : M\}_K,\ B\ sees\ K \vdash \Box_B B\, sen\, M$, i.e., $B\, said\, \{from\, B : M\}_K,\ B\ sees\ K \vdash B\, said\, M$. By lemma 6.3.1.1, we get $\Box_A B\, said\, \{from\, B : M\}_K,\ \Box_A B\ sees\ K,\ \Box_A K\ secret\ of\ G \vdash \Box_A B\, said\, M$. Condition *R1* follows by lemma 6.3.1.1 applied to (6.3). □

Of course, axiom (6.3) is only applicable to a group G of honest agents who supply sender fields inside their ciphertexts. But, a weaker form of origination axiom is more generally applicable:

$$K\ secret\ of\ G \rightarrow (A\ rec\ \{M\}_K \rightarrow \bigvee_{B \in G} (B\ said\ \{M\}_K \wedge B\ sees\ K) \tag{6.4}$$

Proposition 6.3.4 *Any BAN theory that contains the weaker origination axiom (6.4) satisfies the condition:*

- $A\, sees\, \{M\}_K,\ \Box_A K\, secret\, of\, G \vdash \Box_A \bigvee_{B \in G} B\, said\, M$

Proof From axiom *Mono*, $B\ sen\ \{M\}_K \vdash B\ sen\ M$. By lemma 6.3.1.2, we get $B\ said\ \{M\}_K, B\ sees\ K \vdash B\ said\ M$, i.e., we obtain $\bigvee_{B\in G}(B\ said\ \{M\}_K \wedge B\ sees\ K) \vdash \bigvee_{B\in G} B\ said\ M$. By lemma 6.3.1.1, $\Box_A K\ secret\ of\ G, \Box_A \bigvee_{B\in G}(B\ said\ \{M\}_K \wedge B\ sees\ K) \vdash \Box_A \bigvee_{B\in G} B\ said\ M$. The proposition follows by lemma 6.3.1.1 applied to (6.4). □

Theorems 6.3.2 and 6.3.3 and proposition 6.3.4 provide some justification to our definition of *sees* and *said*. The following proposition lends some further support.

Corollary 6.3.5 *Any BAN theory contains:*

1. $A\ sees\ M \rightarrow \Box_A A\ sees\ M$
2. $\neg A\ sees\ M \rightarrow \Box_A \neg A\ sees\ M$
3. $A\ said\ M \rightarrow \Box_A A\ said\ M$
4. $\neg A\ said\ M \rightarrow \Box_A \neg A\ said\ M$
5. $A\ received\ M \rightarrow A\ sees\ M$
6. $A\ sent\ M \rightarrow A\ said\ M$

Proof (1): Axiom *4*. (2): Axiom *5*. (3): Axiom *4*. (4): Axiom *5*. (5): From axiom *M1* and lemma 6.2.2, $\Box_A A\ received\ M \rightarrow \Box_A A\ rec\ M$, i.e., by axiom *I*, $A\ received\ M \rightarrow A\ sees\ M$. (6): From axiom *M2* and lemma 6.2.2, $\Box_A A\ sent\ M \rightarrow \Box_A A\ sen\ M$, i.e., by axiom *I*, $A\ sent\ M \rightarrow A\ said\ M$. □

6.4 Theory Base

Theorem 6.3.3 and proposition 6.3.4 suggest that we might be interested in BAN theories generated from a base of "extra axioms". In fact, BAN-style protocol analysis normally add protocol specific rules.[4]

Example 6.4.1 *Consider the Needham-Schröder Shared Key Protocol [65] between principals A and B and with key server S. If the server sends the cipher text* $\{N \cdot B \cdot K \cdot M\}_{K_A}$*, and* K_A *is A:s server key, then the server generated K for A and B:*

$$S\ said \{N \cdot B \cdot K \cdot M\}_{K_A}, K_A\ secret\ of \{A, S\}, fresh\ N \rightarrow K\ secret\ of \{A, B, S\} \quad (6.5)$$

Furthermore, agent A does not send the kind of cipher texts sent by the key server S:

$$K_A\ secret\ of \{A, S\} \rightarrow \neg A\ said \{N \cdot B \cdot K \cdot M\}_{K_A} \quad (6.6)$$

[4]Either explicitly (cf. [53, 77, 83]) or implicitly by substituting "idealized" messages for messages in the protocol description.

Assume a BAN theory that contains protocol specific axioms (6.5) and (6.6), for all keys N, K and K_a and all messages M, and contains the weaker origination axiom (6.4) for $G = \{A, S\}$. Then, the BAN theory also contains the following authentication specification:

$$A \text{ received } \{N \cdot B \cdot K \cdot \{K \cdot A\}_{K_B}\}_{K_A}, \Box_A K_A \text{ secret of } \{A, S\}, \Box_A \text{fresh } N \\ \rightarrow \Box_A K \text{ secret of } \{A, B, S\}$$

stating that if A sees the message $\{N \cdot B \cdot K \cdot \{K \cdot A\}_{K_B}\}_{K_A}$ from the server, knows the key K_A to this message, and knows that the nonce N inside is fresh, then A knows that the key K provided inside is secret between A, B and S. The derivation proceeds as follows. From (6.6), K_A secret of $\{A, S\}$, $\bigvee_{A' \in \{A,S\}} A' \text{ said } \{N \cdot B \cdot K \cdot M\}_{K_A} \vdash S \text{ said } \{N \cdot B \cdot K \cdot M\}_{K_A}$. By lemma 6.3.1,

$$\Box_A K_A \text{ secret of } \{A, S\}, \Box_A \bigvee_{A' \in \{A,S\}} A' \text{ said } \{N \cdot B \cdot K \cdot M\}_{K_A} \\ \vdash \Box_A S \text{ said } \{N \cdot B \cdot K \cdot M\}_{K_A} \tag{6.7}$$

From weak origination axiom (6.4) and lemma 6.3.1, we get $\Box_A K_A$ secret of $\{A, S\}$, A sees $\{N \cdot B \cdot K \cdot M\}_{K_A} \vdash \Box_A \bigvee_{A' \in \{A,S\}} A' \text{ said } \{N \cdot B \cdot K \cdot M\}_{K_A}$. By corollary 6.3.5.5,

$$\Box_A K_A \text{ secret of } \{A, S\}, A \text{ received } \{N \cdot B \cdot K \cdot M\}_{K_A} \\ \vdash \Box_A \bigvee_{A' \in \{A,S\}} A' \text{ said } \{N \cdot B \cdot K \cdot M\}_{K_A} \tag{6.8}$$

Combining (6.7) and (6.8),

$$\Box_A K_A \text{ secret of } \{A, S\}, A \text{ received } \{N \cdot B \cdot K \cdot M\}_{K_A} \vdash \Box_A S \text{ said } \{N \cdot B \cdot K \cdot M\}_{K_A}$$

The specification follows from this and the application of lemma 6.3.1 on (6.5).

We define the BAN theory induced by a finite set $\boldsymbol{A}$ of statements, in symbols $L\boldsymbol{A}$, as the smallest BAN theory containing the finite set $\boldsymbol{A}$; We shall refer to $\boldsymbol{A}$ as the theory base of $L\boldsymbol{A}$. Note that the origination schemata (6.3) and (6.4), as well as the protocol specific axiom schemata in example 6.4.1, are finite, since the message space is finite.

6.5 Extended Message Passing Systems

Next, we define the extended message passing systems with respect to which we obtain soundness and completeness for BAN theories. The action vocabulary of message passing systems (example 2.2.1) is extended with an action `begin epoch` that signals the start of a new time period ("epoch"); This action will be used

to interpret the predicate *unfresh*, along the lines of [7]. To aid the completeness construction, we also add a set of internal ("silent") actions; These will be used to enforce message correlations in the canonical countermodel to non-theorems.

The set Π now contains the actions:

$$A\,\texttt{sends},\ A\,\texttt{receives},\ A\,int,\ \texttt{begin epoch}$$

for $A \in \mathcal{A}$, and where *int* ranges over a finite set of primitive internal actions, and `begin epoch` is an action that takes no message argument. Agents observe the "global time" and their own communication actions and internal actions:

$$\Pi|A = \{A\,\texttt{sends},\ A\,\texttt{receives},\ A\,int,\ \texttt{begin epoch} \mid \text{Internal action } int\}$$

The observation function $|$ is lifted to execution histories as usual. In details, we have:

$$\begin{aligned}
i|A &= \texttt{init}\,i(A)\\
(h \cdot A\,\texttt{sends}\,M)|A &= (h|A) \cdot A\,\texttt{sends}\,M\\
(h \cdot B\,\texttt{sends}\,M)|A &= (h|A),\ B \neq A\\
(h \cdot A\,\texttt{receives}\,M)|A &= (h|A) \cdot A\,\texttt{receives}\,M\\
(h \cdot B\,\texttt{receives}\,M)|A &= (h|A),\ B \neq A\\
(h \cdot A\,int\,M)|A &= (h|A) \cdot A\,int\,M\\
(h \cdot B\,int\,M)|A &= (h|A),\ B \neq A\\
(h \cdot \texttt{begin epoch})|A &= (h|A) \cdot \texttt{begin epoch}
\end{aligned}$$

Throughout the rest of chapter 6, the action vocabulary Π and the observation function $|$ are fixed according to the above definitions. Thus, each system $S = \langle \Pi, H, | \rangle$ can be identified with its underlying set H of histories.

The interpretation of predicates is also fixed. To begin with, the predicates *A received*, *A sent*, *A rec* and *A sen* are interpreted as before:

$$\begin{aligned}
I(A\,sent, h) &= \{M \mid (A\,\texttt{sends}\,M) \in actions(h)\}\\
I(A\,received, h) &= \{M \mid (A\,\texttt{receives}\,M) \in actions(h)\}\\
I(A\,rec, h) &= \{M \mid \exists M' \geq M.\,(A\,\texttt{receives}\,M') \in actions(h)\}\\
I(A\,sen, h) &= \{M \mid \exists M' \geq M.\,(A\,\texttt{sends}\,M') \in actions(h)\}
\end{aligned}$$

The predicate *unfresh* is interpreted along the lines of [7], through the `begin epoch` action: $I(unfresh, h)$ contains sub-messages of messages sent prior to the latest epoch: $M \in I(unfresh, h)$, if and only if,

$$h = \theta_- \cdot \texttt{begin epoch} \cdot \theta_+ \text{ and } M \in I(A\,sen, \theta_-)$$

for some $A \in \mathcal{A}$ and some action traces θ_- and θ_+. The interpretation of *unfresh* is not critical: Other accounts can be dealt with by routine changes.

The interpretation of predicate *exists* is now modified so that internal actions do not effect what messages exist. Write $purge(h)$ for the result of removing all internal actions from history h:

$$\begin{aligned} purge(i) &= i \\ purge(h \cdot A\,int\,M) &= purge(h) \\ purge(h \cdot \sigma) &= purge(h) \cdot \sigma, \text{ if } \sigma \text{ is not of the form } A\,int\,M \end{aligned}$$

A message is now said to exist if it is a sub-message in the purged history:

$$I(exists, h) = \{M \mid \exists M' \geq M.M' \in messages(purge(h))\}$$

Admittedly, the new interpretation of *exists* makes requirement (5.4) for inductive interpretations of predicate A *infers* slightly ad hoc. Even so, we assume that I is inductive. Trivially, the proofs of theorems 5.3.3 and 5.3.5 carry over to the new interpretation of *exists*.

Under the fixed interpretation of predicates, each system H determines a unique interpreted system $\langle H, I\rangle$, and we say that statement F is valid in H, in symbols $\models_H F$, if $\models_{\langle H,I\rangle} F$ for the unique permitted interpretation I on H.

6.6 Soundness, Completeness and Decidability

Write $\|\Delta\|$ for the set of all systems H validating all statements in Δ. The set Δ is sound with respect to a class C of systems, if $C \subseteq \|\Delta\|$. The set Δ is complete with respect to C, if Δ contains all statements valid in all systems in C.

Theorem 6.6.1 (Soundness) *$L\boldsymbol{A}$ is sound with respect to $\|\boldsymbol{A}\|$.*

Proof Boolean tautologies and modus ponens: Routine. *PNec*: Lemma 4.3.2. *K* and *T*: Proposition 4.2.4. *4* and *5*: Corollary 5.3.5 (which remains valid for the modified exists predicate). *I*: Proposition 4.2.6. *Red*: Induction property (5.4). *GC*: Since $\texttt{begin}\,\texttt{epoch} \in \Pi|A$ and $A\,\texttt{sends} \in \Pi|A$. Non-epistemic axioms: Routine. □

Theorem 6.6.2 (Completeness) *$L\boldsymbol{A}$ is complete with respect to $\|\boldsymbol{A}\|$.*

Proof Section 6.7. □

Thus, the protocol base $\boldsymbol{A}$ semantically guarantees a specification only if the specification is a theorem of $L\boldsymbol{A}$. Contrast this with the usual verification practice in BAN, based on an open ended proof system: If the specification is unprovable, it can be concluded that either the protocol assumptions do not ensure the specification or the base logic needs to be extended (cf. [16, 78]).

Completeness theorem 11.2.2 is evidence that our notion of validity is faithful to BAN. In fact, since the protocol base is freely chosen, the theorem suggests not only

that validity with respect to all systems is faithful to BAN, but also that validity with respect to selected classes of systems is faithful. Clearly, applications such as model checking require the latter and stronger form of faithfulness.

Theorem 6.6.3 (Decidability) *$L\boldsymbol{A}$ is decidable.*

Proof Section 6.7. □

6.7 Completeness Construction

We shall reach completeness and decidability by way of a finite model property: If F is not a theorem of $L\boldsymbol{A}$, then there is a finite system $H_F \in \|A\|$ such that H_F invalidates F. To construct the countersystem H_F, we first lift the semantics from systems to a more general class of structures, counterpart models (section 6.7). We then build a canonical counterpart model $\mathcal{C}_{L\boldsymbol{A}}$ that validates precisely the theorems of $L\boldsymbol{A}$ (section 6.7). Finally, we transform $\mathcal{C}_{L\boldsymbol{A}}$, while preserving validity of $\mathbf{A}$ and non-validity of F, into a finite system H_F (section 6.7).

Counterpart Model

We abstract from our semantics on systems to a semantics on abstract counterpart models [56]. A counterpart model is a triple $\mathcal{C} = \langle W, \longrightarrow, Int\rangle$, where W is a set of worlds (states), $\longrightarrow_A^\rho \subseteq W \times W$ for each agent $A \in \mathcal{A}$ and each message permutation ρ, and $Int(p, w)$ is a set of messages, intuitively the set of messages satisfying predicate p at w.[5] Intuitively, $w \longrightarrow_A^\rho w'$ says that any M at w, could, for all A knows, be $\rho(M)$ at w'. The semantics of section 4.2 is generalized in the obvious way:

$$\begin{array}{lcl} w \models_{\mathcal{C}} \Box_A F & \Leftrightarrow & \forall \rho : \forall w' \in W : w \longrightarrow_A^\rho w' \Rightarrow w' \models_{\mathcal{C}} \rho(F) \\ w \models_{\mathcal{C}} p(M) & \Leftrightarrow & M \in Int(p, w) \end{array}$$

Truth conditions for boolean operators are standard.

Counterpart models are used in counterpart semantics [56], a semantics for first order modal logic. However, in counterpart semantics, one renames the assignment to variables as one moves along the possibility relation from one state to another, rather than, as we do here, rename the evaluated statement F. We return to counterpart semantics in part II (cf. section 11.3).

Canonical Counterpart Model

Next, we build a canonical counterpart model that validates precisely the theorems of a given BAN theory. Assume a BAN theory L. A set Δ of statements is consistent

[5] In this chapter, all predicates are unary.

if there is no statement $\neg F$ such that $\Delta \vdash \neg F$ and $\Delta \vdash F$.[6] Δ is maximal consistent if there is no consistent set Δ' such that $\Delta' \supset \Delta$. Using the standard Lindenbaum construction we obtain:

Lemma 6.7.1 (Extension Lemma) *If $\Delta \nvdash F$, there is a maximal consistent set $\Delta' \supseteq \Delta$ such that $F \notin \Delta'$.*

The canonical counterpart model for BAN theory L is $\mathcal{C}_L = \langle W_L, \underset{L}{\longrightarrow}, Int_L \rangle$, where

- W_L is the set of all maximal consistent sets
- $Int_L(w, p) = \{M \mid p(M) \in w\}$
- $w \underset{L}{\longrightarrow}{}^{\rho}_{A} w' \Leftrightarrow \rho \triangleleft Int_L(A\ infers, w)$ and $\forall F : \Box_A F \in w \Rightarrow \rho(F) \in w'$

Lemma 6.7.2 (Truth lemma) $w \models_{\mathcal{C}_L} F \Leftrightarrow F \in w$.

Proof By induction in (the number of statement operators in) F, using permutation normality (Lemma 6.2.2). The base case, for atomic F, is immediate. The induction step, for boolean operators, uses standard properties of maximal consistent sets. For the epistemic modality let $w|A$ be the set $\{F \mid \Box_A F \in w\}$. For the only-if direction first:

$\Box_A F \notin w$		
$\Rightarrow \rho(w\|a) \nvdash \rho(F)$ & $\rho \triangleleft Int_L(a\ uses, w)$, $\exists \rho$	(By permutation normality)	(6.9)
$\Rightarrow \rho(w\|a) \subseteq w'$ & $\rho(F) \notin w'$, $\exists w' \in W_L$	(By lemma 6.7.1)	(6.10)
$\Rightarrow w' \not\models_{\mathcal{C}_L} \rho(F)$	(By the ind. hyp.)	(6.11)
$\Rightarrow \forall F : \Box_A F \in w \Rightarrow \rho(F) \in w'$	(By (6.10))	(6.12)
$\Rightarrow w \underset{L}{\longrightarrow}{}^{\rho}_{A} w'$	(By (6.9) and (6.12))	(6.13)
$\Rightarrow w \not\models_{\mathcal{C}_L} \Box_A F$	(By (6.11) and (6.13))	

For the if-direction:

$\Box_A F \in w$ & $w \underset{L}{\longrightarrow}{}^{\rho}_{A} w'$ & $w' \in W_L$		
$\Rightarrow \rho(F) \in w'$		
$\Rightarrow w' \models_{\mathcal{C}_L} \rho(F)$	(By the ind. ass.)	
$\Rightarrow w \models_{\mathcal{C}_L} \Box_A F$	(By the assumptions)	

□

The canonical counterpart model validates precisely all theorems.

Corollary 6.7.3 (Canonical Model Corollary) $\models_{\mathcal{C}_L} F \Leftrightarrow \vdash_L F$.

[6] Since the BAN theory L is clear from the context, we drop the subscripted L from $\vdash_L$.

Proof From extension lemma 6.7.1 and truth lemma 6.7.2. □

If w is related to w' under permutation ρ, then ρ transforms what the agent knows in w to what the agent knows in w'.

Lemma 6.7.4 *If* $w \underset{L}{\longrightarrow}^{\rho}_{A} w'$, *then* $\Box_A F \in w \Leftrightarrow \Box_A \rho(F) \in w'$.

Proof From axioms 4 and 5. Assume that $w \underset{L}{\longrightarrow}^{\rho}_{A} w'$.

$\Box_A F \in w$
$\Rightarrow \Box_A \Box_A F \in w$ (Axiom 4)
$\Rightarrow \Box_A \rho(F) \in w'$ (Since $w \underset{L}{\longrightarrow}^{\rho}_{A} w'$)

For the converse:

$\Box_A F \notin w$
$\Rightarrow \neg \Box_A F \in w$
$\Rightarrow \Box_A \neg \Box_A F \in w$ (Axiom 5)
$\Rightarrow \neg \Box_A \rho(F) \in w'$ (Since $w \underset{L}{\longrightarrow}^{\rho}_{A} w'$)
$\Rightarrow \Box_A \rho(F) \notin w'$

□

Filtration

In the section following this one, we transform the canonical model into a system, while preserving validity of theorems and non-validity of a given non-theorem F. In this section, we lay down conditions that assure that such a transformation succeeds: We define a notion of filtration from a counterpart model to an interpreted system, such that the filtration preserves truth values in a set Γ of statements.

Assume a set Γ of statements, a counterpart model $\mathcal{C} = \langle W, \longrightarrow, Int \rangle$ and an interpreted system $\mathcal{I} = \langle H, I \rangle$. A relation $\rightsquigarrow \subseteq W \times H$ is a Γ-filtration from $\mathcal{C}$ to $\mathcal{I}$ if whenever $w \rightsquigarrow h$ then

1. $Int(p, w) = I(p, h)$
2. $w \longrightarrow^{\rho}_{A} w' \Rightarrow \exists h' \in H : w' \rightsquigarrow h',\ h \sim^{\rho}_{A} h'$
3. $h \sim^{\rho}_{A} h' \Rightarrow \exists w' \in W : w' \rightsquigarrow h',\ w \models_{\mathcal{C}} \Box_A F \Rightarrow w' \models_{\mathcal{C}} \rho(F),$ if $\Box_A F \in \Gamma$

From now on, we assume that Γ is closed in two respects: Γ is closed under sub-statements, i.e., if $F \in \Gamma$ and F' is a sub-statement of F then $F' \in \Gamma$, and Γ is closed under message permutations, i.e., if $F \in \Gamma$ and ρ is any permutation of messages then $\rho(F) \in \Gamma$.[7]

[7] Since the message space is finite, there are finitely many permutations.

Lemma 6.7.5 (Filtration Lemma) *Assume that $\rightsquigarrow$ is a Γ-filtration from $\mathcal{C}$ to $\mathcal{I}$, $w \rightsquigarrow h$ and $F \in \Gamma$. Then, $w \models_{\mathcal{C}} F \Leftrightarrow h \models_{\mathcal{I}} F$.*

Proof By induction on F. The base case, for atomic F, is filtration condition (i). The induction step, for boolean operators, is immediate. The induction step, for the epistemic modality: Assume, first, $h \models_{\mathcal{I}} \Box_A F$.

$w \longrightarrow_A^{\rho} w'$
- $\Rightarrow w' \rightsquigarrow h' \wedge h \sim_A^{\rho} h', \exists h' \in H$ (Filt.cond. (ii))
- $\Rightarrow h' \models_{\mathcal{I}} \rho(F)$ (Since $h \models_{\mathcal{I}} \Box_A F$)
- $\Rightarrow w' \models_{\mathcal{C}} \rho(F)$ (Induct. assum., Γ is closed)
- $\Rightarrow w \models_{\mathcal{C}} \Box_A F$ (w' and r are arbitrary)

For the converse, assume that $w \models_{\mathcal{C}} \Box_A F$.

$h \sim_A^{\rho} h'$
- $\Rightarrow w' \rightsquigarrow h' \wedge (w \models_{\mathcal{C}} \Box_A F \Rightarrow w' \models_{\mathcal{C}} \rho(F)), \exists w'$ (Filt.cond. (iii))
- $\Rightarrow w' \models_{\mathcal{C}} \rho(F)$ (Since $w \models_{\mathcal{C}} \Box_A F$)
- $\Rightarrow h' \models_{\mathcal{I}} r(F)$ (Induct. assum., Γ is closed)
- $\Rightarrow h \models_{\mathcal{I}} \Box_A F$ (h' and ρ are arbitrary)

□

Canonical System

We build a filtration from the canonical counterpart model $\mathcal{C}_L = \langle W_L, \underset{L}{\longrightarrow}, Int_L \rangle$ into an interpreted system, transforming each maximal consistent set w into one or more histories h. To this end, we first transform an arbitrary set Δ of statements into two actions sets, a set $Actions^-(\Delta)$ of "old" actions and a set $Actions^+(\Delta)$ of "recent" actions:

$$Actions^-(\Delta) = \bigcup_{A \in \mathcal{A}} Actions^-(\Delta, A)$$
$$Actions^+(\Delta) = \bigcup_{A \in \mathcal{A}} Actions^+(\Delta, A)$$

where $Actions^-(\Delta, A)$ is the set:

1 $\{A \,\mathsf{sends}\, M : (A\, sent\, M) \in \Delta \wedge (\Box_A\, unfresh\, M) \in \Delta\}$

and $Actions^+(\Delta, A)$ is the union of three sets:

2 $\{A \,\mathsf{receives}\, M : (A\, received\, M) \in \Delta\}$

3 $\{A \,\mathsf{sends}\, M : (A\, sent\, M) \in \Delta \wedge (\Box_A\, unfresh\, M) \notin \Delta\}$

4 $\{A \,\mathsf{int}\, F : (\Box_A F) \in \Delta\}$

In (4), we assume that internal actions are of the form $a\,\mathtt{int}\,F$, where F is any statement.[8]

Assume a set Γ of statements. We relate a state w in the canonical counterpart model to a history h, in symbols $w \rightsquigarrow h$, if and only if, for some initialization i and some action traces θ^- and θ^+:

- $h = i \cdot \theta^- \cdot \mathsf{begin\ epoch} \cdot \theta^+$
- $i(A) = \{M \mid (A\ \mathit{infers}\ M) \in w \cap \Gamma\}$, $A \in \mathcal{A}$
- $\mathit{Actions}(\theta^-) = \mathit{Actions}^-(w \cap \Gamma)$
- $\mathit{Actions}(\theta^+) = \mathit{Actions}^+(w \cap \Gamma)$

In order to obtain a finite system, we exclude any history that repeats actions, i.e., contains at least two occurrences of the same action $\pi(M)$. Thus, we define the canonical system – the system that we filter the canonical counterpart model into – as the set H_L of all repetition-free histories obtained from states in W_L:

$$H_L = \{h : \exists w \in W_L \text{ s.t. } w \rightsquigarrow h \text{ and } h \text{ is repetition-free}\}$$

Let the canonical interpretation I_L interpret predicates $A\ \mathit{sent}$, $A\ \mathit{received}$, $A\ \mathit{rec}$, $A\ \mathit{sen}$ and exists according to the requirements in section 6.5:

$$I_L(A\ \mathit{sent}, h) \quad = \quad \{M \mid (A\ \mathtt{sends}\ M) \in \mathit{actions}(h)\},$$

and so on for the other predicates. For the remaining predicate, $A\ \mathit{infers}$, let:

$$I_L(A\ \mathit{infers}, i \cdot \theta) \quad = \quad i(A)$$

Finally, set the canonical interpreted system to $\mathcal{I}_L = \langle H_L, I_L \rangle$. We proceed to show that $\rightsquigarrow$ is a Γ-filtration from $\mathcal{C}_L$ to $\mathcal{I}_L$, under certain assumptions on Γ: We assume, from now on, that Γ is finite and contains all atomic statements and contains $\Box_A A\ \mathit{received}\ M$, $\Box_A A\ \mathit{sent}\ M$, $\Box_A A\ \mathit{infers}\ M$, $\Box_A \mathit{exists}\ M$ and $\Box_A \mathit{unfresh}\ M$ for all $A \in \mathcal{A}$ and messages M.[9] As before, we also assume that Γ is closed under sub-statements and message permutations ρ.

Lemma 6.7.6 (Filtration Condition 1) *If $w \rightsquigarrow h$, then $\mathit{Int}_L(p, w) = I_L(p, h)$.*

Proof Assume that $w \rightsquigarrow h$. Case $p = A\ \mathit{received}$:

$M \in \mathit{Int}_L(A\ \mathit{received}, w)$
$\Leftrightarrow$ $A\ \mathit{received}\ M \in w$
$\Leftrightarrow$ $A\ \mathit{received}\ M \in w \cap \Gamma$ (Since Γ contains atomic statements)
$\Leftrightarrow$ $A\,\mathsf{receives}\,M \in \mathit{Actions}^+(w \cap \Gamma)$
$\Leftrightarrow$ $A\,\mathsf{receives}\,M \in \mathit{Actions}(h)$ (Since $w \rightsquigarrow h$)
$\Leftrightarrow$ $M \in I_L(A\ \mathit{received}, h)$

[8] This assumes a slightly different definition of internal action than that of Section 6.5. Alternatively, we could introduce $a\,\mathtt{int}\,F$ as an abbreviation for an internal action of the form $a\ \mathit{int}\ M$.

[9] Recall that the message space is finite.

Case $p = A\,rec$:

$M \in Int_L(A\,rec, w)$
$\Leftrightarrow$ $A\ rec\,M \in w$
$\Leftrightarrow$ $A\ received\,M' \in w$, $\exists M' \geq M$ (By *M1*, *M5*, *Mono*)
$\Leftrightarrow$ $M' \in Int_L(A\,received, w)$, $\exists M' \geq M$
$\Leftrightarrow$ $M' \in I_L(A\,received, h)$, $\exists M' \geq M$ (By case $p = A\,received$)
$\Leftrightarrow$ $M \in I_L(A\,rec, h)$

Cases $p = A\,sent$ and $p = A\,sen$ are analogous. Case $p = unfresh$:

$M \in Int_L(unfresh, w)$
$\Leftrightarrow$ $unfresh\,M \in w$
$\Leftrightarrow$ $A\ sent\,M', \Box_A unfresh\,M' \in w$, $\exists A.\exists M' \geq M$ (By *GC*, *T*, *Mono*)
$\Leftrightarrow$ $A\ sent\,M', \Box_A unfresh\,M' \in w \cap \Gamma$, $\exists A.\exists M' \geq M$ (By conditions on Γ)
$\Leftrightarrow$ $A\,\mathsf{sends}\,M' \in Actions^-(w \cap \Gamma)$, $\exists A.\exists M' \geq M$
$\Leftrightarrow$ $M \in I_L(unfresh, h)$ (Since $w \rightsquigarrow h$)

Case $p = exists$: Let $Actions(\Delta) = Actions^-(\Delta) \cup Actions^+(\Delta)$.

$M \in Int_L(exists, w)$
$\Leftrightarrow$ $exists\,M \in w$
$\Leftrightarrow$ $(A\ sent\,M' \vee A\ received\,M' \vee A\,infers\,M') \in w$, $\exists A.\exists M' \geq M$ (By *M7*, *M3*, *M4*, *Mono*, and *Red*, *T*)
$\Leftrightarrow$ $A\ sent\,M' \in w \cap \Gamma$
or $A\ received\,M' \in w \cap \Gamma$
or $A\,infers\,M' \in w \cap \Gamma$, $\exists A.\exists M' \geq M$ (By conditions on Γ)
$\Leftrightarrow$ $A\,\mathsf{sends}\,M' \in Actions(w \cap \Gamma)$
or $A\,\mathsf{receives}\,M' \in Actions(w \cap \Gamma)$
or $A\,infers\,M' \in w \cap \Gamma$, $\exists A.\exists M' \geq M$
$\Leftrightarrow$ $M' \in messages(purge(h))$, $\exists M' \geq M$ (Since $w \rightsquigarrow h$)
$\Leftrightarrow$ $M \in I_L(exists, h)$

Case $p = A\,infers$:

$M \in Int_L(A\ infers, w)$
$\Leftrightarrow$ $A\ infers\,M \in w$
$\Leftrightarrow$ $A\ infers\,M \in w \cap \Gamma$ (By conditions on Γ)
$\Leftrightarrow$ $M \in I_L(A\ infers, h)$ (Since $w \rightsquigarrow h$)

□

Lemma 6.7.7 (Filtration Condition 2) *If $w \rightsquigarrow h$ and $w \underset{L}{\longrightarrow}{}^{\rho}_{A} w_1$, there is $h_1 \in H_L$ such that $w_1 \rightsquigarrow h_1$ and $h \sim^{\rho}_{A} h_1$ in $\mathcal{I}_L$.*

Proof Assume that $w \rightsquigarrow h$ and $w \underset{L}{\longrightarrow}{}^{\rho}_{A} w_1$. From the latter assumption, $\rho \lhd Int_L(A\ infers, w)$, i.e., by lemma 6.7.6 and the first assumption, $\rho \lhd I_L(A\ infers, h)$. Pick some $h_1 \in H_L$ such that $w_1 \rightsquigarrow h_1$. Let:

$$\begin{aligned} h|A &= (\mathtt{init}\,\kappa) \cdot \theta^- \cdot \mathsf{begin\ epoch} \cdot \theta^+ \\ h_1|A &= (\mathtt{init}\,\kappa_1) \cdot \theta_1^- \cdot \mathsf{begin\ epoch} \cdot \theta_1^+ \end{aligned}$$

for some action traces θ^-, θ^+, θ_1^-, θ_1^+ and sets $\kappa, \kappa_1 \subseteq \mathcal{T}$. We shall show that:

$$\rho(\kappa) = \kappa_1 \tag{6.14}$$
$$\rho(Actions(\theta^-)) = Actions(\theta_1^-) \tag{6.15}$$
$$\rho(Actions(\theta^+)) = Actions(\theta_1^+) \tag{6.16}$$

The lemma then follows by shuffling the inside of θ_1^- and the inside of θ_1^+: After shuffling, we obtain $\rho(h|A) = h_1|A$, and so $h \sim^{\rho}_{A} h_1$, but still $h_1 \in H_L$ and $w_1 \rightsquigarrow h_1$. For (6.14):

$M \in \kappa$
- $\Leftrightarrow$ $(A\ infers\ M) \in w \cap \Gamma$ (Since $w \rightsquigarrow h$)
- $\Leftrightarrow$ $(A\ infers\ M) \in w$ (By conditions on Γ)
- $\Leftrightarrow$ $(\Box_A\ exists\ M) \in w$ (By *Red*)
- $\Leftrightarrow$ $(\Box_A\ exists\ \rho(M)) \in w_1$ (By lemma 6.7.4, $w \underset{L}{\longrightarrow}{}^{\rho}_{A} w_1$)
- $\Leftrightarrow$ $(A\ infers\ \rho(M)) \in w_1$ (By *Red*)
- $\Leftrightarrow$ $(A\ infers\ \rho(M)) \in w_1 \cap \Gamma$ (By conditions on Γ)
- $\Leftrightarrow$ $\rho(M) \in \kappa_1$ (Since $w_1 \rightsquigarrow h_1$)

For (6.15):

$A\ \mathsf{sends}\ M \in Actions(\theta^-)$
- $\Leftrightarrow$ $A\ \mathsf{sends}\ M \in Actions^-(w \cap \Gamma)$ (Since $w \rightsquigarrow h$)
- $\Leftrightarrow$ $(A\ sent\ M) \in w \cap \Gamma$ and $(\Box_A\ unfresh\ M) \in w \cap \Gamma$
- $\Leftrightarrow$ $(\Box_A A\ sent\ M) \in w \cap \Gamma$ and $(\Box_A\ unfresh\ M) \in w \cap \Gamma$ (By I, T, conditions on Γ)
- $\Leftrightarrow$ $(\Box_A A\ sent\ \rho(M)) \in w_1 \cap \Gamma$ and $(\Box_A\ unfresh\ \rho(M)) \in w_1 \cap \Gamma$ (By lemma 6.7.4, $w \underset{L}{\longrightarrow}{}^{\rho}_{A} w_1$)
- $\Leftrightarrow$ $(A\ sent\ \rho(M)) \in w_1 \cap \Gamma$ and $(\Box_A\ unfresh\ \rho(M)) \in w_1 \cap \Gamma$ (By I, T)
- $\Leftrightarrow$ $A\ \mathsf{sends}\ \rho(M) \in Actions^-(w_1 \cap \Gamma)$
- $\Leftrightarrow$ $A\ \mathsf{sends}\ \rho(M) \in Actions(\theta_1^-)$ (Since $w_1 \rightsquigarrow h_1$)

To establish (6.16), we show that A receives $M \in \mathit{Actions}(\theta^+)$ iff A receives $\rho(M) \in \mathit{Actions}(\theta_1^+)$, and similarly for internal and send actions. For receive actions:

A receives $M \in \mathit{Actions}(\theta^+)$

$\Leftrightarrow$ A receives $M \in \mathit{Actions}^+(w \cap \Gamma)$ (Since $w \rightsquigarrow h$)

$\Leftrightarrow$ A *received* $M \in w \cap \Gamma$

$\Leftrightarrow$ $\Box_A A$ *received* $M \in w \cap \Gamma$ (By I, T, conditions on Γ)

$\Leftrightarrow$ $\Box_A A$ *received* $\rho(M) \in w_1 \cap \Gamma$ (By lemma 6.7.4 and $w \underset{L}{\longrightarrow}{}^{\rho}_{A} w_1$ and conditions on Γ)

$\Leftrightarrow$ A *received* $\rho(M) \in w_1 \cap \Gamma$ (By I, T, conditions on Γ)

$\Leftrightarrow$ A receives $\rho(M) \in \mathit{Actions}^+(w_1 \cap \Gamma)$

$\Leftrightarrow$ A receives $\rho(M) \in \mathit{Actions}(h_1)$ (Since $w_1 \rightsquigarrow h_1$)

$\Leftrightarrow$ A receives $\rho(M) \in \mathit{Actions}(\theta_1^+)$

The proof for internal actions is similar and left to the reader. For send actions:

A sends $M \in \mathit{Actions}(\theta^+)$

$\Leftrightarrow$ A sends $M \in \mathit{Actions}^+(w \cap \Gamma)$ (Since $w \rightsquigarrow h$)

$\Leftrightarrow$ $(A\ \mathit{sent}\ M) \in w \cap \Gamma$
and $(\Box_A \mathit{unfresh}\, M) \notin w \cap \Gamma$

$\Leftrightarrow$ $(\Box_A A\ \mathit{sent}\ M) \in w \cap \Gamma$
and $(\Box_A \mathit{unfresh}\, M) \notin w \cap \Gamma$ (By I, T, conditions on Γ)

$\Leftrightarrow$ $(\Box_A A\ \mathit{sent}\ \rho(M)) \in w_1 \cap \Gamma$
and $(\Box_A \mathit{unfresh}\, \rho(M)) \notin w_1 \cap \Gamma$ (By lemma 6.7.4, $w \underset{L}{\longrightarrow}{}^{\rho}_{A} w_1$)

$\Leftrightarrow$ $(A\ \mathit{sent}\ \rho(M)) \in w_1 \cap \Gamma$
and $(\Box_A \mathit{unfresh}\, \rho(M)) \notin w_1 \cap \Gamma$ (By I, T)

$\Leftrightarrow$ A sends $\rho(M) \in \mathit{Actions}^+(w_1 \cap \Gamma)$

$\Leftrightarrow$ A sends $\rho(M) \in \mathit{Actions}(\theta_1^+)$ (Since $w_1 \rightsquigarrow h_1$)

□

Lemma 6.7.8 (Filtration Condition 3) *Assume that $h \sim^{\rho}_{A} h'$ in $\mathcal{I}_L$ and $w \rightsquigarrow h$. Then, there is $w' \in W_L$ such that $w' \rightsquigarrow h'$, and for all $(\Box_A F) \in \Gamma$: $w \models_{\mathcal{C}_L} \Box_A F \Rightarrow w' \models_{\mathcal{C}_L} \rho(F)$.*

Proof Assume that $w \rightsquigarrow h$ and $h \sim^{\rho}_{A} h'$ in $\mathcal{I}_L$. Then, $h' \in H_L$, i.e., $w' \rightsquigarrow h'$ for some $w' \in W_L$. Assume $(\Box_A F) \in \Gamma$. Then,

$w \models_{\mathcal{C}_L} \Box_A F$

$\Rightarrow$ $(\Box_A F) \in w \cap \Gamma$ (By lemma 6.7.2)

$\Rightarrow$ A int $F \in \mathit{Actions}^+(w \cap \Gamma)$

$\Rightarrow$ A int $F \in \mathit{Actions}(h)$ (By $w \rightsquigarrow h$)

$\Rightarrow A\,\mathsf{int}\,\rho(F) \in Actions(h')$ (By $h \sim^{\rho}_{A} h'$)
$\Rightarrow A\,\mathsf{int}\,\rho(F) \in Actions^{+}(w' \cap \Gamma)$ (By $w' \rightsquigarrow h'$)
$\Rightarrow \Box_A \rho(F) \in w'$
$\Rightarrow \rho(F) \in w'$ (By axiom T)
$\Rightarrow w' \models_{\mathcal{C}_L} \rho(F)$ (By lemma 6.7.2)

□

Having thus established all three filtration conditions, we know that $\rightsquigarrow$ is a filtration.

Corollary 6.7.9 *$\rightsquigarrow$ is a Γ-filtration from the canonical counterpart model to the canonical interpreted system.*

Proof From lemmas 6.7.6, 6.7.7 and 6.7.8. □

To reach completeness theorem 11.2.2, it remains to be shown that $\mathcal{I}_L$ is inductive.

Lemma 6.7.10 *The canonical interpreted system is inductive.*

Proof We show first that the canonical interpretation function I_L is fixed point. Assume that $w \rightsquigarrow h$.

$h \models_{\mathcal{I}_L} A\,infers\,K$
$\Leftrightarrow A infers\,K \in w$ (Lemma 6.7.6)
$\Leftrightarrow \Box_A exists\,K \in w$ (Axiom *Red*)
$\Leftrightarrow h \models_{\mathcal{I}_L} \Box_A exists\,K$ (Lemmas 6.7.2 + 6.7.5, corollary 6.7.9 and $\Box_A exists\,K \in \Gamma$)

To show that I_L is minimal among fixed point variants, we show that if $K \in I_L(A\,infers, h)$ then $h \models_{\langle H_L, I'\rangle} \Box_A exists\ K$ for any variant I' of I_L.

$K \in I_L(A\,infers, h)$ and $h \sim^{\rho}_{A} h'$ in $\langle H_L, I'\rangle$ (6.17)
$\Rightarrow K \in messages(purge(h|A))$ (From (6.17)) (6.18)
$\Rightarrow \rho(h|A) = h'|A$ (From (6.17)) (6.19)
$\Rightarrow \rho(K) \in messages(purge(h'|A))$ (From (6.18) + (6.19))
$\Rightarrow \rho(K) \in I'(exists, h')$
$\Rightarrow h' \models_{\langle H_L, I'\rangle} exists\,\rho(K)$
$\Rightarrow h \models_{\langle H_L, I'\rangle} \Box_A exists K$ (Since h' and ρ are arbitrary)

□

Lemma 6.7.11 (Finite Model Property) *If $\not\vdash_{L\boldsymbol{A}} F$, there is a finite system $H \in \|\boldsymbol{A}\|$ such that $\not\models_H F$.*

Proof From canonical model corollary 6.7.3, filtration lemma 6.7.5, lemma 6.7.9 and lemma 6.7.10. Assume that $\not\vdash_{L\boldsymbol{A}} F$. From canonical model corollary 6.7.3, $\not\models_{\mathcal{C}_{L\boldsymbol{A}}} F$ and $\models_{\mathcal{C}_{L\boldsymbol{A}}} \boldsymbol{A}$. Let Γ be the smallest set closed under message permutations and sub-statements, and containing F and $\boldsymbol{A}$, and containing all atomic statements, containing $\Box_A A$ *received* M, $\Box_A A$ *sent* M, $\Box_A$*exists* M, $\Box_A A$ *infers* M and $\Box_A$*unfresh*$\,M$, for all $A \in \mathcal{A}$ and messages M. Γ is finite, since $\boldsymbol{A}$ is finite. By filtration lemma 6.7.5, lemma 6.7.9 and lemma 6.7.10, $\not\models_{H_{L\boldsymbol{A}}} F$ and $\models_{H_{L\boldsymbol{A}}} \boldsymbol{A}$, where $H_{L\boldsymbol{A}}$ is the canonical system for theory $L\boldsymbol{A}$ and filtration set Γ. By construction, $H_{L\boldsymbol{A}}$ is finite, as Γ is finite. □

From Finite Model Property 6.7.11, we immediately get completeness theorem 11.2.2. By soundness and the proof of completeness it is not difficult to find a bound n such that $F \in L\boldsymbol{A}$, if and only if, F is valid in all systems in $\|A\|$ with at most n histories, each of size less than n. This is sufficient to establish Decidability Theorem 6.6.3.

Those are my principles, and If you don't like them ... well, I have others.

Groucho Marx

Part II

First-Order Epistemic Logic and Feasibly Computable Functions

Chapter 7

Relativized Static Equivalence

In this chapter, cryptography is modeled using private constants and arbitrary feasibly computable operations, as in the Applied Pi-calculus [32]. A relativized indistinguishability relation based on static equivalence [32] is introduced.

7.1 Static Equivalence

In process algebra based analysis of security protocols, security goals – often confidentiality goals – are defined in terms of an observational equivalence of programs (cf. [4, 32, 71]): A program successfully hides a condition if varying the condition has no observable effect. For example, an electronic voting protocol ensures voter anonymity if, approximately, reshuffling the votes among the voters preserves observational equivalence.

For security protocols that rely on one-way functions, the choice of observational equivalence is a delicate matter. Intuitively, not every difference in observable behavior makes a difference to what the external observer is able to infer, at least not with feasible computational resources. For instance, even if two instances of the electronic voting protocol output different encryptions on a public channel, the two instances might be indistinguishable to an observer, as long as the observer cannot (with feasible resources) decrypt the output.

Static equivalence [32] has recently emerged as a natural starting point for observational equivalences with respect to formal cryptography. In this section, we define a variant of static equivalence. Let f range over a countable set Σ of public, feasibly computable operators, each equipped with an arity. Let $A, B, ...$ range over a finite, non-empty set $\mathcal{A} \subseteq \Sigma$ of 0-arity operators, representing public names of distinct agents. Let c range over a countably infinite set SEC of secret constants, and $x, y, z...$ range over a countably infinite set VAR of variables. Message terms t are:

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

where f has arity n. Write $VAR(t)$ for the set of variables in t. Let $M, K, N, ...$ 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 the notation, we write M for the equivalence class $[M]_{\equiv}$, and f for its induced operation on classes.

Example 7.1.1 *To model pairing and asymmetric encryption, we assume the least congruence over ground terms satisfying the following equations:*

$$fst(pair(M, M')) \equiv M \tag{7.1}$$
$$snd(pair(M, M')) \equiv M' \tag{7.2}$$
$$dec(enc(M, pk(K)), K) \equiv M$$

Informally, fst/snd picks out first/second components, pk produces a matching encryption key and enc/dec encrypts/decrypts the first argument using the second as key. To model pairing and random asymmetric encryption, we assume the least congruence over ground terms satisfying (7.1) and (7.2), in addition to the following equation:

$$dec(enc(M, pk(K), N), K) \equiv M$$

The encryption operation enc now takes a third argument, N, as a random seed.

Throughout this part of the thesis, 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 function.*

Assume a non-empty, countable set LOC of store locations l. 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 7.1.2 *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 term, s-terms:

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

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

Corollary 7.1.3 *Inferable*$(s) = \{s(\alpha) : \alpha \in s\text{-terms}\}$.

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

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

Definition 7.1.4 *States s and s' are statically equivalent, written $s \approx s'$, if and only if, dom(s) = dom(s') and:*

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

Intuitively, for any s-terms α and α', the equality $\alpha = \alpha'$ represents an experiment available at state s; Two states are equivalent if expermients yield the same result at both states.

Relating the above definitions to [32], constants c corresponds to private/fresh names, states s correspond to frames, *Inferable*(s) corresponds to messages deduction ($\vdash$) from the frame s, and $s \approx s'$ is static equivalence between (finite) frames s and s'.

7.2 Indistinguishability under Permutation

Static equivalence $\approx$ is based on the the intuition that the message $s(\alpha)$ at s corresponds to the message $s'(\alpha)$ at s', in the sense that both messages are reached through the same computation α:

- $s(l)$ at s corresponds to $s'(l)$ at $'$.
- $enc(s(l_1), s(l_2))$ at s corresponds to $enc(s'(l_1), s'(l_2))$ at s'.
- $enc(pair(s(l_1), s(l_2)), s(l_3))$ at s corresponds to $enc(pair(s'(l_1), s'(l_2)), s'(l_3))$ at s'.

  ⋮

In this section, we reformulate static equivalence in terms of message correspondences. The reformulation, which makes message correspondences an explicit part of the equivalence, is reminiscent of framed bisimulation [3].

We define an indistinguishability $\sim$ between states which is relativized to a permutation on $\mathcal{T}_{\equiv}$. Informally, if $s \sim^{\rho} s'$, then s is statically equivalent to s' and 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 7.2.1 *$s \sim^{\rho} s'$, if and only if, dom(s) = dom(s') and:*

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

Lemma 7.2.2 *If $s \sim^{\rho} s'$ then $\rho(\mathit{Inferable}(s)) = \mathit{Inferable}(s')$.*

Proof By induction on inference length. □

Proposition 7.2.3 *The following hold:*

1. $s \sim^{Id} s$
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 7.2.2. (3) From lemma 7.2.2. □

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

Corollary 7.2.4 *Assume a permutation π on $\overline{\mathit{Inferable}(s)}$. Extend π to a permutation ρ on $\mathcal{T}_{\equiv}$ such that $\rho(M) = M$ for $M \in \mathit{Inferable}(s)$. Then, $s \sim^{\rho} s$.*

A state s is *normal* if s has countably infinite many non-inferred messages, i.e., $\overline{\mathit{Inferable}(s)}$ is countably infinite. This corresponds to the assumption in [32] that there always are fresh private names available. In the remainder of this section, we assume states are normal.

Lemma 7.2.5 *$s \sim^{\rho} s'$ if, and only if, $\mathit{dom}(s) = \mathit{dom}(s')$ and $\rho(s(\alpha)) = s'(\alpha)$ for all s-terms α.*

Proof From corollary 7.1.3. □

We reach the following permutation-based characterization of static equivalence.

Theorem 7.2.6 (Permutation-Based Characterization) *$s \approx s'$, if and only if, there is a permutation ρ such that $s \sim^{\rho} s'$.*

Proof Assume that $s \approx s'$. Define ρ by: (i) $\rho(s(\alpha)) = s'(\alpha)$, for all s-terms α, and (ii) $\rho(M_i) = N_i$ where $M_1, M_2, ...$ is an enumeration (without repetitions) of $\overline{\mathit{Inferable}(s)}$ and $N_1, N_2, ...$ is an enumeration (without repetitions) of $\overline{\mathit{Inferable}(s')}$. By corollary 7.1.3, $s \sim^{\rho} s'$. The converse is immediate from lemma 7.2.5. □

Chapter 8

Generalized First-Order Kripke Semantics

In this chapter, we generalize first-order Kripke semantics by updating the assignment (of data to logical variables) as we move from a state to an indistinguishable state. The update to the assignment is determined by the relativized indistinguishability of section 7.2.

8.1 Systems and Statements

Multi-Agent System We instantiate the multi-agent system framework [31, 66] 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 observation function $|$ assigns a set $LOC|A \subseteq LOC$ of locations observed (accessed) by agent A. The observation function 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 observation function $|$.

Example 8.1.1 *We model a system where either agent A or agent B posts a message, but agent C cannot observe whom. Assume the message congruence for pairing and asymmetric encryption from example 7.1.1. Assume two locations: $LOC = \{sender, post\}$. The state space is $S = \{s : LOC \to \mathcal{T}_{\equiv} \mid s(sender) \in \{A, B\}\}$. Agent C observes only the post location: $LOC|C = \{post\}$. The system is $\mathcal{S} = \langle LOC, S, | \rangle$.*

Inference and indistinguishability naturally relativize to an agent A:

- $Inferable(A, s) =_{df} Inferable(s|A)$
- $s \sim^{\rho}_{A} s'$, if and only if, $s|A \sim^{\rho} s'|A$.

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

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

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. Informally, the statement $\forall m.F[m/x]$ expresses the countably infinite conjunction:

$$\bigwedge_{M \in \mathcal{T}} F[M/x]$$

For instance, the statement

$$\forall m.(A \, \textit{received} \, m \rightarrow \Box_A A \, \textit{received} \, m) \tag{8.1}$$

informally expresses the conjunction:

$$\bigwedge_{M \in \mathcal{T}} (A \, \textit{received} \, M \rightarrow \Box_A A \, \textit{received} \, M)$$

The distinction between variables x and place holders m reflects the *de re*/*de dicto* dichotomy (cf. section 1.10): Variables $x \in VAR$ refer *de re*, while closed terms $M \in \mathcal{T}$ refer *de dicto*. For instance, statement (8.1) expresses knowledge *de dicto* and is intuitively invalid; Agent A need not know the structure of messages received, i.e., A need not know what terms are applicable to the messages received. By contrast, the the statement:

$$\forall x.(A \, \textit{received} \, x \rightarrow \Box_A A \, \textit{received} \, x)$$

expresses knowledge *de re* and is intuitively valid: Every value agent A receives is known by A to be received (cf. proposition 4.2.6). To highlight their respective use, we refer to the $\forall x$-quantifier and the $\forall m$-quantifier as, respectively, the *de re* quantifier and the *de dicto* quantifier.

Although we believe that the use of the *de dicto* quantifier is of independent interest, its motivation here is mainly technical. To obtain a complete axiomatization, we need an axiom stating that each variable x refers to some message M. Using the *de dicto* quantifier, we can express this grounding by the statement $\exists m.x = m$. In section 10.2, we show that the *de dicto* quantifier cannot be reduced to the *de re* quantifier.

Interpreted System A predicate interpretation I on a system $\mathcal{S}$ assigns, to each predicate p and state $s \in S$, a relation $I(p, s)$ in $\mathcal{T}_{\equiv}$ (matching the arity of p). An interpreted system based on a system $\mathcal{S} = \langle LOC, S, | \rangle$ is a structure $\mathcal{I} = \langle LOC, S, |, I \rangle$ where I is an interpretation on $\mathcal{S}$.

In some examples and propositions, we explicitly introduce the special unary predicates $A \, \textit{infers}$ and $@_l$, for $A \in \mathcal{A}$ and $l \in LOC$. When we do so, we implicitly require that $I(A \, \textit{infers}, s) = \textit{Inferable}(A, s)$ and $I(@_l, s) = \{s(l)\}$.

8.2 Counterpart Semantics Based on Static Equivalence

In this section, we interpret the epistemic modality through a counterpart semantics [56] based on the relativized indistinguishability of section 7.2: An agent knows a statement if the statement holds with respect to *corresponding* assignments at indistinguishable states. Assume an interpreted system $\mathcal{I}$, and an assignment $V : VAR \longrightarrow \mathcal{T}_{\equiv}$. Assignments are extended homomorphically to terms in the usual way, and $V[x \mapsto M]$ is V except that x is assigned M.

Definition 8.2.1 (Truth)

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

For Boolean operators we assume standard truth conditions. 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 8.2.2 *As an illustration of the difference between the de re quantifier and the quantifier, we have:*

$$\begin{array}{rl} \models & \forall x.(@_l\, x \rightarrow \Box_A @_l\, x),\ l \in \Pi|A \\ \not\models & \forall m.(@_l\, m \rightarrow \Box_A @_l\, m),\ l \in \Pi|A \end{array}$$

Example 8.2.3 *Consider the interpreted system $\mathcal{I}$ from example 8.1.1. Since $sender \notin LOC|C$, agent C does not know the sender:*

$$\models_{\mathcal{I}} \forall x.(@_{sender}\, x \rightarrow \neg\Box_C @_{sender}\, x)$$

However, since $post \in LOC|C$, agent C knows (as "bitstring") what message is posted:

$$\models_{\mathcal{I}} \forall x.(@_{post}\, x \rightarrow \Box_C @_{post}\, x) \tag{8.2}$$

On the other hand, C need not know the structure of the posted message:

$$\not\models_{\mathcal{I}} \forall m.(@_{post}\, m \rightarrow \Box_C @_{post}\, m) \tag{8.3}$$

The truth condition for the epistemic modality follows counterpart semantics in that it checks F at s' with respect to the corresponding assignment $\rho \circ V$ instead of the original assignment V. By contrast, in basic Kripke semantics (cf. [14]), the assignment is unchanged by the move from state s to state s':

$$s, V \models \Box_A F \Leftrightarrow \forall s' : s \sim_A s' \Rightarrow s', V \models F \tag{8.4}$$

The departure from basic Kripke semantics should not be over-stressed. The relativized indistinguishability relation $\sim_A^\rho$ induces a two-dimensional indistinguishability relation between evaluation points, i.e., pairs s, V:

Definition 8.2.4 *$s, V \sim_A s', V'$, if and only if, there is a permutation ρ such that $s \sim_A^\rho s'$ and $V' = \rho \circ V$.*

Corollary 8.2.5 (Two-Dimensional Reformulation)

$$s, V \models_{\mathcal{I}} \Box_A F \Leftrightarrow \forall s' \in S : \forall V' : s, V \sim_A s', V' \Rightarrow s', V' \models_{\mathcal{I}} F$$

Thus, the semantics might be described as a "two-dimensional" generalization of basic Kripke semantics. As expected, the two-dimensional $\sim_A$ is an equivalence.

Corollary 8.2.6 *The two-dimensional $\sim_A$ is an equivalence on evaluation points:*

- $s, V \sim_A s, V$
- *If $s, V \sim_A s', V'$ and $s', V' \sim_A s'', V''$ then $s, V \sim_A s'', V''$*
- *If $s, V \sim_A s', V'$ then $s', V' \sim_A s, V$*

Proof From proposition 7.2.3. □

In turn, the two-dimensional $\sim_A$ can be reformulated using static equivalence and s-terms, without quantifying over permutations ρ. In the following proposition, assume that local states $s|A$ and $s'|A$ are normal.

Proposition 8.2.7 *$s, V \sim_A s', V'$, if and only if,*

1. $s|A \approx s'|A$
2. $V(x) = s|A(\alpha) \Leftrightarrow V'(x) = s'|A(\alpha)$, *$s|A$-terms* α
3. $V(x) = V(y) \Leftrightarrow V'(x) = V'(y)$

Proof Assume $s, V \sim_A s', V'$. (1) By theorem 7.2.6. (2) $\Rightarrow$: By lemma 7.2.5. $\Leftarrow$: By proposition 7.2.3.3 and lemma 7.2.5. (3) Immediate. Conversely, assume (1), (2) and (3). By (1) and lemma 7.2.6, $s \sim_A^\rho s'$ for some ρ. By (2) and corollary 7.1.3 and lemma 7.2.5, $V'(x) = \rho \circ V(x)$ for all $V(x) \in \mathit{Inferable}(A, s)$. By (3), there is permutation π on $\mathit{Inferable}(A, s)$ such that $V'(x) = \rho \circ \pi \circ V(x)$ for all $V(x) \notin \mathit{Infers}(A, s)$.[1] Thus, $V' = \rho \circ \pi \circ V$. But, $s \sim_A^{\rho \circ \pi} s'$, since $s \sim_A^\rho s'$. □

[1] π is extended to a permutation on $\mathcal{T}_{\equiv}$ in the expected way: $\pi(M) = M$ if $M \in \mathit{Inferable}(A, s)$.

8.3 Interaction Between Knowledge and Cryptography

The use of counterpart semantics (definition 8.2.1) provides a handle on the notorious issue of mathematical omniscience in epistemic logic. Under basic Kripke semantics (truth condition (8.4)), agents know all the cryptographic equalities, agents are *cryptographically omniscient*:

$$t = t' \rightarrow \Box_A\, t = t' \tag{8.5}$$

For example,

$$x = enc(y, z) \rightarrow \Box_A x = enc(y, z)$$

Validity of cryptographic omniscience (8.5) follows under truth condition (8.4) from the fact that term equalities depend only on the assignment, i.e.,

$$s, V \models_I t = t' \implies s', V \models_I t = t'$$

Under cryptographic omniscience (8.5), knowledge of an equality does not reflect that the equality is feasible to compute. Instead, the epistemic modality is vacuous on cryptographic equations. In fact, all counterexamples to logical omniscience (cf. sections 1.8 and 3.1) – for languages with *de re* reference of closed terms M – translate directly into counter examples to cryptographic omniscience – for languages with *de re* reference of variables x and *de dicto* reference of closed terms M (cf. section 9.2).

By contrast, cryptographic omniscience (8.5) fails in our counterpart semantics (definition 8.2.1), 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 $(\rho \circ V)(M) = M$.

Example 8.3.1 *Continuing example 8.2.3, from (8.2) and (8.3), we obtain:*

$$\not\models_{\mathcal{I}} x = M \rightarrow \Box_C x = M$$

i.e., cryptographic omniscience (8.5) fails in $\mathcal{I}$

Even though cryptographic omniscience is invalid, a weaker form still holds.

Corollary 8.3.2 *The following schema is valid:*

$$y = f(\overline{x}) \rightarrow A\ infers\ \overline{x} \rightarrow \Box_A y = f(\overline{x}) \tag{8.6}$$

According to (8.6), an agent knows, at least, feasible computable relationships between inferred messages. By contrast, we obtain that an agent knows almost nothing about non-inferred messages. More precisely, we obtain that the agent knows a property of some non-inferred values $\overline{x}$ only if this property holds for any non-inferred values $\overline{z}$ with the same pattern of identities.

Corollary 8.3.3 *The following schema is valid:*[2]

$$\Box_A F(\overline{x}) \rightarrow \neg A\, infers\, \overline{x}, \overline{z} \rightarrow \bigwedge_{i,j}(x_i = x_j \leftrightarrow z_i = z_j) \rightarrow F[\overline{z}/\overline{x}] \tag{8.7}$$

Proof From corollary 7.2.4. □

According to corollary 8.3.3, if agent A cannot infer any of the messages $\overline{x}$ nor any of the messages $\overline{z}$, and $\overline{x}$ and $\overline{z}$ have the same pattern of identities, then $\Box_A F(\overline{x}) \rightarrow F[\overline{z}/\overline{x}]$.

[2] *Recall that $F(\overline{x})$ signifies that the list $\overline{x}$ consists precisely of all variables free in F.*

Chapter 9

Security Protocol Examples

In this chapter, we illustrate the language on various security protocols. All logical specifications considered depend on the absence of cryptographic omniscience.

9.1 Mix

Consider a mix in the style of [17], which decrypts and shuffles a sequence of random asymmetric ciphertexts (cf. example 7.1.1). There is a decryption key K_{mix} which is only known to the mix, while the corresponding encryption key $pk(K_{mix})$ is publicly known. The mix inputs a sequence of encryptions:

$$enc(M_1, pk(K_{mix}), N_1), \ldots, enc(M_l, pk(K_{mix}), N_l)$$

with some arbitrary content $M_1, \ldots, M_l$ and arbitrary random seeds $N_1, \ldots, N_l$. The mix outputs the encryption content in random order:

$$M_{\pi(1)}, \ldots, M_{\pi(l)}$$

for some random permutation π on $\{1, ..., l\}$.

The goal of the mix is that an eavesdropping spy should be unable to infer which input contains which output:

$$mix\ inputs\, x \wedge mix\ outputs\, y \rightarrow \neg \Box_{spy} x\ contains\, y \tag{9.1}$$

$$mix\ inputs\, x \wedge mix\ outputs\, y \rightarrow \Diamond_{spy} x\ contains\, y \tag{9.2}$$

where

$$x\ contains\, y =_{def} \exists z. \exists z'. x = enc(y, z, z')$$

It might be that our primary interest is that the mix detects, rather than prevents, information leakage, i.e., whenever the spy determines a link, the mix knows this:

$$mix\ inputs\, x \wedge mix\ outputs\, y \rightarrow \Box_{spy} x\ contains\, y \rightarrow \Box_{mix} \Box_{spy} x\ contains\, y \tag{9.3}$$

We check the above security goals in an interpreted system implementing the protocol. Let $\equiv$ be the congruence for pairing and random asymmetric encryption in example 7.1.1. Write $M_1 \cdots M_n$ for the list $pair(M_1, pair(M_2, \cdots))$. Fix an integer $l > 2$ as the size of message buffer of the mix. For any $k \in SEC$ as the private decryption key of the mix, we assume an input-output relation $InOut_k$ which relates any input list:

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

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:

$$m_{\pi(1)} \cdots m_{\pi(l)}$$

where π is an arbitrary permutation on $\{1, ..., l\}$. 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:

$$S_1 = \{s : LOC \to \mathcal{T}_{\equiv} \mid \langle s(in), s(out)\rangle \in InOut_{s(priv)} \wedge s(pub) = pk(s(priv))\}$$

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 $\mathcal{S}_1 = \langle LOC, S_1, |\rangle$. We assume location predicates $@_l$, for $l \in LOC$, and introduce some abbreviations:

$$\begin{aligned} mix\ inputs\, x &=_{def} \exists y_1 ... \exists y_l . @_{in}\,(y_1 \cdots y_l) \wedge \bigvee_i x = y_i \\ mix\ outputs\, x &=_{def} \exists y_1 ... \exists y_l . @_{out}\,(y_1 \cdots y_l) \wedge \bigvee_i x = y_i \end{aligned}$$

Proposition 9.1.1 *$\mathcal{S}_1$ satisfies (9.3), but neither (9.1) nor (9.2).*

Proof (i): $\mathcal{S}_1$ satisfies (9.3): Since $s|spy \subseteq s|mix$, $\models_{\mathcal{S}_1} \Box_{spy} F \to \Box_{mixF}$. But, $\models \Box_{spy} F \to \Box_{spy}\Box_{spy} F$ by proposition 7.2.3.2. (ii) $\mathcal{S}_1$ does not satisfy (9.1): Pick a system state $s \in S_1$ were all inputs are identical: $s(in) = M \cdots M$ and $s(out) = N \cdots N$ for some messages M and N. Pick any $s' \in S_1$ and a permutation ρ such that $s \sim^{\rho}_{spy} s'$. Since $\rho \circ s|spy = s'|spy$, we have $s'(in) = \rho \circ s(in) = \rho(M \cdots M) =$ (By the equations for pairing and since ρ is a homomorphism from inferred messages) $= \rho(M) \cdots \rho(M)$. Similary, $s'(out) = \rho(N) \cdots \rho(N)$. But, since $s' \in S_1$, each output in state s' is part of some input, i.e., $s', V[x := \rho(M), y := \rho(N)] \models x\ contains\, y$. I.e., $s', \rho \circ V[x := M, y := N] \models x\ contains\, y$. Since s' and ρ were chosen at random, $s, V[x := M, y := N] \models \Box_{spy} x\ contains\, y$. Thus, (9.1) fails in $\mathcal{S}_1$. (iii): $\mathcal{S}_1$ does not satisfy (9.2): Pick a system state $s \in S_1$ were exactly two inputs are identical. (9.2) fails at s, since $l > 2$. □

We modify the implementation so that the mix checks for replays and the spy performs size-comparisons. To achieve the latter, we add a length-computing operator len, and equations:

$$\begin{aligned} len(enc(M,K,N)) &\equiv len(M) \\ len(c) &\equiv len(c'),\ c =_{len} c' \end{aligned}$$

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 $InOut_k$:

$$enc(m_i, pk(k), n_i) \neq enc(m_j, pk(k), n_j),\ i \neq j \tag{9.4}$$

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 9.1.2 *$\mathcal{S}_2$ satisfies (9.3), but neither (9.1) nor (9.2).*

Proof (i): $\mathcal{S}_2$ satisfies (9.3): See proof of proposition 9.1.1. (ii) $\mathcal{S}_2$ does not satisfy (9.1): Pick $s \in S_2$ and assignment V such that $s(in) = V(x_1)\cdots V(x_l)$ and $s(out) = V(y_1)\cdots V(y_l)$ and $len(V(x_i)) \neq len(V(x_j))$ for all $i \neq j$. Assume $s, V \models_{\mathcal{S}_2} x_i\ contains\ y_j$. By equations for length, $len(V(x_i)) = len(V(y_j))$. Assume $s \sim^{\rho}_{spy} s'$. 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 $s', \rho \circ V \models_{\mathcal{S}_2} x_i\ contains\ y_j$. Since s' and ρ are aritrary, $s, V \models_{\mathcal{S}_2} \Box_{spy} x_i\ contains\ y_j$. (iii) $\mathcal{S}_2$ does not satisfy (9.2): Shown similarly to (ii). □

In both $\mathcal{S}_1$ and $\mathcal{S}_2$, the spy determines what is inside 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 (and, in the case of $\mathcal{S}_2$, by performing size-comparisons). In particular, the spy knows that at every possible state, each output is part of some input.

9.2 Crowds

We recall the Crowds-style protocol from section 2.3, 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 the symmetric encryption $enc(pair(B, M), K)$ to some random crowd member C_1, who in turn sends the ciphertext to B or to a random forwarder C_2, and so on until the ciphertext reaches B. In addition to crowd members, there are local spies. Each spy observes and controls the traffic in some part of the network.

Receiver anonymity means that a spy cannot tell the intended destination of a given message x:

$$x \textit{ is for } A \rightarrow \neg\Box_{spy} x \textit{ is for } A$$

$$x \textit{ is for } A \rightarrow \Diamond_{spy} x \textit{ is for } B$$

for crowd members A and B who are outside the domain of the local spy *spy*, i.e., *spy* does not observe the traffic in and out of A or B. Since spies can block messages that crowd members send, the statement x *is for* A must be defined in terms of message structure, and not in terms of where x eventually ends up:

$$x \textit{ is for } A =_{def} \exists y.fst(dec(x,y)) = A \wedge \bigvee_B B \textit{ sent } x$$

where B ranges over crowd members. (B *sent* might be a primitive predicate.) Sender anonymity, on the other hand, means that a spy cannot tell the originator of a message:

$$A \textit{ originated } x \rightarrow \neg\Box_{spy} A \textit{ originated } x \tag{9.5}$$

$$A \textit{ originated } x \rightarrow \Diamond_{spy} B \textit{ originated } x \tag{9.6}$$

for crowd members A and B who are outside the reach of *spy*. (A *originated* might be a primitive predicate.)

Although specifications (9.5) and (9.6), unlike the other specifications in chapter 9, do not directly specify knowledge of cryptographic structure, cryptographic omniscience is problematic also for these specifications. 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:

$$x \textit{ has source field } A \rightarrow \neg B \textit{ originated } x,\ B \neq A$$

where

$$x \textit{ has source field } A =_{def} \exists y.fst(snd(dec(x,y))) = A$$

By the rule of normality (i.e., the necessitation rule together with axiom K) and cryptographic omniscience,

$$x \textit{ has source field } A \rightarrow \Box_{spy} \neg B \textit{ originated } x,\ B \neq A$$

Thus, (9.6) necessarily fails, contrary to intuition. Specification (9.6) is similarly problematic under cryptographic ommniscience.

9.3 Dual Signature

Consider a purchasing protocol involving three parties, a customer C, a merchant M, and a bank B.[1] To order an item x_i using payment data (credit card number,

[1] This example is inspired by [9].

etc.) x_p, the customer produces a dual signature [62] using the private signing key x_s:

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

Both the merchant and the bank receive the dual signature $dual(x_i, x_p, x_s)$. The merchant receives, in addition, the order item x_i and the hash $h(x_p)$ of the payment data. Conversely, the bank receives the payment data x_p and the hash $h(x_i)$ of the order item. The idea is that the dual signature hides the order item x_i from the bank, and the payment data x_p from the merchant, but nonetheless the dual signature links x_i to x_p so that their correspondence cannot later be disputed. We now 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 the customer 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:

$$\Box_B C \; signed \; x_d$$

where

$$C \; signed \; x_d =_{def} \exists x_s.\exists y.x_d = sign(y, x_s) \wedge x_s \; sign \; key \; of C$$

($sign \; key \; of C$ might be a primitive predicate.) Using $h(x_i)$ and x_p, the bank can determine the payment data x_p inside:

$$\Box_B x_d \; contains \; payment \; x_p$$

where

$$x_d \; contains \; payment \; x_p =_{def} \exists x_i.\exists x_s.x_d = dual(x_i, x_p, x_s)$$

But, the bank cannot determine the order item:

$$\neg\Box_B x_d \; contains \; item \; x_i$$

where

$$x_d \; contains \; item \; x_i =_{def} \exists x_p.\exists x_s.x_d = dual(x_i, x_p, x_s)$$

Finally, the bank is assured that the merchant can determine the order item:

$$\Box_B \exists x_i.\Box_M x_d \; contains \; item \; x_i$$

Chapter 10

Expressiveness Results

In this chapter, we collect various results concerning the expressiveness of the language: A definition of message deduction in terms of the epistemic modality; A logical characterization of static equivalence; An undefinability result for the *de dicto* quantifier; A preservation result for the non-normal modality of part I of the thesis; And, finally, correspondence results for conditions on substitutions ρ.

10.1 Characterization of Message Deduction and Static Equivalence

As corollaries 8.3.2 and 8.3.3 show, there are strong interactions between the epistemic modality and message deduction. In fact, if the background equational theory contains a hash function h, message deduction reduces to the epistemic modality. Assume that for each $s \in \mathcal{S}$, there are at least two messages that A cannot infer at s, i.e., $|\overline{\mathit{Inferable}(A,s)}| \geq 2$.

Theorem 10.1.1 (Characterization of Inference) *The following is valid in $\mathcal{S}$:*

$$A \mathit{\ infers\ } x \leftrightarrow \exists y.\Box_A y = h(x)$$

Proof Assume that $s, V \models_{\mathcal{I}} A \mathit{\ infers\ } x$. By corollary 7.1.3, 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 $V(x) \notin \mathit{Inferable}(A,s)$. Assume $V(y) = h(V(x))$ for some given y. Pick a message M such that $M \notin \mathit{Inferable}(A,s)$ and $V(x) \neq M$. (There are at least two non-inferred messages, by our restriction on systems.) Let permutation ρ be the identity on $\mathcal{T}_{\equiv}$, except that $\rho(V(x)) = M$ and $\rho(M) = V(x)$. By corollary 7.2.4, $s \sim_A^{\rho} s$. 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., $s, V \not\models_{\mathcal{I}} \Box_A y = h(x)$. □

According to theorem 10.1.1, the agent deduces value x just in case the agent can recognize the hash of x as being the hash of x. (Recall that the interpretation of predicate A *infers* at state s is $\mathit{Inferable}(A, s)$). In light of theorem 10.1.1, we introduce $\Box_A\, x$, read "A knows x", as an abbreviation of the statement:

$$\exists y.\Box_A y = h(x)$$

We write $\Box_A\, \overline{x}$ for $\bigwedge_i \Box_A\, x_i$, and we write $\neg\Box_A\, \overline{x}$ for $\bigwedge_i \neg\Box_A\, x_i$. Theorem 10.1.1 is related to a result in [2], which reduces message deduction to static equivalence, while assuming a hash function.

In the remainder of section 10.1, we assume that local states $s|A$ and $s'|A$ are normal and that predicates $\mathcal{P}$ includes $@_l$, for $l \in \mathit{LOC}$. The following result provides a logical characterization of static equivalence.

Theorem 10.1.2 (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) ⇒ (2): By proposition 7.2.3 and theorem 7.2.6. (2) ⇒ (1): Assume (1) fails. Then, there is a statement $F =_{df} \exists \overline{x}.t = t' \wedge \bigwedge_i @_{l_i}(x_i)$, where locations $l_i \in \mathit{LOC}|A$ 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 $s \models t = t' \rightarrow \Box_A\, \mathit{VAR}(t) \rightarrow \Box_A t = t'$. The latter can be shown directly, or from lemma 11.1.1.10 and soundness theorem 11.2.1. □

The logical characterization of static equivalence, though immediate, gives added credence to the semantics, and allows the transfer of computational soundness results, such as that of [1], to the 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 states in s and s' are computationally indistinguishable.

10.2 Undefinability of the De Dicto Quantifier

We show that the *de dicto* quantifier adds to the expressive power. Assume a set Γ of statements, free from *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. $\mathsf{w} : S \longrightarrow S'$ is a surjective map

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

Morphism condition (3) might appear tautological, but this is not so. As explained in section 8.3, $V(t) = V(t')$ need not imply that $(\rho \circ V)(t) = (\rho \circ V)(t')$.

Lemma 10.2.1 *$s, V \models_{\mathcal{S}} F$ iff $\mathsf{w}(s), (\mathsf{d}_s \circ V) \models_{\mathcal{S}'} F$, for $F \in \Gamma$.*

Proof Straightforward induction on F. □

Next, we show that lemma 10.2.1 fails if Γ contains *de dicto* quantifiers. Approximately, 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 10.2.2 *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 $\mathsf{w} : S \longrightarrow S'$ such that $\mathsf{w}(s_1) = \mathsf{w}(s_2) = s_1$. 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 10.2.3 *$\mathcal{S}'_{cd}$ but not $\mathcal{S}_{cd}$ satisfies $\exists m.\exists x.x \neq A \wedge \Box_A x = m$.*

Theorem 10.2.4 *No statement free of the* de dicto *quantifier 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 10.2.1 and lemma 10.2.2, F does not distinguish between $\mathcal{S}'_{cd}$ and $\mathcal{S}_{cd}$. But, by corollary 10.2.3, the statement $\exists m.\exists x.x \neq A \wedge \Box_A x = m$ does distinguish the two systems. □

10.3 Preservation Result for Non-normal Modality

The present first-order language preserves the expressiveness of the form of propositional, non-normal language in part I, in which all terms refer *de re*. Let a propositional statement be any statement β containing no variables, no quantifiers and no equality symbol. Define a translation τ from propositional statements to (first-order) statements as follows.

Definition 10.3.1 (Translation τ)

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

where $\overline{M}$ is a list $\langle M_1, ..., M_n\rangle$ of all ground terms occurring as arguments to predicates in β, $\beta(\overline{x})$ is the result of substituting x_i for M_i, $\exists\overline{x}$ abbreviates $\exists x_1 ... \exists x_n$, and $\overline{x} = \overline{M}$ abbreviates $\bigwedge_i x_i = M_i$.

For instance, τ translates $\Box_A\Box_B A\,\mathit{received}\,\mathit{enc}(M,K)$ to $\exists x.x = \mathit{enc}(M,K) \wedge \Box_A\Box_B A\,\mathit{received}\,x$. Write $s \models^\tau_{\mathcal{I}} \beta$ for $s \models_{\mathcal{I}} (\beta)^\tau$.

Proposition 10.3.2 *The following are equivalent:*

- $s \models^\tau_{\mathcal{I}} \Box_A\beta(\overline{M})$
- $\forall s' \in S : \forall\rho : s \sim^\rho_A s' \Rightarrow s' \models^\tau_{\mathcal{I}} \beta(\rho(\overline{M}))$

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

Again, proposition 10.3.2 is rather immediate. Still, it shows that the present normal semantics preserves the expressiveness of the non-normal semantics of part I of the thesis: The translation τ induces the form of propositional semantics studied in part I, where instead of updating the assignment, terms inside the evaluated statement are updated as one follows the indistinguishability relation from a state s to another state s'. Of course, the relativized indistinguishability relation $\sim^\rho_A$ used in part I is different, the main difference being a restriction to pairing and symmetric encryption.

10.4 Abstract Correspondence Results

The relativized indistinguishability relation $\sim^\rho_A$ was defined by certain conditions on message substitutions ρ. In this section, we provide correspondence results for these, and some other, conditions. Throughout section 10.4, 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 10.4.1 *$s \sim^\rho_A s'$ implies that*

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

respectively, if and only if,

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

respectively.

Proof Routine. □

Refer to the schema in proposition 10.4.1.4 as *local state introspection*. The third schema is the weakening of cryptographic omniscience (8.5) that first appeared in corollary 8.3.2. The first two correspondences above are well-known in counterpart semantics (cf. [22]).

The following correspondence for cryptographic omniscience (8.5) is also well-known.

Proposition 10.4.2 *$s \sim_A^{\rho} s'$ implies $\rho(M) = M$, if and only if, $\mathcal{S}$ satisfies (8.5).*

Proof Routine. □

By proposition 10.4.1.4 and proposition 10.4.2, we obtain an instance of classical multi-agent semantics (section 3.2), if we define the relativized accessibility relation for A as the most inclusive $\sim_A$ which validates cryptographic omniscience (8.5) as well as local state introspection (i.e., the schema in proposition 10.4.1.4). The combination of cryptographic omniscience and local state introspection leads to *local state omniscience*:

$$@_l\, M \rightarrow \Box_A @_l\, M,\ l \in LOC|A \tag{10.1}$$

which goes against the assumption of limited decryption power of agents, even if variables do not intend *de re* reference. For instance, (10.1) leads to:

$$@_l\, enc(M, K) \rightarrow \Box_A \exists x.@_l\, enc(M, x),\ l \in LOC|A$$

In classical multi-agent semantics based on message passing systems (section 2.2), local state omniscience manifests itself in similar counter intuitive validities, for example (cf. section 3.2):

$$A\ received\, enc(M, K) \rightarrow \Box_A \exists x.A\ received\, enc(M, x)$$

Finally, we provide a correspondence result for schema (8.7), which was used in corollary 8.3.3. 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 10.4.3 *Non-deducible messages are anonymous, if and only if, $\mathcal{S}$ satisfies schema (8.7).*

Proof Only-if direction: Straightforward. If direction: Assume that non-deducible messages are not anonymous, i.e., there is a state $s \in S$ and finite $X \subseteq \overline{Infers(A,s)}$ and permutation π on X such that $s \not\sim_A^\pi s$. Pick $x_1, ..., x_n, z_1, ...z_n \in VAR$ and assignment V such that $dom(\pi) = \{V(x_1), ..., V(x_n)\}$ 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 $s' \in S$, except that $\langle V(z_1), ..., V(z_n)\rangle \notin I(p,s)$. Let $\mathcal{I}$ be the interpreted system based on $\mathcal{S}$ and I. Since $s \not\sim_A^\pi s$, we have $s, V \models_\mathcal{I} \Box_A p(x_1, ..., x_n)$. Also, $s, V \models_\mathcal{I} \neg A\, infers\, \overline{x}, \overline{z}$. 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, $s, V \not\models_\mathcal{I} p(z_1, ..., z_n)$. □

Chapter 11

Axiomatization

In this chapter, we provide the main result of part II, a sound and complete axiomatization of validity. The axiomatization extends the background equational theory with standard axioms and rules from first-order modal S5, an omega-rule for quantifiers, plus some novel axioms for the interaction between the epistemic modality and the equational theory.

11.1 Proof System

In table 11.1, we define a Hilbert-style axiomatization, relative to a message congruence $\equiv$ with a hash function h. The first group of axioms and rules is inherited from first-order logic, and includes the less standard axiom $(m\, x)$, 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$2) is well-known from first-order modal logic, the other four axioms are new. Axiom ($\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 agent knows a property of some non-inferred values $\overline{x}$ only if this property holds for any non-inferred values $\overline{z}$ with the same pattern of identities.[1] Axiom ($\Box$5) reflects the restriction on systems needed for theorem 10.1.1, namely that there are at least two messages that agent A does not infer. Propositions 10.4.1 and 10.4.3 provide correspondence results related to axioms ($\Box$1) and ($\Box$4). The fourth group includes all equalities and inequalities from $\equiv$ and an omega-rule for the *de dicto* quantifier. The omega-rule is unfortunate, since it produces infinite branching proof trees. But, since the equational theory is arbitrary, some infinitary machinery is needed.[2] Write $\vdash F$ when F is a derivable

[1] Recall that $F(\overline{x}, \overline{y})$ signifies that the list $\overline{x}, \overline{y}$ consists precisely of all variables free in F.

[2] By incompleteness of arithmetic, no finitary axiomatization is possible, not even for the modality-free fragment.

First-Order

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

Modal S5

(K)	$\Box_A(F \to F') \to (\Box_A F \to \Box_A F')$
(T)	$\Box_A F \to F$
(4)	$\Box_A F \to \Box_A \Box_A F$
(5)	$\neg\Box_A F \to \Box_A \neg\Box_A F$
(Nec)	$\dfrac{F}{\Box_A F}$

Knowledge and Cryptography

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

Omega

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

Figure 11.1: Axioms and Rules

theorem.

Lemma 11.1.1 *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. $\Box_A F(\overline{x}, \overline{y}) \rightarrow \Box_A \overline{y} \rightarrow \neg\Box_A \overline{x}, \overline{z} \rightarrow \bigwedge_{i,j}(x_i = x_j \leftrightarrow z_i = z_j) \rightarrow \Box_A F[\overline{z}/\overline{x}]$
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$4), Lemma 11.1.1.3, 11.1.1.4 and 11.1.1.5. (10): By induction on t. Base case: Axiom ($\Box$2). Induction step: Axiom ($\Box$1). □

For a message congruence $\equiv$ without a hash function, we obtain a sound and complete axiomatization if we take $\Box_A\, x$ as a primitive unary predicate and add the schema in lemma 11.1.1.3 as an additional axiom. The completeness construction in the following sections is not affected.

11.2 Soundness and Completeness

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

Theorem 11.2.1 (Soundness) $\vdash F \Rightarrow \models F$

Proof ($\Box$1): Theorem 10.1.1 and corollary 8.6. ($\Box$2): Since ρ is a function. ($\Box$3): Theorem 10.1.1. ($\Box$4): Corollary 7.2.4 and theorem 10.1.1. ($\Box$5): Theorem 10.1.1 and our restriction on systems. (T), (4) and (5): Proposition 7.2.3. (K) and (Nec): Independent of the definition of the relativized $\sim_A$. Non-epistemic axioms and rules are routine. □

Theorem 11.2.2 (Completeness) $\models F \Rightarrow \vdash F$

In the rest of this chapter, we build the completeness construction. The chapters that follow afterwards can be read independently. The completeness construction uses abstract counterpart models, with arbitrary states ("possible worlds") w, arbitrary domain of quantification, arbitrary accessibility relation $\longrightarrow_A^\rho$ 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 $\mathcal{K}^\star$ 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 $\mathcal{K}^\star$ into a counterpart model $\mathsf{d}(\mathcal{K}^\star)$ with a rigid interpretation of function symbols f, given by the background message equivalence $\equiv$. Finally, a morphism w transforms $\mathsf{d}(\mathcal{K}^\star)$ into a counterpart model $\mathsf{w}(\mathsf{d}(\mathcal{K}^\star))$, which is equivalent to an interpreted system.

11.3 Abstract Counterpart Model

We review some basics from (a variant of) counterpart semantics (cf. [34]). 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 for A, w and w' are indistinguishable, and for A, each $d \in D$ at w corresponds 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: $V(x, w) = V(x)$, $V(c, w) = I(c, w)$, $V(f(t_1, ..., t_n), w) = I(f, w)(V(t_1, w), ..., V(t_n, w))$. Truth conditions are as follows:

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

where ρ ranges over mappings $D \longrightarrow D$. 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 8.1 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 11.3.1 *$s, V \models_\mathcal{I} F$ iff $s, V \models_{\mathcal{C}_I} F$.*

A counterpart model $\mathcal{C}$ is Kripkean if $w \longrightarrow_A^\rho w'$ implies that $\rho = Id$, where Id is the identity on D. When $\mathcal{C}$ is Kripkean, we omit the index Id, and write $w \longrightarrow_A w'$ for

the transition $w \longrightarrow_A^{Id} w'$. We say that substitutions are bijective in $\mathcal{C}$, if $w \longrightarrow_A^{\rho} w'$ implies ρ is a permutation on D.

Assume a counterpart model $\mathcal{C} = \langle W, D, \longrightarrow, I\rangle$. 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:

- $\mathsf{w} : W \longrightarrow W'$ is a bijective map
- $\mathsf{d}_w : D \longrightarrow D'$ 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 the 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 $\mathsf{wd}(\mathcal{C}) = \langle W', D', \xrightarrow{\mathsf{wd}}, I^{\mathsf{wd}}\rangle$, where

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

Thus, $\mathsf{wd}(\mathcal{C})$ is the result of pointwise "relabeling" $\mathcal{C}$ through w and d.

Lemma 11.3.2 $w, V \models_{\mathcal{C}} F \Leftrightarrow \mathsf{w}(w), \mathsf{d}_w \circ V \models_{\mathsf{wd}(\mathcal{C})} F$.

Proof By induction on t, $(\mathsf{d}_w \circ V)(t, \mathsf{w}(w)) = \mathsf{d}_w(V(t, w))$. The lemma then follows by induction on F. □

11.4 Canonical Kripke Model

In this section, we obtain the truth lemma for a canonical Kripke model in a standard way [37]. A statement F is derivable from a set Γ of statements, in symbols $\Gamma \vdash F$, if there is a finite number of statements $F_1, ..., F_n \in \Gamma$ such that $\vdash F_1, ..., F_n \rightarrow F$. The set Γ is consistent if $\Gamma \nvdash \bot$, 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 11.4.1 $\emptyset$ *is omega-complete.*

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

Lemma 11.4.2 *If* Γ *is omega-complete then so is* $\Gamma \cup \{F\}$.

Proof Omega-completion for *de re* quantifiers: Standard. Assume that Γ is omega-complete w.r.t. *de dicto* quantifiers. Assume that $\Gamma, F_0 \vdash F[M/x]$ 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 11.4.3 *If* Γ *is omega-complete then so is* $\Gamma|A$.

Proof Omega-completion for *de re* quantifiers: Standard. Assume that Γ is omega-complete w.r.t. *de dicto* quantifiers. Assume that $\Gamma|A \vdash F[M/x]$ 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 11.1.1.2, $\Gamma \vdash \Box_A \forall m.F[m/x]$, i.e., $\Gamma|A \vdash \forall m.F[m/x]$. □

Lemma 11.4.4 (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 $F_1, F_2, ...$ of all statements. We define a sequence of extensions of Γ as follows:

- $\Gamma_0 = \Gamma$.
- If Γ_{n-1}, $F_n \vdash \bot$, $F_n = \forall m.F[m/x]$,
  then $\Gamma_n = \Gamma_{n-1} \cup \{\neg\forall m.F[m/x], \neg F[M/x]\}$
- else if Γ_{n-1}, $F_n \vdash \bot$, $F_n = \forall x.F$,
  then $\Gamma_n = \Gamma_{n-1} \cup \{\neg\forall x.F, \neg F[y/x]\}$
- else if Γ_{n-1}, $F_n \vdash \bot$, 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 that Γ_{n-1} is consistent. Assume that $\Gamma_{n-1}, \forall m.F[m/x] \vdash \bot$. Assume that there is no appropriate M, i.e., assume that $\Gamma_{n-1}, \neg\forall m.F[m/x], \neg F[M/x] \vdash \bot$ for all M, i.e., $\Gamma_{n-1}, \neg\forall m.F[m/x] \vdash F[M/x]$ all M. By lemma 11.4.2, 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, $\Gamma_{n-1}, \forall m.F[m/x] \vdash \bot$, and so, $\Gamma_{n-1} \vdash \bot$, contrary to assumptions. In the same way, lemma 11.4.2 tells us that there is an appropriate y. Thus, Γ_n is consistent,and, consequently, $\Gamma^\star = \bigcup_n \Gamma_n$ is a maximal consistent set. Trivially, $\Gamma^\star$ is omega-complete. Thus, $\Gamma^\star$ is saturated and $\Gamma \subseteq \Gamma^\star$.□

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: $w \longrightarrow_A w' \Leftrightarrow w|A \subseteq w'$, where $w|A$ is $\{F : \Box_A F \in w\}$. Finally, the interpretation is defined as follows: $I(f, w)(|x_1|, ..., |x_n|) = |y|$ iff $(f(x_1, ..., x_n) = y) \in w$, and $I(c, w) = |y|$ iff $(c = y) \in w$. The canonical assignment $V_\mathcal{K}$ assigns $|x|$ to variable x.

Lemma 11.4.5 (Truth Lemma for $\mathcal{K}$) $w, V_{\mathcal{K}} \models_{\mathcal{K}} F \Leftrightarrow F \in w$

Proof From lemmas 11.4.2, 11.4.3 and 11.4.4. The proof is standard. □

Corollary 11.4.6 $w \longrightarrow_A w' \Leftrightarrow w|A = w'|A$

Proof S5. □

11.5 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 $\mathcal{C} = \langle W, D \longrightarrow, I\rangle$. Write $\mathit{Inferable}_{\mathcal{C}}(A, w)$ for the set of items inferred by agent A at world w, i.e, $\mathit{Inferable}_{\mathcal{C}}(A, w)$ is $\{d \in D \mid w, V[x \mapsto d] \models_{\mathcal{C}} \Box_A x\}$. The anonymization of $\mathcal{C}$ is the model $\mathcal{C}^\star = \langle W, D \xrightarrow{\star}, I\rangle$, where $\xrightarrow{\star}$ is the least extension of $\longrightarrow$ such that

$$w \xrightarrow{\star}{}^{\rho}_{A} w' \Rightarrow w \xrightarrow{\star}{}^{\rho\circ\pi}_{A} w'$$

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

Corollary 11.5.1 *$w \xrightarrow{\star}{}^{\rho}_{A} w'$, if and only if, there is ρ' and π such that $\rho = \rho' \circ \pi$ and $w \longrightarrow^{\rho'}_{A} w'$ and π is a permutation on $\overline{\mathit{Inferable}_{\mathcal{C}}(A, w)}$.*

Proof Immediate. □

Lemma 11.5.2 *Assume that $\mathcal{C}$ validates the schema in lemma 11.1.1.9. Assume that substitutions are bijective in $\mathcal{C}$. Then, $w, V \models_{\mathcal{C}} F \Leftrightarrow w, V \models_{\mathcal{C}^\star} F$.*

Proof By induction on the complexity of F. Base case, and induction step for Boolean operators and quantifiers: Immediate. Induction step for modal operators: If $w, V \models_{\mathcal{C}^\star} \Box_A F$ then $w, V \models_{\mathcal{C}} \Box_A F$, since $\xrightarrow{\star}{}_A \supseteq \longrightarrow_A$, from corollary 11.5.1. For the converse, assume

$$w, V \models_{\mathcal{C}} \Box_A F \tag{11.1}$$

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

$$w, V \models_{\mathcal{C}} \neg\Box_A x_i \tag{11.2}$$

$$w, V \models_{\mathcal{C}} \Box_A y_i \tag{11.3}$$

Assume that $w \xrightarrow{\star}{}^{\rho}_{A} w'$. By corollary 11.5.1, there is ρ' and π such that $\rho = \rho' \circ \pi$ and $w \longrightarrow^{\rho'}_{A} w'$ and π is a permutation on $\overline{\mathit{Inferable}_{\mathcal{C}}(A, w)}$. Thus,

$$\rho'(V(y_i)) = \rho(V(y_i)) \tag{11.4}$$

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

$$\rho'(d_i) = \rho(V(x_i)) \tag{11.5}$$

and

$$d_i = d_j \Leftrightarrow V(x_i) = V(x_j) \tag{11.6}$$

Pick fresh variables $z_1, ..., z_m$ (i.e., fresh w.r.t. F). By (11.6),

$$w, V[z_1 \mapsto d_1, ...z_m \mapsto d_m] \models_{\mathcal{C}} z_i = z_j \leftrightarrow x_i = x_j \tag{11.7}$$

We have:

$$w, V[z_1 \mapsto d_1, ...z_m \mapsto d_m] \models_{\mathcal{C}} \neg\Box_A z_i \tag{11.8}$$

To see this, assume that $d_i \in \mathit{Inferable}_{\mathcal{C}}(A, w)$. Then, $\rho(d_i) = \rho' \circ \pi(d_i) = \rho'(d_i)$ = (by (11.5)) = $\rho(V(x_i))$. Since ρ is bijective, $d_i = V(x_i)$, contradicting (11.2). From (11.1), (11.2), (11.3), (11.7) and (11.8) and the assumption that $\mathcal{C}$ validates the schema in lemma 11.1.1.9,

$$w, V[z_1 \mapsto d_1, ...z_{mm}] \models_{\mathcal{C}} \Box_A F[\overline{z}/\overline{x}]$$

Thus,

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

By induction assumption,

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

By (11.4) and (11.5), we get $w', \rho \circ V \models_{\mathcal{C}^\star} F$. Since w' and ρ were chosen arbitrarily, we conclude that $w, V \models_{\mathcal{C}^\star} \Box_A F$. □

Lemma 11.5.3 *$w, V \models_{\mathcal{K}} F \Leftrightarrow w, V \models_{\mathcal{K}^\star} F$.*

Proof From lemma 11.5.2, since the assumptions for that lemma hold: Substitutions are bijective in $\mathcal{K}$: By construction of $\mathcal{K}$, $w \longrightarrow^{\rho}_{A} w'$ implies that ρ is the identity on D, i.e., a bijection. $\mathcal{K}$ validates the schema in lemma 11.1.1.9: Lemma 11.1.1.9 and lemma 11.4.5. □

11.6 Rigid Operators

We define a domain-morphism d, which morphs $\mathcal{K}^\star$ into a model $\mathsf{d}(\mathcal{K}^\star)$ 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 $(\not\equiv)$. For each $w \in W$, we relate D and $\mathcal{T}_{\equiv}$ by the relation:

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

Lemma 11.6.1 *d is a morphism from $\mathcal{K}^\star$ to W and $\mathcal{T}_{\equiv}$.*

Proof From axioms $(m\ x)$, $(\equiv)$, $(\not\equiv)$ and $(Subst)$, and lemma 11.1.1.8. □

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

Lemma 11.6.2 *$I^d(f, w) = f$ and $I^d(c, w) = c$.*

Proof From axioms $(\equiv)$, $(\not\equiv)$ and $(Subst)$ and lemma 11.6.1. □

We end this section with two lemmas that will be used in the final transformation 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 11.6.3 *Assume that $w \xrightarrow{d}{}^{\rho}_{A} w'$. Then,*

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

Proof Assume that $w \xrightarrow{\mathsf{d}}{}^{\rho}_{A} w'$. (1): From corollary 11.5.1 and lemma 11.6.1. (3): By construction of $\xrightarrow{\mathsf{d}}_A$, $w \xrightarrow{\star}{}^{\rho'}_{A} w'$ where $\rho' = \mathsf{d}_{w'}^{-1} \circ \rho \circ \mathsf{d}_w$. By corollary 11.5.1, $\rho'(|x|) = |x|$ for $|x| \in Inferable_{\mathcal{K}}(w, A)$. Thus, $\rho(M) = \mathsf{d}_{w'} \circ \mathsf{d}_w^{-1}(M)$ if $M \in \mathsf{d}(Inferable_{\mathcal{K}}(w, A))$. (2): Assume that $M_1, ..., M_n \in \mathsf{d}(Inferable_{\mathcal{K}}(w, A))$. I.e., there are variables $x_1, ...x_n$ such that $\Box_A\, x_i \in w$ and $x_i = M_i \in w$. Pick variable y such that $y = f(x_1, ...x_n) \in w$. By axiom $(\Box 1)$, $\Box_A y = f(x_1, ...x_n) \in w$. By corollary 11.4.6, $y = f(x_1, ...x_n) \in w'$. By lemma 11.6.2, $\mathsf{d}_{w'}(|y|) = f(\mathsf{d}_{w'}(|x_1|, ..., \mathsf{d}_{w'}(|x_n|)))$ and $\mathsf{d}_w(|y|) = f(\mathsf{d}_w(|x_1|), ..., \mathsf{d}_w(|x_n|))$. By axiom $(\Box 3)$, $\Box_A\, y \in w$. By (3), $\rho(\mathsf{d}_w(|y|)) = \mathsf{d}_{w'}(|y|)$. But, from above, we have that $\mathsf{d}_{w'}(|y|) = f(\mathsf{d}_{w'}(|x_1|), ..., \mathsf{d}_{w'}(|x_n|))) = f(\rho(\mathsf{d}_w(|x_1|)), ..., \mathsf{d}_w(|x_n|))$. Thus, $\rho(f(M_1, ..., M_n)) = f(\rho(M_1), ..., \rho(M_n))$, since $\mathsf{d}_w(|y|) = f(M_1, ..., M_n)$ from axiom $(Subst)$ and the fact that $y = f(X_1, ...x_n) \in w$. □

Lemma 11.6.4 *Assume that*

1. *$w \longrightarrow_A w'$.*
2. *ρ is a permutation on $\mathcal{T}_\equiv$.*
3. *$\rho(M) = \mathsf{d}_{w'} \circ \mathsf{d}_w^{-1}(M)$ if $M \in \mathsf{d}(Inferable_{\mathcal{K}}(w, A))$.*

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

Proof Let $\rho' = \mathsf{d}_{w'}^{-1} \circ \rho \circ \mathsf{d}_w$. By assumptions (2) and (3) and lemma 11.6.1, ρ' is identity on $Inferable_{\mathcal{K}}(w, A)$ and permutes $\overline{Inferable_{\mathcal{K}}(w, A)}$. By assumption (1) and corollary 11.5.1, $w \xrightarrow{\star}{}^{\rho'}_{A} w'$. By construction of $\xrightarrow{\mathsf{d}}_A$, $w \xrightarrow{\mathsf{d}}{}^{\rho}_{A} w'$. □

11.7 Canonical Interpreted System

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

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

(where D is the domain in $\mathcal{K}$ and $\mathcal{K}^\star$). 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. $\mathsf{w}(w)(\langle |x|, A\rangle) = \mathsf{d}_w(|x|)$, if $|x| \in \mathit{Inferable}_{\mathcal{K}}(w, A)$.
2. $\mathsf{w}(w)(\langle |x|, A\rangle) = \bot$, if $|x| \notin \mathit{Inferable}_{\mathcal{K}}(w, A)$.
3. $\mathsf{w}(w)(\langle F, A\rangle) = \top$, if $\Box_A F \in w$.
4. $\mathsf{w}(w)(\langle F, A\rangle) = \bot$, if $\Box_A F \notin w$.
5. $\mathsf{w}(w)(F) = \top$, if $F \in w$.
6. $\mathsf{w}(w)(F) = \bot$, if $F \notin w$.

where $\bot$ 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 $\bot = A$ and $\top = h(A)$.) Requirements (3) and (4) on w encode the knowledge state $w|A$ inside the local state $\mathsf{w}(w)|A$. Requirements (5) and (6) ensure injectivity. Requirements (1) and (2), together with (3) and (4), ensure that the same permutations ρ are possible between $\mathsf{w}(w)$ and $\mathsf{w}(w')$ in $\sim_A$ as between w and w' in $\xrightarrow{\mathsf{d}}_A$.

Corollary 11.7.1 *w is a morphism from $\mathsf{d}(\mathcal{K}^\star)$ to S and $\mathcal{T}_{\equiv}$.*

Proof Since w is injective. □

Lemma 11.7.2 $\mathit{Inferable}(A, \mathsf{w}(w)) = \mathsf{d}_w(\mathit{Inferable}_{\mathcal{K}}(A, w))$.

Proof $\mathit{Inferable}(\mathsf{w}(w), A) \supseteq \mathsf{d}_w(\mathit{Inferable}_{\mathcal{K}}(w, A))$: From condition (1) in the definition of w. $\mathit{Inferable}(\mathsf{w}(w), A) \subseteq \mathsf{d}_w(\mathit{Inferable}_{\mathcal{K}}(w, A))$: By induction on length of the derivation that establishes $M \in \mathit{Inferable}(\mathsf{w}(w), A)$. Base case. Assume that $M \in ran(\mathsf{w}(w)|A)$. If $M \in \{\top, \bot\}$ then $M \in \mathsf{d}_w(\mathit{Inferable}_{\mathcal{K}}(w, A))$, by lemma 11.1.1.6. On the other hand, if M is $\mathsf{d}_w(|x|)$ and $|x| \in \mathit{Inferable}_{\mathcal{K}}(w, A)$, then, trivially, $M \in \mathsf{d}_w(\mathit{Inferable}_{\mathcal{K}}(w, A))$. Induction step. Assume that $M_1, ..., M_n \in \mathit{Inferable}(\mathsf{w}(w), A)$. By induction assumption, $M_1, ..., M_n \in \mathsf{d}_w(\mathit{Inferable}_{\mathcal{K}}(w, A))$. I.e., there are $|x_1|, ..., |x_n| \in \mathit{Inferable}_{\mathcal{K}}(w, A)$ such that $\mathsf{d}_w(|x_i|) = M_i$. By axiom (*Subst*), $\Box_A x_i \in w$. Since $\vdash \exists y.y = f(x_1, ..., x_n)$, we have $y = f(x_1, ..., x_n) \in w$

for some $y \in VAR$. By lemma 11.6.2, $\mathsf{d}_w(|y|) = f(\mathsf{d}_w(|x_1|), ..., \mathsf{d}_w(|x_n|))$. By axiom ($\Box 3$), $\Box_A\, y \in w$, i.e., $|y| \in \mathit{Inferable}_{\mathcal{K}}(w, A)$, i.e., $\mathsf{d}_w(|y|) \in \mathsf{d}_w(\mathit{Inferable}_{\mathcal{K}}(w, A))$, i.e., $f(\mathsf{d}_w(|x_1|), ..., \mathsf{d}_w(|x_n|)) \in \mathsf{d}_w(\mathit{Inferable}_{\mathcal{K}}(w, A))$, i.e., we obtain $f(M_1, ..., M_n) \in \mathsf{d}_w(\mathit{Inferable}_{\mathcal{K}}(w, A))$. □

Lemma 11.7.3 *$w \xrightarrow{\mathsf{d}}{}^{\rho}_{A} w'$, if and only if, $\mathsf{w}(w) \sim^{\rho}_{A} \mathsf{w}(w')$.*

Proof $\Rightarrow$-direction: Assume that $w \xrightarrow{\mathsf{d}}{}^{\rho}_{A} w'$. We need to show that ρ is a permutation on $\mathcal{T}_{\equiv}$, ρ respects Σ on $\mathit{Inferable}(\mathsf{w}(w), A)$ and ρ respects $LOC|A$ between $\mathsf{w}(w)$ and $\mathsf{w}(w')$, i.e., $\rho \circ \mathsf{w}(w)|A = \mathsf{w}(w')|A$. (i) ρ is a permutation on $\mathcal{T}_{\equiv}$: Lemma 11.6.3.1. (ii) ρ respects Σ on $\mathit{Inferable}(\mathsf{w}(w), A)$: Lemma 11.6.3.2 and lemma 11.7.2. (iii) ρ respects $LOC|A$ between $\mathsf{w}(w)$ and $\mathsf{w}(w')$, i.e., $\rho(\mathsf{w}(w)|A) = \mathsf{w}(w')|A$: We show that $\rho(\mathsf{w}(w)(\langle |x|, A\rangle)) = \mathsf{w}(w')(\langle |x|, A\rangle)$; Respect for other locations is is shown similarly. By construction of $\xrightarrow{\mathsf{d}}_A$, we have $w \longrightarrow_A w'$. Assume that $|x| \in \mathit{Inferable}_{\mathcal{K}}(w, A)$. By corollary 11.4.6, $|x| \in \mathit{Inferable}_{\mathcal{K}}(w', A)$. By lemma 11.6.3.3 and condition (1) in the construction of w, $\rho(\mathsf{w}(w)(\langle |x|, A\rangle)) = \mathsf{w}(w')(\langle |x|, A\rangle)$. Assume that $|x| \notin \mathit{Inferable}_{\mathcal{K}}(w, A)$. By corollary 11.4.6, $|x| \notin \mathit{Inferable}_{\mathcal{K}}(w', A)$. By condition (2) in the construction of w, $\mathsf{w}(w)(\langle |x|, A\rangle) = \mathsf{w}(w')(\langle |x|, A\rangle) = \bot$. But, from lemma 11.6.3.2 and lemma 11.7.2, $\rho(\bot) = \bot$.

$\Leftarrow$-direction: Assume that $\mathsf{w}(w) \sim^{\rho}_{A} \mathsf{w}(w')$. We show conditions (1), (2) and (3) in lemma 11.6.4, from which it follows that $w \xrightarrow{\mathsf{d}}{}^{\rho}_{A} w'$. Condition (1): Since ρ respects Σ on $\mathit{Inferable}(\mathsf{w}(w), A)$, $\rho(\bot) = \bot$ and $\rho(\top) = \top$. Thus, since ρ respects $LOC|A$ between $\mathsf{w}(w)$ and $\mathsf{w}(w')$, we have $w|A = w'|A$, from conditions (3) and (4) in the construction of w. By corollary 11.4.6, $w \longrightarrow_A w'$. Condition (2): By construction of $\longrightarrow_A$. Condition (3): Since $w \longrightarrow_A w'$, by corollary 11.4.6, $|x| \in \mathit{Inferable}_{\mathcal{K}}(w, A)$ iff $|x| \in \mathit{Inferable}_{\mathcal{K}}(w', A)$. Let $|x| \in \mathit{Inferable}_{\mathcal{K}}(w, A)$. Since ρ respects Σ on $\mathit{Inferable}(\mathsf{w}(w), A)$, $\rho(\mathsf{d}_w(|x|)) = \mathsf{d}_{w'}(|x|)$, from condition (1) in the construction of w. □

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

Lemma 11.7.4 *$I(p, \mathsf{w}(w)) = I^{\mathsf{d}}(p, w)$.*

Proof From axiom (*Subst*). □

Lemma 11.7.5 *$\mathsf{w}(\mathsf{d}(\mathcal{K}^\star))$ is induced by $\mathcal{I}$.*

Proof From lemmas 11.6.1, 11.6.2, 11.7.3 and 11.7.4. □

Lemma 11.7.6 *$\mathsf{w}(w), \mathsf{d}_w \circ V_{\mathcal{K}} \models_{\mathcal{I}} F \Leftrightarrow F \in w$.*

Proof $F \in w$ iff (lemma 11.4.5) $w, V_{\mathcal{K}} \models_{\mathcal{K}} F$ iff (lemma 11.5.3) $w, V_{\mathcal{K}} \models_{\mathcal{K}^\star} F$ iff (lemmas 11.3.2 and 11.6.1) $w, \mathsf{d}_w \circ V_{\mathcal{K}} \models_{\mathsf{d}(\mathcal{K}^\star)} F$ iff (lemma 11.3.2 and corollary 11.7.1) $\mathsf{w}(w), \mathsf{d}_w \circ V_{\mathcal{K}} \models_{\mathsf{w}(\mathsf{d}(\mathcal{K}^\star))} F$ iff (lemmas 11.3.1 and 11.7.5) $\mathsf{w}(w), \mathsf{d}_w \circ V_{\mathcal{K}} \models_{\mathcal{I}} F$. □

Lemma 11.7.7 $\overline{\mathit{Inferable}(\mathsf{w}(w), A))}$ *has at least two members.*

Proof By axiom ($\Box 5$), $\overline{\mathit{Inferable}_{\mathcal{K}}(w, A)}$ has at least two members $|x|$ and $|y|$. By lemmas 11.6.1 and 11.7.2, $\overline{\mathrm{Inferable}(\mathsf{w}(w), A))}$ has at least the two members $\mathsf{d}_w(|x|)$ and $\mathsf{d}_w(|y|)$. □

Theorem 11.7.8 *Every consistent statement is satisfiable in some interpreted system.*

Proof Assume that $\nvdash \neg F$. By lemmas 11.4.1, 11.4.2 and 11.4.4, 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 11.7.6, $\mathsf{w}(w_0), \mathsf{d}_{w_0} \circ V_{\mathcal{K}} \models_{\mathcal{I}} F$. By lemma 11.7.7, $\mathcal{I}$ satisfies our requirement on systems. □

From theorem 11.7.8, we get completeness theorem 11.2.2.

Chapter 12

Embedding of BAN and SVO

In this chapter, we illustrate the axiomatization by embedding characteristic rules from BAN logic [16] and SVO logic [77].

12.1 BAN-Like Modality

By translation τ (definition 10.3.1) the axiomatization contains a propositional logic with *de re* reference of complex terms. We illustrate how the embedded propositional logic forms a BAN-like logic.

Recall from section 6.1 that BAN logic has no general proof rules for the epistemic modality, only rules specific to each predicate. In section 4.3, two weakenings of the rule of normality were introduced, and one of them later used in section 6.2 for a completeness result for BAN-like logics. These weakenings of normality arose in a context where the crypto algebra is the term algebra formed from operators for pairing and symmetric encryption. In the present context, the crypto algebra is given by an equational theory of feasible computable operators. For this more general context, we propose that the following omega-weakening of the rule of necessitation is faithful to BAN:

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

where $\overline{c}$ is all constants from *SEC* occurring in β. For instance:

$$\frac{\mathit{enc}(M,K)\ \mathit{contains}\ M,\ \mathrm{all}\,M,K}{\Box_A\,M,K \rightarrow \Box_A\ \mathit{enc}(M,K)\ \mathit{contains}\ M}$$

Let WNec^τ be the τ-translations of *WNec*.

Proposition 12.1.1 *WNec^τ 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 that

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

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 11.1.1.2,

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

By axiom ($m\,x$),

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

i.e.,

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

By lemma 11.1.1.10, 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.,

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

□

Using $WNec^\tau$, we proceed to derive the τ-translation of the following two BAN-style axioms:

$$A\, sees\, enc(M,K) \rightarrow \Box_A K\, good\, for\, G \rightarrow \Box_A \bigvee_{B \in G} B\, said\, M \qquad (WMMR)$$

$$fresh\, M \rightarrow \Box_A K\, good\, for\, G \rightarrow \Box_A\, fresh\, enc(M,K) \qquad (Fresh)$$

Schema (*WMMR*) weakens BAN's message meaning rule (*R1* in table 6.1), abstracting from the assumption that encryptions contain a reliable sender field (cf. proposition 6.3.4).[1] Schema (*Fresh*) is BAN rule *R9* (in table 6.1). Other BAN-style axioms can be derived similarly. Let *BAN* be the conjunction of the following four assumptions:

$$\forall x.A\, sees\, x \rightarrow \Box_A A\, sees\, x$$

[1] Schema *WMMR* also replaces the predicate *secret* in *R1* by *good*, with the intended meaning that a key is good for a group of agents if no one but group members send messages encrypted with that key. Original BAN logic [16] includes both predicates.

$$\exists x.\neg\Box_A\, x \wedge \neg x\ good\, for\, G$$

$$\exists x.\neg\Box_A\, x \wedge \neg A\ sees\, x$$

$$\exists x.\neg\Box_A\, x \wedge \neg fresh\, x$$

Trivially, an interpreted system $\mathcal{I}$ satisfies the first conjunct of BAN if, and only if, $\rho(I(A\ sees,\ s)) \subseteq I(A\ sees,\ s')$ whenever $s \sim_A^{\rho} s'$ in $\mathcal{I}$. Following [7], we assume the following abbreviation:

$$x\ good\, for\, G =_{df} \forall y.\bigvee_{A\in\mathcal{A}} A\ sees\ enc(y,x) \rightarrow \bigvee_{B\in G} B\ said\ y$$

Corollary 12.1.2 *The following are theorems:*

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

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

Proposition 12.1.3 $\vdash BAN \rightarrow (WMMR)^{\tau}$, *assuming* $\mathrm{dec}(\mathrm{enc}(M,K),K) \equiv M$.

Proof By proposition 12.1.1,

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

is a theorem. By corollary 12.1.2.1 and corollary 12.1.2.2,

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

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

Proposition 12.1.4 $\vdash BAN \rightarrow (Fresh)^{\tau}$, *if we add an additional axiom:* $fresh\, t \rightarrow fresh\ enc(t,t')$.

Proof By assumption,

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

is a theorem, for all M, K. By proposition 12.1.1,

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

is a theorem. By corollary 12.1.2.3 and corollary 12.1.2.4, $\vdash BAN \rightarrow (Fresh)^{\tau}$. □

12.2 SVO-Like Modality

Protocol derivations in SVO [77], a successor to BAN named after Syverson and van Oorschot, 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 for asymmetric encryption and pairing in example 7.1.1,

$$A\,sees\,enc(pair(x,x'),pk(z)),\ A\,infers\,z \rightarrow \Box_A A\,sees\,enc(pair(x,x'),pk(z)) \quad (12.1)$$

Implications from seeing to knowledge of seeing, such as (12.1), are not justified by the proof system in [77], but the authors remark that it would be straightforward to capture such implications in an axiom. We propose the following axiom:

$$A\,sees\,T \rightarrow \Box_A\ VAR(T) \rightarrow \Box_A A\,sees\,T \qquad (SEE)$$

where T is any term without constants from *SEC*. The semantics in [77] does not support (12.1) or *SEE*. More generally, the semantics there does not support *de re* reference of variables. We show, however, that our semantics fits (12.1) and *SEE*. Let SVO be the conjunction of thw following two statements:

$$\forall x.(A\,sees\,x \rightarrow \Box_A A\,sees\,x)$$

$$\exists x.(\neg\Box_A\,x \wedge \neg A\,sees\,x)$$

Proposition 12.2.1 *The following hold:*

1. $\vdash SVO \rightarrow A\,sees\,x \rightarrow \Box_A\,x.$
2. $\vdash SVO \rightarrow SEE$
3. $\vdash SVO \rightarrow (12.1)$

Proof (1): By axiom ($\Box 4$). (2): From lemma 11.1.1.10. (3): By equations in example 7.1.1, and axioms ($\equiv$) and ($\Box 3$) and proposition 12.2.1.1,

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

is a theorem. By proposition 12.2.1.2, $\vdash SVO \rightarrow (12.1)$. $\Box$

Chapter 13

Concluding Remarks

Our semantics in part II 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 [14]. 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 7.1. Such intensions refer non-rigidly, in that they pick out a different message at different states. 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 would appear to translate to something of the form:

$$\exists y.x = y \wedge A\text{-}term\,(y) \wedge \Box_A F(y)$$

where A-*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 this, the statement $\Box_B \Box_A F(x)$ translates to

$$\exists y.(x = y \wedge B\text{-}term(y) \wedge \Box_B \exists z.(z = y \wedge A\text{-}term(z) \wedge \Box_A F(z)))$$

One issue left open 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, using only this rule, it is difficult to see how to obtain a lemma corresponding to lemma 11.4.3 (with a suitably adjusted definition of omega-completion). In any case, the *de dicto* quantifier may have independent interset. (According to

theorem 10.2.4, the *de dicto* quantifier adds to the expressive power.) In an ongoing work with Mads Dam, it would appear that *de dicto* quantifiers, in combination with *de re* quantifiers, enable epistemic characterizations of knowledge concepts in information flow security, such as delimited release [72].

In the future, we plan to extend the completeness result (theorem 11.2.2) to include temporal modalities. It is known that first-order temporal logics (excluding some weak variants) are not finitely axiomatizable (cf. [26]). However, in our case, adding temporal modalities need not incur much extra cost, since even the modal-free first-order fragment required an infinitary proof rule. One possibility would be to add a binary next-time modality $\bigcirc$, taking a statement F and a term t as arguments: $\bigcirc_t F$ expresses that after t time steps, fact F will hold. Assuming that the equational theory includes a successor operator *suc* and the constant 0 one could add the axiom:

$$\bigcirc_{suc(t)} F \leftrightarrow \bigcirc_1 \bigcirc_t F$$

(where 1 abbreviates $suc(0)$) in addition to standard next-time axioms for modality $\bigcirc_1$, standard axioms for the interaction between next-time and knowledge and a standard rigidity axiom:

$$t = t' \rightarrow \bigcirc_1 t = t'$$

The modality $\Box$ for "It will always be the case that" can be introduced by abbreviation:

$$\Box F =_{df} \forall x.\, \bigcirc_x F$$

Assuming that the equational theory allows for the definition of a smaller-than relation $<$, the *until* modality can also be introduced by abbreviation:

$$F \,\text{UNTIL}\, F' =_{df} \exists x.\, \bigcirc_x F' \wedge \forall y.(y < x \rightarrow \bigcirc_y F)$$

There is, of course, a question of what $\bigcirc_t F$ should mean when t is not a number (i.e., when t is not equal to any of $0, suc(0), suc(suc(0)), \ldots$). One option would be to rule out such statements, by introducing sorts into the language.

Finally, it would be interesting to combine the completeness result (theorem 11.2.2) with computational completeness results for static equivalence (cf. [63]), thereby obtaining axiomatizations which are complete with respect to computational models of cryptography.

Chapter 14

List of Symbols for Part I

Messages

$\mathcal{T}$	Set of all message terms
$\mathcal{C}$	Set of all message atoms
$\mathcal{A}$	Set of all agents
M, K	Message
c	Message atom
A, B, C	Agent
$\cdot$	Pairing
$\{\}$	Symmetric encryption
κ	Set of messages
x, y, z	Variable
t	Open message term

Language

F	Statement
$\mathcal{P}$	Set of all predicates
$\mathcal{F}$	Set of all statements
Δ, Γ	Set of statements

System

Π	Action vocabulary
π	Action
i	Initialization
h	History
$\vert$	Observation function
θ	Action trace
$\mathcal{S}$	(Multi-agent) system
I	Predicate interpretation
$\mathcal{I}$	Interpreted system

Indistinguishability

$\sim_A$	Indistinguishability w.r.t. agent A
$\sim_A^\rho$	Indistinguishability w.r.t. A relativized to ρ
ρ	Permutation of messages
$[M - M']$	Permutation exchanging$M$ and M'
$[c - c'/\kappa]$	Permutation exchanging c and c' in parts inaccessible to κ
$\triangleleft$	Consistency of permutation w.r.t. set of messages

Canonical Counterpart Model

L	BAN theory
$L\boldsymbol{A}$	BAN theory genereated from $\boldsymbol{A}$
$\mathcal{C}_L$	Canonical counterpart model
W_L	Set of all maximal consistent sets
$\underset{L}{\longrightarrow_A^\rho}$	Accessibility between maximal consistent sets
Int_L	Interpretation function on maximal consistent sets
w	Maximal consistent set
$\rightsquigarrow$	Filtration from counterpart model to interpreted system

Canonical System

$\mathcal{I}_L$	Canonical interpreted system
H_L	Set of histories in $\mathcal{I}_L$
I_L	Interpretation function in $\mathcal{I}_L$

Chapter 15

List of Symbols for Part II

Message terms/Messages

$\mathcal{T}$	Set of all ground message terms
$\mathcal{A}$	Set of all agents
VAR	Set of variables
SEC	Set of all secret constants
Σ	Set of all feasibly computable operators
M, K	Ground message term/Message
c	Secret constant
A, B, C	Agent
f	Feasibly computable operator
enc	(Symmetric/Asymetric/Random) encryption function
dec	Decryption function
h	Hash function
len	Length function
x, y, z	Variable
t	Open term
$\equiv$	Congruence on ground terms
$\mathcal{T}_{\equiv}$	Set of all messages

Static Equivalence

LOC	Set of store locations
l	Location
s	Store
α	s-term built from $dom(s)$ and Σ
$\approx$	Static equivalence
ρ	Permutation of messages
$\sim^{\rho}$	Indistinguishability relativized to ρ

Multi-agent System

$\mathcal{S}$	(Multi-agent) system
S	Set of stores
$\|$	Observation function
I	Predicate interpretation
$\mathcal{I}$	Interpreted system
$\sim_A^\rho$	Indistinguishability w.r.t. A relativized to ρ

Canonical Kripke Model

$\mathcal{K}$	Canonical Kripke model
W	Set of all saturated sets
D	Domain of all equivalence classes $\|x\|$
$\longrightarrow_A$	Accessibility between saturated sets
I	Interpretation function on saturated sets
w	Saturated set
$V_\mathcal{K}$	Canonical assignment

Anonymized Canonical Model

$\mathcal{K}^\star$	Canonical model with anonymous non-inferred items
$\xrightarrow{\star}$	Accessibility in $\mathcal{K}^\star$

Rigid Anonymized Canonical Model

$\mathsf{d}(\mathcal{K}^\star)$	Canonical model with rigid operators
$\xrightarrow{\mathsf{d}}$	Accessibility in $\mathsf{d}(\mathcal{K}^\star)$
I^d	Interpretation function in $\mathsf{d}(\mathcal{K}^\star)$

Grouned Canonical Model

$\mathsf{w}(\mathsf{d}(\mathcal{K}^\star))$	Canonical model grounded by an interpreted system

Bibliography

[1] M. Abadi, M. Baudet, and B. Warinschi. Guessing attacks and the computational soundness of static equivalence. In L. Aceto and A. Ingólfsdóttir, editors, *Foundations of Software Science and Computation Structures, 9th International Conference (FOSSACS 2006)*, volume 3921 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 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 A. D. Gordon. A calculus for cryptographic protocols: The Spi calculus. *Inf. Comput.*, 148(1):1–70, 1999.

[5] M. Abadi and J. Jürjens. Formal eavesdropping and its computational interpretation. In N. Kobayashi and B. C. Pierce, editors, *Theoretical Aspects of Computer Software, 4th International Symposium (TACS 2001)*, volume 2215 of *Lecture Notes in Computer Science*, pages 82–94. Springer, 2001.

[6] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In J. van Leeuwen, O. Watanabe, M. Hagiya, P. D. Mosses, and T. Ito, editors, *Theoretical Computer Science, Exploring New Frontiers of Theoretical Informatics, International Conference (IFIP TCS 2000)*, volume 1872 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2000.

[7] M. Abadi and M. Tuttle. A semantics for a logic of authentication. In *PODC'91*, pages 201–216, 1991.

[8] M. Abadi and B. Warinschi. Password-based encryption analyzed. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Automata, Languages and Programming, 32nd International Colloquium (ICALP 2005)*, volume 3580 of *Lecture Notes in Computer Science*, pages 664–676. Springer, 2005.

[9] N. Agray, W. van der Hoek, and E. P. de Vink. On ban logics for industrial security protocols. In B. Dunin-Keplicz and E. Nawarecki, editors, *From Theory to Practice in Multi-Agent Systems, Second International Workshop of Central and Eastern Europe on Multi-Agent Systems (CEEMAS 2001)*, volume 2296 of *Lecture Notes in Computer Science*, pages 29–36. Springer, 2001.

[10] F. Belardinelli and A. Lomuscio. A quantified epistemic logic for reasoning about multi-agent systems. In *Proceedings of the 6th International Conference on Autonomous Agents and Multi-Agent systems (AAMAS07)*, pages 115–132. ACM Press, 2007.

[11] P. Bieber. A logic of communication in hostile environments. In *Third IEEE Computer Security Foundations Workshop (CSFW'90)*, pages 14–22. IEEE Computer Society Press, 1990.

[12] P. Bieber and F. Cuppens. A definition of secure dependencies using the logic of security. In *4th IEEE Computer Security Foundations Workshop (CSFW'91)*, pages 2–11, 1991.

[13] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd Annual Symposium on Foundations of Computer Science, 3-5 November 1982, Chicago, Illinois, USA*, pages 112–117. IEEE, 1982.

[14] 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.

[15] J. Bryans, M. Koutny, L. Mazaré, and P. Y. A. Ryan. Opacity generalised to transition systems. In T. Dimitrakos, F. Martinelli, P. Y. A. Ryan, and S. A. Schneider, editors, *Formal Aspects in Security and Trust (FAST 2005)*, volume 3866 of *Lecture Notes in Computer Science*, pages 81–95. Springer, 2005.

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

[17] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981.

[18] J. A. Clark and J. L. Jacob. A survey of authentication protocol literature. Technical Report 1.0, 1997.

[19] M. Cohen and M. Dam. A completeness result for BAN logic. In *2005 International Workshop on Methods for Modalities (M4M-05)*, pages 202–219, 2005.

[20] M. Cohen and M. Dam. Logical omniscience in the semantics of BAN logic. In *Foundations of Computer Security (FCS'05)*, pages 121–132, 2005.

[21] M. Cohen and M. Dam. A complete axiomatization of knowledge and cryptography. In *22th IEEE Symposium on Logic in Computer Science (LICS 2007)*. IEEE Computer Society, 2007. To appear.

[22] G. Corsi. Counterpart semantics. a foundational study on quantified modal logics. Research reprt PP-2002-20, ILLC, 2002.

[23] G. Corsi. A unified completeness theorem for quantified modal logics. *Journal of Symbolic Logic*, 67:1483–1510, 2002.

[24] V. Cortier. Observational equivalence and trace equivalence in an extension of Spi-calculus. Application to cryptographic protocols analysis. Technical Report LSV-02-3, Lab. Specification and Verification, ENS de Cachan, 2002.

[25] V. Cortier, S. Kremer, R. Küsters, and B. Warinschi. Computationally sound symbolic secrecy in the presence of hash functions. In S. Arun-Kumar and N. Garg, editors, *Foundations of Software Technology and Theoretical Computer Science, 26th International Conference (FSTTCS 2006)*, volume 4337 of *Lecture Notes in Computer Science*, pages 176–187. Springer, 2006.

[26] I. M. H. D. M. Gabbay and M. A. ReynoldsR. *Temporal Logic: Mathematical Foundations and Computational Aspects, volume 1*. Clarendon Press, 1994.

[27] A. H. Dekker. C3P0: A tool for automatic sound cryptographic protocol analysis. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop (CSFW'00)*, pages 77–87. IEEE Computer Society Press, 2000.

[28] D. Dolev and A. C.-C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–207, 1983.

[29] R. A. Eberle. A logic of believing, knowing and inferring. *Synthese*, 26:356–382, 1974.

[30] R. Fagin and J. Y. Halpern. Belief, awareness, and limited reasoning. . *Artif. Intell.*, 34(1), 1987.

[31] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[32] C. Fournet and M. Abadi. Mobile values, new names, and secure communication. In *The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, ACM SIGPLAN Notices 36(3)*, pages 104–115, 2001.

[33] T. French, R. van der Meyden, and M. Reynolds. Axioms for logics of knowledge and past time: Synchrony and unique initial states. In R. A. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, editors, *Advances in Modal Logic (AiML)*, pages 53–72. King's College Publications, 2004.

[34] D. Gabbay, V. Shehtman, and D. Skvortsov. Quantification in nonclassical logic. 2006. Manuscript.

[35] P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In R. Alur and D. Peled, editors, *Computer Aided Verification, 16th International Conference, CAV 2004*, volume 3114 of *Lecture Notes in Computer Science*, pages 479–483. Springer, 2004.

[36] J. Garson. Unifying quantified modal logic. *Journal of Philosophical Logic*, 34:621–649, 2005.

[37] J. W. Garson. Quantification in modal logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Volume II*. Reidel, 1984.

[38] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[39] L. Gong, R. M. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In *IEEE Symposium on Security and Privacy*, pages 234–248. IEEE Computer Society Press, 1990.

[40] J. Y. Halpern, Y. Moses, and M. Y. Vardi. Algorithmic knowledge. In R. Fagin, editor, *5th Conference on Theoretical Aspects of Reasoning about Knowledge (TARK)*, pages 255–266. Morgan Kaufmann, 1994.

[41] J. Y. Halpern and K. R. O'Neill. Secrecy in multiagent systems. In *15th IEEE Computer Security Foundations Workshop (CSFW-15 2002)*, pages 32–, 2002.

[42] J. Y. Halpern and K. R. O'Neill. Anonymity and information hiding in multiagent systems. In *16th IEEE Computer Security Foundations Workshop (CSFW-16 2003)*, pages 75–88, 2003.

[43] J. Y. Halpern and R. Pucella. Modeling adversaries in a logic for security protocol analysis. In A. E. Abdallah, P. Ryan, and S. Schneider, editors, *Formal Aspects of Security, First International Conference (FASec 2002)*, volume 2629 of *Lecture Notes in Computer Science*, pages 115–132. Springer, 2002.

[44] J. Y. Halpern and R. Pucella. On the relationship between strand spaces and multi-agent systems. *ACM Trans. Inf. Syst. Secur.*, 6(1):43–70, 2003.

[45] J. Y. Halpern, R. Pucella, and R. van der Meyden. Revisiting the foundations of authentication logics. Manuscript, 2003.

[46] J. Y. Halpern, R. van der Meyden, and M. Y. Vardi. Complete axiomatizations for reasoning about knowledge and time. *SIAM J. Comput.*, 33(3):674–703, 2004.

[47] J. Y. Halpern and M. Y. Vardi. Model checking vs. theorem proving: A manifesto. In *KR*, pages 325–334, 1991.

[48] J. Hintikka. *Knowledge and Belief: An Introduction into the logic of the two notions*. Cornell University Press, Ithaca, 1962.

[49] D. Hughes and V. Shmatikov. Information hiding, anonymity and privacy: a modular approach. *Journal of Computer Security*, 12(1):3–36, 2004.

[50] M. Kacprzak, A. Lomuscio, T. Lasica, W. Penczek, and M. Szreter. Verifying multi-agent systems via unbounded model checking. In M. G. Hinchey, J. L. Rash, W. Truszkowski, and C. Rouff, editors, *Formal Approaches to Agent-Based Systems, Third International Workshop (FAABS 2004)*, volume 3228 of *Lecture Notes in Computer Science*, pages 189–212. Springer, 2004.

[51] R. Kailar. Accountability in electronic commerce protocols. *IEEE Trans. Software Eng.*, 22(5):313–328, 1996.

[52] V. Kessler and G. Wedel. AUTLOG - an advanced logic of authentication. In *Seventh IEEE Computer Security Foundations Workshop (CSFW'94)*, pages 90–99, 1994.

[53] D. Kindred and J. Wing. losing the idealization gap with theory generation. In *Proceedings of the DIMACS Workshop on Cryptographic Protocol Design and Verification*, pages 3–5. Rutgers, May 1997.

[54] K. Konolige. *A Deduction Model of Belief*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1986.

[55] S. Kramer. Logical concepts in cryptography. Cryptology ePrint Archive, Report 2006/262, 2006.

[56] D. Lewis. Counterpart theory and quantified modal logic. *Journal of Philosophy*, 65:113–126, 1968.

[57] H. Liu and M. Li. SVO logic based formalisms of GSI protocols. In K.-M. Liew, H. Shen, S. See, W. Cai, P. Fan, and S. Horiguchi, editors, *Parallel and Distributed Computing: Applications and Technologies, 5th International Conference (PDCAT 2004)*, volume 3320 of *Lecture Notes in Computer Science*, pages 744–747. Springer, 2004.

[58] A. Lomuscio and F. Raimondi. MCMAS: A model checker for multi-agent systems. In H. Hermanns and J. Palsberg, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 12th International Conference (TACAS 2006)*, volume 3920 of *Lecture Notes in Computer Science*, pages 450–454. Springer, 2006.

[59] A. Lomuscio and B. Wozna. A combination of explicit and deductive knowledge with branching time: Completeness and decidability results. In M. Baldoni, U. Endriss, A. Omicini, and P. Torroni, editors, *Declarative Agent Languages and Technologies III, Third International Workshop (DALT 2005)*, volume 3904 of *Lecture Notes in Computer Science*, pages 188–204. Springer, 2005.

[60] A. Lomuscio and B. Wozna. A complete and decidable security-specialised logic and its application to the TESLA protocol. In H. Nakashima, M. P. Wellman, G. Weiss, and P. Stone, editors, *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, pages 145–152. ACM, 2006.

[61] G. Lowe. An attack on the Needham-Schroeder public-key authentication protocol. *Inf. Process. Lett.*, 56(3):131–133, 1995.

[62] Mastercard and VISA. SET Secure Electronic Transaction Specification. 1997.

[63] D. Micciancio and B. Warinschi. Completeness theorems for the Abadi-Rogaway language of encrypted expressions. *Journal of Computer Security*, 12(1):99–130, 2004.

[64] D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. In M. Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference (TCC 2004)*, volume 2951 of *Lecture Notes in Computer Science*, pages 133–151. Springer, 2004.

[65] R. M. Needham and M. D. Schroeder. Using encryption for authentication in large networks of computers. *Commun. ACM*, 21(12):993–999, 1978.

[66] R. Parikh and R. Ramanujam. Distributed processes and the logic of knowledge. In R. Parikh, editor, *Logics of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 256–268. Springer, 1985.

[67] W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. *Fundam. Inform.*, 55(2):167–185, 2003.

[68] R. Pucella. *Reasoning about Resource-Bounded Knowledge: Theory and Application to Security Protocol Analysis*. Ph.D. Thesis, Cornell University, 2004.

[69] R. Pucella. Deductive algorithmic knowledge. *J. Log. Comput.*, 16(2):287–309, 2006.

[70] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1), 1998.

[71] P. Y. A. Ryan and S. A. Schneider. Process algebra and non-interference. *Journal of Computer Security*, 9(1/2):75–103, 2001.

[72] A. Sabelfeld and A. C. Myers. A model for delimited information release. In K. Futatsugi, F. Mizoguchi, and N. Yonezaki, editors, *Software Security - Theories and Systems, Second Mext-NSF-JSPS International Symposium (ISSS 2003)*, volume 3233 of *Lecture Notes in Computer Science*, pages 174–191. Springer, 2003.

[73] A. Sabelfeld and D. Sands. A PER model of secure information flow in sequential programs. *Higher Order Symbol. Comput.*, 14(1):59–91, 2001.

[74] S. G. Stubblebine and R. N. Wright. An authentication logic with formal semantics supporting synchronization, revocation, and recency. *IEEE Trans. Softw. Eng.*, 28(3):256–285, 2002.

[75] P. F. Syverson. Towards a strand semantics for authentication logics. In *Electronic Notes in Theoretical Computer Science*, 20,2000.

[76] P. F. Syverson and S. G. Stubblebine. Group principals and the formalization of anonymity. In J. M. Wing, J. Woodcock, and J. Davies, editors, *FM'99 - Formal Methods, World Congress on Formal Methods in the Development of Computing Systems*, volume 1708 of *Lecture Notes in Computer Science*, pages 814–833. Springer, 1999.

[77] P. F. Syverson and P. C. van Oorschot. A unified cryptographic protocol logic. NRL Publication 5540-227, Naval Research Lab, 1996.

[78] W. Teepe. Proving possession of arbitrary secrets while not giving them away. new protocols and a proof in GNY logic. *Synthese - Knowledge, Rationality and Action*, 49(2):409–443, 2006.

[79] 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.

[80] R. van der Meyden and N. V. Shilov. Model checking knowledge and time in systems with perfect recall (extended abstract). In C. P. Rangan, V. Raman, and R. Ramanujam, editors, *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 1738 of *Lecture Notes in Computer Science*, pages 432–445. Springer, 1999.

[81] R. van der Meyden and K. shu Wong. Complete axiomatizations for reasoning about knowledge and branching time. *Studia Logica*, 75(1):93–123, 2003.

[82] R. van der Meyden and K. Su. Symbolic model checking the knowledge of the dining cryptographers. In *17th IEEE workshop on Computer Security Foundations (CSFW '04)*, page 280, Washington, DC, USA, 2004. IEEE Computer Society.

[83] G. Wedel and V. Kessler. Formal semantics for authentication logics. In E. Bertino, H. Kurth, G. Martella, and E. Montolivo, editors, *Computer Security - ESORICS 96, 4th European Symposium on Research in Computer Security*, volume 1146 of *Lecture Notes in Computer Science*, pages 219–241. Springer, 1996.

[84] M. Wooldridge. Computationally grounded theories of agency. In *4th International Conference on Multi-Agent Systems (ICMAS 2000)*, pages 13–22. IEEE Computer Society, 2000.