Perfect Cryptography, S5 Knowledge, and Algorithmic Knowledge

Sabina Petride
Oracle Corporation
sabina.petride@oracle.com

Riccardo Pucella
Northeastern University
riccardo@ccs.neu.edu

Abstract

We propose a principled approach to model secrecy in multiagent systems, by defining a set of possible observations and providing agents with algorithms used to distinguish the possible states of the system. Our approach fits naturally within a knowledge-based account of secrecy. By adjusting both the kind of observations and the capabilities of the agents, we can capture in a natural way different forms of secrecy in the presence of perfect cryptography. In particular, we show how to model extraction secrecy. Our formalization suggests a unified definition of secrecy for cryptographic protocols and for systems that seek to prevent inadmissible flows of information.

1 Introduction

An important problem facing applications of formal methods to security is to develop general models for specifying and verifying security properties. A good general model enables a comparison and unification of definitions occurring in different frameworks and tools. Consider secrecy in cryptographic protocols as an example. The goal of many protocols is to ensure that a piece of data is transmitted secretly. Other protocols whose goal is to ensure authentication often rely on the secrecy of an authenticating token. Several definitions of secrecy in the cryptographic protocol analysis literature have been proposed, capturing different forms of secrets in the presence of different adversaries.

Our goal is to develop a framework to relate these different definitions of secrecy for cryptographic protocols, and attempt to devise a general, unifying definition. Relating fields with different perspectives on cryptographic protocols and highlighting possible subtleties underlying definitions in specific frameworks are but two benefits of such a unifying definition.

Halpern and O'Neill [12] define a secret to be a fact whose truth an agent cannot establish based only on what she can observe.[1] The Halpern-O'Neill definition of secrecy relies on a direct notion of indistinguishability on states: two states are indistinguishable for an agent if that agent has exactly the same local state at those states. This notion of indistinguishability, however, is not immediately suitable for reasoning about cryptographic protocols. For instance, if an agent does not possess key k, then she should not be able to distinguish message v encrypted with key k (written $\{\!|v|\!\}_k$) from any other message v' encrypted with key k, under any reasonable encryption scheme. However, if we define two states to be indistinguishable to agent i when the local state of i is the same at both states, then any two states where i's local state differ only by having one containing $\{\!|v|\!\}_k$ and the other containing $\{\!|v'|\!\}_k$ (for some other message v') are distinguishable by agent i. Thus, a fact such as "v is the content of the message in the local state" cannot be a secret with respect to that indistinguishability relation; in a sense, it is too strong. Many approaches have been used to solve that problem.

The main contribution of our work is a *principled approach* for deriving an indistinguishability relation, to be used as a foundation for reasoning about secrecy in the presence of encryption. The approach is computational: information might be present in the local state of an agent (e.g., in an encrypted message), but the agent cannot compute that information (e.g., cannot decrypt the message). We define a set of possible observations that agents can make, and supply agents with algorithms that they use to access the information stored in their local states. Two states are

[1] This is a possibilistic definition of secrecy. Although an agent may well learn that a fact has a higher probability of being true than false, we do not consider probabilities in this paper.

indistinguishable if the algorithms return the same information for all observations. By adjusting both observations and algorithms, we can model in a natural way different assumptions on the powers of agents and capture various forms of secrecy described in the cryptographic protocol analysis literature. Putting it differently, our work can be understood as a principled way to apply the Halpern-O'Neill framework for secrecy in the presence of cryptography. As an additional benefit, we inherit a characterization of secrecy in terms of knowledge, giving us a language in which to express facts to be kept secret.

We focus on secrecy in the presence of perfect cryptography, and compare our definition with those used in the symbolic cryptographic protocol analysis literature. (This lets us abstract away from probabilities, as a first stab at the problem.) Symbolic cryptographic protocol analysis is an approach to protocol analysis that treats encryption and decryption as symbolic operations operating on a term algebra of messages—rather than concrete operations on bit strings—and that assumes that agents can compose messages, replay them, or decipher them if they know the right keys, but cannot otherwise "crack" encrypted messages [9]. Symbolic analysis enables automated analysis of protocols, using model checking [8, 19], theorem proving [20], or programming language techniques [1, 6, 21].

As evidence for the suitability of our approach, we relate our general definition of secrecy in the presence of cryptography to existing definitions in the literature of cryptographic protocol analysis. In particular, many approaches to cryptographic protocol analysis, such as CSP-based approaches [18], capture secrecy by restricting the direct computational power of the adversary: a piece of information is secret if an adversary cannot directly compute (or extract) that piece of information from information he has available. We show that under natural assumptions on the kind of information to be kept secret, extraction secrecy can be captured using a particular algorithm and set of observations. In the full version of this paper, we also relate our definition to that of secrecy in the spi calculus [1] and in the setting of information-flow security [16].

The rest of the paper is structured as follows. In §2, we review the Halpern-O'Neill model of secrecy. In §3, we give a general definition of secrecy in multiagent systems that accounts for encrypted messages, and give examples. In §4, we relate our definition of secrecy to a definition based on the ability to extract data from an agent's local state.

2 The Halpern-O'Neill Model of Secrecy

Halpern and O'Neill [12] introduced a framework to study secrecy of partial information in multiagent systems that do not use cryptography, which we review here.

A multiagent system [10] consists of n agents and an environment, each of which is in some local state at a given point in time. An agent's local state encapsulates all the information to which the agent has access (e.g., initial keys, messages sent and received, clock readings). The environment state describes information relevant to the analysis which may not be in any agent's state. We can then view the whole system as being in some global state, formally a tuple $(s_e, s_1, \ldots, s_n)$, where s_i is agent i's state, for $i = 1, \ldots, n$, and s_e is the environment state. For the purpose of this paper, local states of agents are sequences of events $\langle init, e_1, \ldots, e_m \rangle$, where *init* contains data initially available to the agent, and e_i for $i \geq 1$ is an event. The initial data of an agent may include information such as keys and nonces to be used by the agent.

A *run* is a function from time to global states. Intuitively, a run is a complete description of what happens over time in one possible execution of the system. A *point* is a pair (r, m) consisting of a run r and a time m. For simplicity, we take time to range over the natural numbers. At a point (r, m), the system is in some global state $r(m)$. If $r(m) = (s_e, s_1, \ldots, s_n)$, then $r_i(m)$ is s_i, agent i's local state at point (r, m). Formally, a *system* is defined to be a set $\mathcal{R}$ of runs. We define an equivalence relation $\mathcal{K}_i^=$ per agent on points, capturing when an agent has the same local state: $((r, m), (r', m')) \in \mathcal{K}_i^=$ if $r_i(m) = r'_i(m')$. We write $\mathcal{K}_i^=(r, m)$ for $\{(r', m') \mid ((r, m), (r', m')) \in \mathcal{K}_i^=\}$.

This definition of systems is arguably very general. In this work, we generally consider *message-passing systems*—systems where agents send and receive messages—in which events include $\mathsf{send}(i, v)$, sending message v with the intention of being delivered to agent i, and $\mathsf{recv}(v)$, receiving message v. The sender is not included in $\mathsf{recv}(v)$; intuitively, the receiver may not be able to determine the sender of a message she receives. Message-passing systems will generally satisfy a number of conditions on runs, such as every recv event being preceded by a send event. Moreover, systems will generally be derived from a description of a protocol for every agent [10]. None of the results in this paper depend on the details of how systems are generated, or on the conditions satisfied by systems.

Partial information about an agent j can be modeled as a function f that takes the same value in two global states that j cannot distinguish. The intuition, which goes back to Sutherland's [23] work on nondeducibility, is that an agent j maintains secrecy with respect to agent i if, at every point in the system, agent i is unable to rule out any of f's values. A *j-information function* on system $\mathcal{R}$ is a function f from the points of $\mathcal{R}$ to some range that depends only on j's local data: if $r_j(m) = r'_j(m')$ then $f(r,m) = f(r',m')$. For example, a function that assigns to every local state of agent j a message in agent j's possession is a j-information function.

Definition 2.1. [12] *If f is a j-information function, then j* maintains total f-secrecy with respect to i in system $\mathcal{R}$ *if for all points (r,m) of $\mathcal{R}$ and all v in the range of f, we have $\mathcal{K}_i^=(r,m) \cap f^{-1}(v) \neq \varnothing$.*

Halpern and O'Neill examine more specialized forms of secrecy that, for instance, take time into account. For simplicity, we focus on total f-secrecy only in this paper. (The issues we consider here are largely orthogonal to the specific definition of secrecy considered.)

Total f-secrecy can be characterized using a formal definition of knowledge. To reason about the knowledge of agents in multiagent systems, we use the standard interpretation of knowledge in economics and artificial intelligence that goes back to Hintikka [14]. An agent knows a fact φ if φ is true at all the worlds that the agent considers as possible alternatives to the actual world. This can be formalized using a logic of knowledge $\mathcal{L}^K$. Starting with a set Φ_0 of primitive propositions, which we can think of as describing basic facts about the system, such as "the key is k" or "agent 1 sent message x to agent 2", formulas of the logic are formed by closing off under negation, conjunction, and the modal operators $K_1, \ldots, K_n$. Formula $K_i\varphi$ is read as "agent i knows fact φ". As usual, we take $\varphi \vee \psi$ to be an abbreviation for $\neg(\neg\varphi \wedge \neg\psi)$ and $\varphi \Rightarrow \psi$ to be an abbreviation for $\neg\varphi \vee \psi$.

The standard models for this logic are based on the idea of possible worlds and Kripke structures [17]. A Kripke structure is a set of possible worlds along with a *possibility relation* $\mathcal{K}_i$ on the worlds (one per agent), where $(w,v) \in \mathcal{K}_i$ if agent i cannot distinguish world w from world v (so that if w is the actual world, agent i would consider v a possible world). An *interpreted system* $\mathcal{I} = (\mathcal{R}, \pi)$ is a system $\mathcal{R}$ along with an interpretation π that assigns truth values to primitive propositions at global states of $\mathcal{R}$. The points of $\mathcal{I}$ are simply the points of $\mathcal{R}$. An interpretation π on global states induces an interpretation over points of $\mathcal{I}$; simply take $\pi(r,m)$ to be $\pi(r(m))$. An interpreted system $\mathcal{I} = (\mathcal{R}, \pi)$ can be viewed as a Kripke structure by taking the possible worlds to be the points of $\mathcal{R}$, and by taking $\mathcal{K}_i$ to be the equivalence relation $\mathcal{K}_i^=$. Thus, agent i considers a point (r',m') possible at a point (r,m) if agent i has the same local state at both points. The truth of a formula φ at a point (r,m) of interpreted system $\mathcal{I}$ is defined inductively as follows:

- $(\mathcal{I},r,m) \models p$ iff $\pi(r,m)(p) = \mathbf{true}$
- $(\mathcal{I},r,m) \models \neg\varphi$ iff $(\mathcal{I},r,m) \not\models \varphi$
- $(\mathcal{I},r,m) \models \varphi \wedge \psi$ iff $(\mathcal{I},r,m) \models \varphi$ and $(\mathcal{I},r,m) \models \psi$
- $(\mathcal{I},r,m) \models K_i\varphi$ iff $(\mathcal{I},r',m') \models \varphi$ for all $(r',m') \in \mathcal{K}_i^=(r,m)$.

Note the fourth clause, designed to capture the intuition that agent i knows φ exactly if φ is true at all points that i thinks are possible. We write $\mathcal{I} \models \varphi$ if $(\mathcal{I},r,m) \models \varphi$ for all points (r,m) of $\mathcal{I}$.

Total f-secrecy in systems has an intuitive syntactic characterization: j maintains total f-secrecy with respect to i if and only if agent i can never know any fact about the value of f. A fact about the value of f in an interpreted system $\mathcal{I}$ is a formula φ whose truth value at some point (r,m) depends entirely on the value $f(r,m)$, that is, if $f(r,m) = f(r',m')$, then $(\mathcal{I},r,m) \models \varphi$ if and only if $(\mathcal{I},r',m') \models \varphi$. Call such a formula f-local in $\mathcal{I}$.

Proposition 2.2. [12] *In any system $\mathcal{R}$, j maintains total f-secrecy with respect to i if and only if for all interpretations π and all formulas φ that are f-local in $\mathcal{I} = (\mathcal{R},\pi)$, if $\mathcal{I} \not\models \varphi$, then $\mathcal{I} \models \neg K_i\varphi$.*

Thus, secrecy, expressed semantically as "there are enough worlds compatible with an agent's local state", can also be captured by statements such as "an agent never knows fact φ". Expressing secrecy syntactically using knowledge provides us with a precise language for specifying what exactly is meant to be kept secret—doing this using a language is sometimes easier than doing it semantically. Moreover, not only do we get an alternative description of secrecy, we can now bring to bear on the problem the extensive machinery developed to reason about knowledge [10], such as proof systems and verification procedures.

3 Observational Knowledge

We extend the Halpern-O'Neill framework to capture the notion of secrecy in the presence of cryptography, using a computational notion of indistinguishability.

We study secrecy under the assumption that cryptography is perfect, that is, where the only way to decrypt an encrypted message is to know the corresponding decryption key. This assumption lets us reason about

cryptographic protocols independently of the specific properties of the encryption scheme used by the protocol. The standard way to model perfect cryptography is to consider a purely symbolic encryption scheme. A message v is either an arbitrary string s in some given set of plaintexts, a key k, an encryption $\{\!|v_1|\!\}_k$ of message v_1 under key k, or a pair (v_1, v_2) of messages v_1 and v_2. Let $\mathcal{M}$ be the set of all such messages. For simplicity, we assume that encryption is symmetric: the same key is used to encrypt and decrypt messages. (There is no difficulty in extending our work to asymmetric encryption.)

Perfect encryption is further captured by assuming that agents can perform only symbolic manipulation on messages, and not attempt to "crack" encrypted messages. This model, first described rigorously by Dolev and Yao [9], can be formalized by a relation $H \vdash_{\text{DY}} v$ between a set H of messages and a message v. Intuitively, $H \vdash_{\text{DY}} v$ means that an agent can "extract" message v from a set of received messages and keys H, using the allowed operations. The derivation is defined using the following inference rules:

$$\frac{v \in H}{H \vdash_{\text{DY}} v} \qquad \frac{H \vdash_{\text{DY}} \{\!|v|\!\}_k \quad H \vdash_{\text{DY}} k}{H \vdash_{\text{DY}} v}$$
$$\frac{H \vdash_{\text{DY}} (v_1, v_2)}{H \vdash_{\text{DY}} v_1} \qquad \frac{H \vdash_{\text{DY}} (v_1, v_2)}{H \vdash_{\text{DY}} v_2}. \tag{1}$$

For instance, if an agent receives messages $H = \{\{\!|v|\!\}_{k_1}, \{\!|k_1|\!\}_{k_2}, k_2\}$, she can derive v using these inference rules.

Observational Systems. *Indistinguishability* is central to most definitions of secrecy. Roughly speaking, a fact φ is secret relative to agent i if agent i cannot distinguish an execution where φ is true from an execution where φ is false. For instance, early semantics for BAN-like logics [3] take two states to be indistinguishable to an agent if they record the same set of received messages, up to messages encrypted using keys not known to the agent. This approach captures the intuition that an agent cannot distinguish messages encrypted using unknown keys. Many approaches have similar, often more complex, definitions of indistinguishability.

We seek a unified and principled way to devise indistinguishability relations. As we now argue, we can use a computational intuition and consider two local states indistinguishable to an agent if that agent can *compute* the same observations in both states. Observations capture aspects of the state that are relevant, and are represented as a set of facts Θ. To model whether an agent can make a particular observation in a local state, we supply every agent with an algorithm that takes an observation θ and a local state as inputs, and computes whether the agent can make observation θ in the given local state. When given θ, the algorithm returns either "Yes", θ is observed; "No", θ is not observed; or "?", the agent has insufficient resources to determine whether θ is observed. An *observational system* is a tuple $(\mathcal{R}, \mathtt{A}_1, \ldots, \mathtt{A}_n, \Theta_1, \ldots, \Theta_n)$ consisting of a system $\mathcal{R}$ and an algorithm $\mathtt{A}_i$ and set of observations Θ_i for every agent in the system.

In observational system $(\mathcal{R}, \mathtt{A}_1, \ldots, \mathtt{A}_n, \Theta_1, \ldots, \Theta_n)$, two points (r, m) and (r', m') are *observationally indistinguishable to agent i*, written $((r, m), (r', m')) \in \mathcal{K}_{\mathtt{A}_i, \Theta_i}$, if

$$\text{for all } \theta \in \Theta_i,\ \mathtt{A}_i(\theta, r_i(m)) = \mathtt{A}_i(\theta, r'_i(m')).$$

Thus, $((r, m), (r', m')) \in \mathcal{K}_{\mathtt{A}_i, \Theta_i}$ states that agent i cannot distinguish global states $r(m)$ and $r'(m')$ if he only makes observations in Θ_i, subject to his observational capabilities as captured by $\mathtt{A}_i$. It is immediate that $\mathcal{K}_{\mathtt{A}_i, \Theta_i}$ is an equivalence relation on the points of $\mathcal{R}$. In the following, we define $\mathcal{K}_{\mathtt{A}_i, \Theta_i}(r, m)$ to be the set $\{(r', m') \mid ((r, m), (r', m')) \in \mathcal{K}_{\mathtt{A}_i, \Theta_i}\}$ of all points that agent i cannot observationally distinguish from (r, m).

All definitions in §2 generalize in a straightforward way. In particular, given f a j-information function, we can define j maintaining total f-secrecy with respect to i in observational system $(\mathcal{R}, \mathtt{A}_1, \ldots, \mathtt{A}_n, \Theta_1, \ldots, \Theta_n)$, using $\mathcal{K}_{\mathtt{A}_i, \Theta_i}$ for $\mathcal{K}_i^=$ in Definition 2.1.

Total f-secrecy in observational systems can also be characterized in terms of a formal definition of knowledge. This is especially interesting because knowledge and the closely related notion of belief have been used extensively as basic primitives in logics for reasoning about cryptographic protocols [7, 3, 5, 11, 22, 24]. An interpreted observational system $\mathcal{I}$ is just an observational system equipped with an interpretation π for primitive propositions. (We do require that algorithms in an interpreted observational system be *sound* with respect to the interpretation of observations in Θ_i: at every point (r, m) of interpreted observational system $\mathcal{I}$, and for every $\theta \in \Theta_i$, $\mathtt{A}_i(\theta, r_i(m)) =$ "Yes" implies $\pi(r, m)(\theta) = \mathbf{true}$, and $\mathtt{A}_i(\theta, r_i(m)) =$ "No" implies $\pi(r, m)(\theta) = \mathbf{false}$.)

This lets us interpret logic $\mathcal{L}^K$ as in §2, where knowledge is now interpreted using relation $\mathcal{K}_{\mathtt{A}_i, \Theta_i}$. Because $\mathcal{K}_{\mathtt{A}_i, \Theta_i}$ is an equivalence relation, the knowledge operator is an S5 modal operator [15]. In particular, $K_i\varphi \Rightarrow \varphi$ is valid in interpreted observational systems. The exact same proof as Proposition 2.2 gives us that j maintains total f-secrecy with respect to i in $(\mathcal{R}, \mathtt{A}_1, \ldots, \mathtt{A}_n, \Theta_1, \ldots, \Theta_n)$ if and only if for all interpretations π and all formulas φ that are f-local in $\mathcal{I} = (\mathcal{R}, \pi)$, if $\mathcal{I} \not\models \varphi$, then $\mathcal{I} \models \neg K_i\varphi$.

Our working hypothesis is that total f-secrecy, for suitable choices of f, $\mathtt{A}_i$, and Θ_i, captures most flavors of secrecy in the literature on symbolic analysis of cryptographic protocols, as well as the literature on information flow.

Dolev-Yao Indistinguishability. We can easily model common indistinguishability relations in our framework. What is interesting is that these indistinguishability relations are motivated in terms of observations. One of the core indistinguishability relations when reasoning about perfect cryptography, Dolev-Yao indistinguishability, states that an agent cannot distinguish two states in which he has received different messages, where the only difference between these messages occurs in the content of encrypted messages for which he does not have the key. For example, an agent should not be able to distinguish messages $(v, \{\!|v'|\!\}_k)$ and $(v, \{\!|v''|\!\}_k)$ if the agent does not know k, although $(v, \{\!|v'|\!\}_k)$ and $(\{\!|v''|\!\}_k, v)$ should still be distinguishable, because order of messages in tuples is relevant.

For each agent i, we define a set $\Theta^{\text{DY},i}$ of observations that permits us to express queries about the shape of data in the local state of the agent. We have a great deal of flexibility in how we choose this set. Here is one approach that has the advantage of being easily extensible. Observations θ are built according to the following grammar:

$$\begin{aligned} t &::= x \mid s \mid k \mid \{\!|t|\!\}_k \mid (t_1, t_2) \\ \theta &::= msg_i(t) \mid \exists x.\theta. \end{aligned}$$

Thus, terms t look like messages, except that they may contain variables. Let $\Theta^{\text{DY},i}$ be the set of all observations θ given by the grammar above where every variable appearing in θ is bound by an existential quantifier $\exists$, and every bound variable x in $\exists x.\theta'$ is used at most once in θ'. For example, observation $\exists y.msg_i((v, y))$ indicates that there is a message (v, y) in agent i's local state, for some unspecified message y. Note that agent i can only use the msg_i constructor.

The corresponding algorithm $\mathtt{A}^{\text{DY},i}$ is straightforward. Recall that we take local states of an agent to contain all messages sent and received by that agent, as well as keys initially known to the agent. We write $initkey(\ell)$ for the set of keys initially known by an agent with local state ℓ. To compute $\mathtt{A}^{\text{DY},i}(\theta, \ell)$, the algorithm checks if some message received by agent i has the shape prescribed by θ, taking into account the keys known to that agent. Knowledge algorithm $\mathtt{A}^{\text{DY},i}$ is given in Figure 1.

The induced indistinguishability relation $\mathcal{K}_{\mathtt{A}^{\text{DY},i},\Theta^{\text{DY},i}}$ is equivalent to the following well-known formulation, taken from Abadi and Rogaway [2]. Given a message v and a set of keys K, let $[v]_K$ be the result of replacing every indecipherable message in v by special symbol $\Box$. Formally, define:

$$\begin{aligned} [s]_K &= s \\ [k]_K &= k \\ [(v_1, v_2)]_K &= ([v_1]_K, [v_2]_K) \\ [\{\!|v|\!\}_k]_K &= \begin{cases} \{\!|[v]_K|\!\}_k & \text{if } k \in K \\ \Box & \text{otherwise.} \end{cases} \end{aligned}$$

Extend $[-]_K$ to sets of messages H by taking $[H]_K = \{[v]_K \mid v \in H\}$, and define $Keys(H) = \{k \mid H \vdash_{\text{DY}} k\}$. Let $[H]$ stand for $[H]_{Keys(H)}$.

Why are we justified in calling this a form of Dolev-Yao indistinguishability? It is easy to relate this definition to derivation according to the Dolev-Yao rules (1); we can check that whenever v is a plaintext s or a key k, then $H \vdash_{\text{DY}} v$ if and only if v is a subterm of $[H]$.

The following result formally shows that in an observational system in which agent i uses algorithm $\mathtt{A}^{\text{DY},i}$ and observations $\Theta^{\text{DY},i}$, the induced indistinguishability relation $\mathcal{K}_{\mathtt{A}^{\text{DY},i},\Theta^{\text{DY},i}}$ corresponds exactly to the Abadi-Rogaway formulation of equivalence on received messages.

Proposition 3.1. *Let $(\mathcal{R}, \mathtt{A}_1, \ldots, \mathtt{A}_n, \Theta_1, \ldots, \Theta_n)$ be an observational system. If $\mathtt{A}_i = \mathtt{A}^{\text{DY},i}$ and $\Theta_i = \Theta^{\text{DY},i}$, then $((r, m), (r', m')) \in \mathcal{K}_{\mathtt{A}_i,\Theta_i}$ if and only if $[\{v \mid \mathsf{recv}(v) \in r_i(m)\}] = [\{v \mid \mathsf{recv}(v) \in r'_i(m')\}]$.*

Algorithm $\mathtt{A}^{\text{DY},i}$ and observations $\Theta^{\text{DY},i}$ generalize in several ways. A variant is to allow an agent to perform comparisons of encrypted messages even if he is not able to decrypt them. Thus, for example, an agent who does not know key k might be able to distinguish message $(\{\!|v_1|\!\}_k, \{\!|v_1|\!\}_k)$ from $(\{\!|v_1|\!\}_k, \{\!|v_2|\!\}_k)$, but not from $(\{\!|v_2|\!\}_k, \{\!|v_2|\!\}_k)$. This corresponds to the following choice of observations:

$$\theta ::= msg_i(t) \mid \exists x.\theta,$$

where terms are as before; the difference is that we do not require bound variables to appear only once in the body of an existential $\exists x.\theta$. (Of course, this interpretation of observations makes sense only if we assume that encryption is deterministic.)

Another variant is to take length into account [4]. We associate a length with every message $v \in \mathcal{M}$. We add a new primitive observation $length(t, n)$, where t is a term and n is an integer, and take observations to be of the following form:

$$\theta ::= msg_i(t) \mid msg_i(t_1) \wedge length(t_2, n) \mid \exists x.\theta,$$

where terms are as before; we require that bound variables appear at most once in any $msg_i(-)$ (although they can appear unconstrained in any number

$$
\begin{aligned}
\mathtt{A}^{\mathrm{DY},i}(\theta,\ell) &= K \leftarrow \mathit{keysof}(\ell)\\
&\quad \text{if some key in } \theta \text{ is not in } K \text{ then return ``?''}\\
&\quad \text{match } \theta \text{ as } \exists x_1,\ldots,x_n.\mathit{msg}_i(t)\\
&\quad \text{for each } \mathsf{recv}(v)\in\ell \text{ do if } \mathit{match}(t,v) \text{ then return ``Yes''}\\
&\quad \text{return ``No''}\\[1ex]
\mathit{match}(t,v) &= \text{if } t = x \text{ then return true}\\
&\quad \text{if } t = s \text{ and } v = s \text{ then return true}\\
&\quad \text{if } t = k \text{ and } v = k \text{ then return true}\\
&\quad \text{if } t = \{\!|t_1|\!\}_k \text{ and } v = \{\!|v_1|\!\}_k \text{ then return } \mathit{match}(t_1,v_1)\\
&\quad \text{if } t = (t_1,t_2) \text{ and } v = (v_1,v_2) \text{ then return } \mathit{match}(t_1,v_1)\wedge\mathit{match}(t_2,v_2)\\
&\quad \text{return false}\\[1ex]
\mathit{keysof}(\ell) &= K \leftarrow \mathit{initkeys}(\ell)\\
&\quad \text{loop until no change in } K\text{: } K \leftarrow \textstyle\bigcup_{\mathsf{recv}(v)\in\ell}\mathit{getkeys}(v,K)\\
&\quad \text{return } K\\[1ex]
\mathit{getkeys}(v,K) &= \text{if } v\in K \text{ then return } \{v\}\\
&\quad \text{if } v \text{ is } \{\!|v_1|\!\}_k \text{ and } k\in K \text{ then return } \mathit{getkeys}(v_1,K)\\
&\quad \text{if } v \text{ is } (v_1,v_2) \text{ then return } \mathit{getkeys}(v_1,K)\cup\mathit{getkeys}(v_2,K)\\
&\quad \text{return } \{\}
\end{aligned}
$$

Figure 1: Algorithm $\mathtt{A}^{\mathrm{DY},i}$

of *length*$(-)$.) We can thus write observations such as $\exists x.\mathit{msg}_i((v,x))\wedge\mathit{length}(x,n)$, which says that agent i has received a message of the form (v,x) for some message x of length n.

Although we do not do so here, it should be clear that we can impose other restrictions on algorithms, such as restricting them to run in polynomial time, or using limited resources.

4 Relationship with Extraction Secrecy

A popular symbolic approach to dealing with secrecy is to restrict the direct computational power of agents: a piece of information is secret if an agent cannot directly compute that piece of information from information she has available. This definition of secrecy, which we call *extraction secrecy* here, relies on ensuring that at no point of the system can an agent "extract" that piece of information from the messages she has received. For a Dolev-Yao agent, this means ensuring that at no point of the system can he derive $H\vdash_{\mathrm{DY}} v$, where H is the set of messages the adversary has received at that point, and v is the secret message. Extraction secrecy is used, for instance, in CSP-based approaches [18]. In this section, we relate extraction secrecy to observational secrecy.

Extraction secrecy is captured using algorithmic knowledge [13]. Algorithmic knowledge can be added to observational systems simply by augmenting logic $\mathcal{L}^K$ with a new operator $X_i\varphi$, read "agent i algorithmically knows φ", or "agent i can compute φ". We use an agent's algorithm to determine whether that agent algorithmically knows fact φ. (An agent's algorithm is called a *knowledge algorithm* in the context of algorithmic knowledge.) The semantics of formulas in interpreted observational systems is extended by:

$$(\mathcal{I},r,m)\models X_i\varphi \text{ if } \mathtt{A}_i(\varphi,r_i(m)) = \text{``Yes''}.$$

Roughly speaking, K_i captures what an agent implicitly knows given his observations of his local state, while X_i captures what an agent can compute explicitly. It is well known that implicit knowledge suffers from the problem of *logical omniscience*: an agent knows all tautologies (i.e., $K_i\varphi$ is valid if φ is valid) and all the logical consequences of her knowledge (i.e., $K_i\varphi\wedge K_i(\varphi\Rightarrow\psi)\Rightarrow K_i\psi$ is valid). In particular, the definition of knowledge in §3 has this property. The reasoning that allows an agent to infer properties of a system also allows an agent to infer properties that cannot be computed by realistic agents in any reasonable amount of time. In contrast, it can be easily verified that algorithmic knowledge does not suffer from logical omniscience.

What is the relationship between extraction secrecy and observational secrecy? Both approaches, as cap-

tured here, rely on a computational intuition and use knowledge algorithms, but in a different way. Extraction secrecy is obtained by restricting the ability of agents to perform deductions based on what they observe, without focusing on what they can observe. Observational secrecy, on the other hand, captures what agents would know with an arbitrary amount of computation that does not involve looking at messages that they are unable to decrypt, that is, we restrict the observational power of agents but not their power to perform deduction based on what they have observed. The exact relationship and tradeoffs between these two approaches is a largely unexplored topic.

We can immediately elucidate some aspects of the relationship. Because of our soundness assumptions, for observations $\theta \in \Theta$, algorithmic knowledge and knowledge relative to Θ coincide when using the same knowledge algorithm: $(\mathcal{I}, r, m) \models X_i\theta$ if and only if $(\mathcal{I}, r, m) \models K_i\theta$. In many cases, knowledge algorithms are sound with respect to local states indistinguishable relative to Θ; that is, many knowledge algorithms have the property that $\mathtt{A}_i(\varphi, r_i(m)) =$ "Yes" implies $(\mathcal{I}, r, m) \models K_i\varphi$, and $\mathtt{A}_i(\varphi, r_i(m)) =$ "No" implies $(\mathcal{I}, r, m) \models \neg K_i\varphi$. It will not, however, generally be the case that $(\mathcal{I}, r, m) \models K_i\varphi$ implies $(\mathcal{I}, r, m) \models X_i\varphi$, if only because K_i is an S5 operator subject to logical omniscience while X_i does not. For instance, while $K_i\varphi \Rightarrow \varphi$ is valid for every formula φ, it need not be the case that $X_i\varphi \Rightarrow \varphi$ is ever true.

In some cases, it is possible to establish a deeper connection between extraction secrecy and observational secrecy. In particular, it is possible to do so for secrecy in the presence of Dolev-Yao agents: observational secrecy is in general more expressive than extraction secrecy (i.e., can express more properties), and in the special case of message containment, both approaches agree. The remainder of this section aims at making this statement precise. We start by considering how to capture extraction secrecy for a Dolev-Yao agent using algorithmic knowledge [13]. We assume that the language includes a primitive proposition $\mathit{contained}_i(v)$ for every message v, saying that message v is contained within a message that agent i has received. Extraction secrecy will essentially aim at ensuring that $\mathit{contained}_i(v)$ is kept secret from an adversary—an adversary cannot derive v from messages he has received.

We can encode the Dolev-Yao capabilities via a knowledge algorithm $\mathtt{A}^{\mathrm{XDY},i}$ for agent i. Knowledge algorithm $\mathtt{A}^{\mathrm{XDY},i}$ is similar to $\mathtt{A}^{\mathrm{DY},i}$. Intuitively, knowledge algorithm $\mathtt{A}^{\mathrm{XDY},i}$ simply implements a search for the derivation of a message v from the messages that the agent has intercepted and the initial set of keys, using rules (1). The most interesting case in the definition of $\mathtt{A}^{\mathrm{XDY},i}$ is when the formula is $\mathit{contained}_i(v)$. To compute $\mathtt{A}^{\mathrm{XDY},i}(\mathit{contained}_i(v), \ell)$, the algorithm simply checks whether some message received by agent i contains v as a submessage, according to the keys that are known to that agent. Checking whether v is a submessage of v' is performed by a function *submsg*. Knowledge algorithm $\mathtt{A}^{\mathrm{XDY},i}(\mathit{contained}_i(v), \ell)$ is defined in Figure 2. We assume that there is sufficient redundancy in messages so that an agent can detect whether a decryption is successful; this lets us use a function *decrypt* that attempts to decrypt messages. It is easy to show that $\mathtt{A}^{\mathrm{XDY},i}$ correctly captures the Dolev-Yao adversary: $X_i(\mathit{contained}_i(v))$ is true at a state if and only if agent i can derive v from messages he has intercepted using the Dolev-Yao inference rules.

The connection between extraction secrecy and observational secrecy for Dolev-Yao adversaries can be captured as follows.

Proposition 4.1. *Suppose*

$$\mathcal{I} = (\mathcal{R}, \mathtt{A}_1, \ldots, \mathtt{A}_n, \Theta_1, \ldots, \Theta_n, \pi)$$
$$\mathcal{I}' = (\mathcal{R}, \mathtt{A}'_1, \ldots, \mathtt{A}'_n, \Theta'_1, \ldots, \Theta'_n, \pi),$$

*with a common $\mathcal{R}$ and π, with the property that $\pi(r, m)(\mathit{contained}_i(v)) =$ **true** if and only if v is a submessage of some message in $r_i(m)$. If $\mathtt{A}_i = \mathtt{A}^{\mathrm{DY},i}$, $\Theta_i = \Theta^{\mathrm{DY},i}$, and $\mathtt{A}'_i = \mathtt{A}^{\mathrm{XDY},i}$, then for every plaintext s and every point (r, m) of $\mathcal{R}$,*

$$(\mathcal{I}, r, m) \models K_i(\mathit{contained}_i(s)) \text{ if and only if } (\mathcal{I}', r, m) \models X_i(\mathit{contained}_i(s)).$$

This result, which essentially says that as far as secrecy of contained plaintexts is concerned, extraction secrecy and observational secrecy are equivalent, illustrates why both extraction-based and indistinguishability-based approaches for analysis of cryptographic protocols have been used successfully and somewhat interchangeably. Proposition 4.1 generalizes beyond plaintexts. In particular, it remains true for messages v that can be constructed using plaintexts, concatenation, and encryption under known keys. The constructibility assumption is important—if a message is not constructible, say it is $\{\!|v|\!\}_k$ for a key k not known to the agent, then we can easily construct $\mathcal{I}$ and $\mathcal{I}'$ as in the statement of Proposition 4.1 in which $X_i(\mathit{contained}_i(\{\!|v|\!\}_k))$ is true at point (r, m) of $\mathcal{I}'$, but $K_i(\mathit{contained}_i(\{\!|v|\!\}_k))$ is false at point (r, m) of $\mathcal{I}$.

5 Conclusion

We have presented a general framework to capture various notions of secrecy used in the symbolic cryptographic protocol analysis literature in a principled way, in terms of explicit observations managed by an algorithm. We examined the relationship between our definition and extraction secrecy in this paper, showing

$$
\begin{aligned}
\mathtt{A}^{\mathrm{XDY},i}(\mathit{contained}_i(v), \ell) = {} & \text{if } v \in \mathit{initkeys}(\ell) \text{ then return ``Yes''} \\
& K \leftarrow \mathit{keysof}(\ell) \\
& \text{for each } \mathsf{recv}(v') \in \ell \text{ do if } \mathit{submsg}(v, v', K) \text{ then return ``Yes''} \\
& \text{return ``?''} \\
\\
\mathit{submsg}(x, y, K) = {} & \text{if } x = y \text{ then return ``Yes''} \\
& \text{if } z = \mathit{decrypt}(y, k) \text{ for some } k \in K \text{ then return } \mathit{submsg}(x, z, K) \\
& \text{if } z = \mathit{decrypt}(y, k) \text{ for some } k \notin K \text{ then return ``?''} \\
& \text{if } y \text{ is } (z_1, z_2) \text{ then return } \mathit{submsg}(x, z_1, K) \vee \mathit{submsg}(x, z_2, K) \\
& \text{return ``No''}
\end{aligned}
$$

Figure 2: Knowledge algorithm $\mathtt{A}^{\mathrm{XDY},i}$

that they agree in the case of Dolev-Yao capabilities. In the full version of this paper, we further examine the relationship between our definition of secrecy and two other existing definitions in the literature: secrecy as expressed in the spi calculus, and secrecy in the context of information flow. In both cases, we capture an existing definition of secrecy by instantiating our framework appropriately with specific observations and a specific algorithm.

One feature of our framework is that it gives fine control over observational capabilities of agents. Our approach provides a way to generalize the kind of secrecy existing approaches use, by simply changing the algorithms.

References

[1] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, 1999.

[2] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, 2002.

[3] M. Abadi and M. R. Tuttle. A semantics for a logic of authentication. In *Proc. 10th ACM Symposium on Principles of Distributed Computing (PODC'91)*, pages 201–216, 1991.

[4] P. Adão, G. Bana, and A. Scedrov. Computational and information-theoretic soundness and completeness of formal encryption. In *Proc. 18th IEEE Computer Security Foundations Workshop (CSFW'05)*, pages 170–184. IEEE Computer Society Press, 2005.

[5] P. Bieber. A logic of communication in hostile environment. In *Proc. 3rd IEEE Computer Security Foundations Workshop (CSFW'90)*, pages 14–22. IEEE Computer Society Press, 1990.

[6] B. Blanchet. An efficient cryptographic protocol verifier based on Prolog rules. In *Proc. 14th IEEE Computer Security Foundations Workshop (CSFW'01)*, pages 82–96. IEEE Computer Society Press, 2001.

[7] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.

[8] E.M. Clarke, S. Jha, and W. Marrero. Using state space exploration and a natural deduction style message derivation engine to verify security protocols. In *Proc. IFIP Working Conference on Programming Concepts and Methods (PROCOMET)*, 1998.

[9] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[10] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[11] L. Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In *Proc. 1990 IEEE Symposium on Security and Privacy*, pages 234–248. IEEE Computer Society Press, 1990.

[12] J. Y. Halpern and K. O'Neill. Secrecy in multiagent systems. In *Proc. 15th IEEE Computer Security Foundations Workshop (CSFW'02)*, pages 32–46. IEEE Computer Society Press, 2002.

[13] J. Y. Halpern and R. Pucella. Modeling adversaries in a logic for reasoning about security protocols. In *Proc. Workshop on Formal Aspects of Security (FASec'02)*, volume 2629 of *Lecture Notes in Computer Science*, pages 115–132, 2002.

[14] J. Hintikka. *Knowledge and Belief*. Cornell University Press, 1962.

[15] G. Hughes and M. Cresswell. *A New Introduction to Modal Logic*. Routledge, 1996.

[16] D. Hutter and A. Schairer. Possibilistic information flow control in the presence of encrypted communication. In *Proc. 9th European Symposium on Research in Computer Security (ESORICS'04)*, volume 3193 of *Lecture Notes in Computer Science*, pages 209–224. Springer-Verlag, 2004.

[17] S. Kripke. A semantical analysis of modal logic I: normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[18] G. Lowe. Casper: A compiler for the analysis of security protocols. *Journal of Computer Security*, 6:53–84, 1998.

[19] J. Mitchell, M. Mitchell, and U. Stern. Automated analysis of cryptographic protocols using Murφ. In *Proc. 1997 IEEE Symposium on Security and Privacy*, pages 141–151. IEEE Computer Society Press, 1997.

[20] L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1/2):85–128, 1998.

[21] S. Schneider. Verifying authentication protocols in CSP. *IEEE Transactions on Software Engineering*, 24(9):741–758, 1998.

[22] S. Stubblebine and R. Wright. An authentication logic supporting synchronization, revocation, and recency. In *Proc. 3rd ACM Conference on Computer and Communications Security (CCS'96)*. ACM Press, 1996.

[23] D. Sutherland. A model of information. In *Proc. 9th National Computer Security Conference*, pages 175–183, 1986.

[24] P. F. Syverson and P. C. van Oorschot. On unifying some cryptographic protocol logics. In *Proc. 1994 IEEE Symposium on Security and Privacy*, pages 14–28. IEEE Computer Society Press, 1994.