A Conceptual Framework for Secrecy-preserving Reasoning in Knowledge Bases

JIA TAO, Bryn Mawr College
GIORA SLUTZKI, Iowa State University
VASANT HONAVAR, Penn State University

In many applications knowledge bases (KBs) contain confidential or private information (secrets). The KB should be able to use this secret information in its reasoning process but in answering user queries care must be exercised so that secrets are not revealed to unauthorized users. We consider this problem under the Open World Assumption (OWA) in a setting with multiple querying agents $M_1, ..., M_m$ that can pose queries against the KB $\mathcal{K}$ and selectively share answers that they receive from $\mathcal{K}$ with one or more other querying agents. We assume that for each M_i, the KB has a pre-specified set of secrets S_i that need to be protected from M_i. Communication between querying agents is modeled by a communication graph, a directed graph with self-loops. We introduce a general framework and propose an approach to secrecy-preserving query answering based on sound and complete proof systems. The idea is to hide the truthful answer from a querying agent M_i by feigning ignorance without lying, i.e., to provide the answer 'Unknown' to a query q if it needs to be protected. Under the OWA, a querying agent cannot distinguish between the case that q is being protected (for reasons of secrecy) and the case that it cannot be inferred from $\mathcal{K}$. In the pre-query stage we compute a set of envelopes E_1, ..., E_m (restricted to a finite subset of the set of formulae that are entailed by $\mathcal{K}$) so that $S_i \subseteq E_i$ and a query α posed by agent M_i can be answered truthfully whenever $\alpha \notin E_i$ and $\neg\alpha \notin E_i$. After the pre-query stage, the envelope is updated as needed. We illustrate this approach with two simple cases: the Propositional Horn KBs and the Description Logic $\mathcal{AL}$ KBs.

Categories and Subject Descriptors: I.2.3 [**Deduction and Theorem Proving**]: Inference engines; K.6.m [**Miscellaneous**]: Security; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms: Algorithm, Security, Theory

Additional Key Words and Phrases: Multiagents, Secrecy-preserving Reasoning

1. INTRODUCTION

With an increasing reliance on networked knowledge bases in virtually all areas of human endeavor that involve interactions among organizations, e.g., those that provide healthcare (hospitals, pharmacies, insurance providers), governmental agencies (e.g., intelligence, law enforcement, public policy), or independent nations acting on matters of global concern (e.g., counter-terrorism, international finance), the need to share information often has to be balanced against the need to protect sensitive information or *secrets* from unintended disclosure, e.g., due to copyright, privacy, security, or commercial considerations.

In the area of privacy and security in information systems, early work on information protection led to the creation of a multi-level security model for mandatory access

The first two authors were supported in part by NSF grant CNS1116050. Dr. Honavar's work was supported by the National Science Foundation while working at the Foundation. Any opinion, finding, and conclusions contained in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

control (MAC) [Bell and LaPadula 1974c; 1974b]. While MAC solves the problem of an unauthorized user tricking an authorized user into disclosing sensitive data that a discretionary access control (DAC) model may have, it restricts the security granularity at object level. Role-Based Access Control (RBAC) [Osborn et al. 2000] is an alternative approach to both DAC and MAC which provides authorization on operations (rather than objects). The primary focus of this work has been on access control mechanisms that prohibit access to sensitive information. Work on logic-based authorization frameworks [Osborn et al. 2000; di Vimercati et al. 2005; Jajodia et al. 2001] focuses on policy languages that go beyond traditional access control methods to address obligation, provision, and delegation of authorization as a basis for protecting sensitive information in computer systems, databases and networked information systems (see [Bertino et al. 2006] for a survey). Most of the work on policy languages for the web [Bonatti and Olmedilla 2007; Tonti et al. 2003; Bonatti et al. 2006; Weitzner et al. 2005; Kagal et al. 2006; Kagal et al. 2004; Kagal et al. 2003; Godik and (ed.) 2002; Kolovski et al. 2007] focuses on specifying syntax-based restrictions on access to specific resources or operations on the web. More recently, [Halpern and Weissman 2008] have proposed a first order logic based approach to reasoning about policies. The main focus of these models is the control of direct access to sensitive information. Baader et al. [2009] introduced an approach to reasoning with ontologies in the presence of access restrictions on specific axioms. They use lattice labeling of ontology axioms with the express purpose of enabling selected sub-ontologies to be “offered as views to users based on criteria like the user’s access rights, the trust level...”. Instead of computing a sub-ontology of the given ontology for each user, their approach labels every axiom in the ontology using an appropriate labeling lattice. The user’s access to an axiom is then determined by comparing its label with the axiom label. Even though different users may have different access rights, this approach does not address the interaction between users.

The controlled query evaluation (CQE) framework [Sicherman et al. 1983] offers a way to answer database queries without revealing secrets. Biskup et al. [2011; 2008; 2010; 2008; 2012] and the references therein, extensively explored the CQE framework and focused on protecting secrets in databases (including relational and incomplete databases). They use techniques that may rely on lying (i.e., responding to queries with answers that are inconsistent with the knowledge base) in addition to refusing to answer. In the multiagent case, Biskup et al.[2008; 2012] focused on providing solutions to scenarios where an agent may hide confidential parts of its own belief from other negotiating agents. In a remark at the end of Section 3.3, we comment on the relationship between CQE and our framework.

In the computer security literature, [Goguen and Meseguer 1982], [McLean 1992], and [Gray and Syverson 1998] have utilized a notion of non-interference to capture the intuition that an agent at a high security level must be unable to interfere with an agent at a lower security level. Sutherland [1986] introduced no-information-flow relation (later renamed as nondeducibility) to capture the intuition that an agent at a low security level is unable to deduce anything about the state of agents at a higher security level. In a more recent paper, [Halpern and O’Neill 2008] have shown that Sutherland’s notion of nondeducibility is closely related to Shannon’s [1949] probabilistic definition of secrecy in the context of cryptography, and extended the approach of Shannon and Sutherland to specify secrecy requirements in multi-agent systems.

As [Weitzner et al. 2008] have noted, “excessive reliance on secrecy and up-front control over information has yielded policies that fail to meet social needs, as well as technologies that stifle information flow...”. Hence, there is an urgent need for novel approaches to flexible sharing of information. Unlike most approaches to information protection that simply forbid the use of secret information in answering queries, e.g., access control methods in databases [Bell and LaPadula 1974a; Jajodia 1996; Jain and Farkas 2006], access control methods on policy languages [Bertino et al. 2006] and those that focus on selective access to information on the web [Bonatti and Olmedilla 2007; Tonti et al. 2003; Bonatti et al. 2006; Kagal et al. 2006], Bao et al. [2007] initiated a more flexible approach for information sharing under the OWA, using *secrecy-preserving query answering methodologies*, albeit in the restricted setting of a hierarchical KB with a single querying agent. More recently, [Tao et al. 2010] provided an approach to the *secrecy-preserving query answering* (SPQA) problem for instance checking in the DL $\mathcal{EL}$, with a single querying agent.

In a recent paper [Bonatti and Sauro 2013], Bonatti and Sauro provided a confidentiality model for ontologies. This model addresses the question of how to prevent attacks using background knowledge, or using complete knowledge about parts of the KB or its signature. In our approach, a knowledge base attemps to formulize the relevant information of an application domain. As a result, such background information becomes part of the knowledge contained in the knowledge base. Due to the OWA, a querying agent being aware of having complete knowledge of parts of the KB is not expressible. Moreover, we assume that the signature of a KB is publicly available and queries that use symbols not in the signature are illegal.

In this paper we present a general conceptual framework for secrecy-preserving reasoning in a setting with multiple querying agents where the secrets that the KB is obliged to protect can differ from one agent to another; moreover, each agent can selectively share the answers it receives with only some of the other agents. We extend [Tao et al. 2010] and specify the SPQA problem as the problem of constructing a secrecy-preserving reasoner for answering queries. Unlike most access control methods that forbid the use of secret information in answering queries, our approach answers queries, freely using secrets, but then shielding the answer if it may compromise some secrets. As a simple example, if $\alpha \wedge \beta$ (is true and) needs to be protected, for a system to be as informative as possible, only one of the truthful answers to α and β, say α, has to be protected not to disclose $\alpha \wedge \beta$ and β could be truthfully answered to the user. Note that β is derived from the secret $\alpha \wedge \beta$. We stress that our secrecy preserving framework is conceptual and it is not suggested that it can be usefully deployed, as is, in practical situations. We attempt to provide a “logical core” on which all kinds of extra features can be added (issues related to statistical analysis, preferences, may be even cryptography). At this point, these extensions remain in the future research categories.

Given a KB $\mathcal{K}$ and a set of querying agents $\mathcal{M} = \{M_1, M_2, ..., M_m\}$, for each M_i, there is a pre-specified set of secrets S_i that $\mathcal{K}$ needs to protect from M_i. We assume that an agent can selectively share answers that it receives (in response to queries posed by it) from $\mathcal{K}$ with one or more other querying agents. Such communication between querying agents is modeled using a *communication graph*, a directed graph with self-loops (a technicality), in which a node corresponds to a querying agent and an edge from node M_i to M_j denotes the ability of M_i to share with M_j the answers it receives from $\mathcal{K}$ (but not answers shared with it by other querying agents, unless they happen to be also received directly from $\mathcal{K}$; again, a technicality). Under OWA, the answer to a query q posed by an agent M_i against $\mathcal{K}$ can be “Yes” (q can be deduced from $\mathcal{K}$), “No” ($\neg q$ can be deduced from $\mathcal{K}$), or “Unknown” (neither q nor $\neg q$ can be deduced from $\mathcal{K}$). The basic idea is to hide the truthful answer from M_i, when it is necessary to do so by feigning ignorance without lying; i.e., answering “Unknown” whenever the truthful answer would compromise any secret that the KB $\mathcal{K}$ is obliged to protect from *any* of the querying agents in $\mathcal{M}$. Under the OWA, a querying agent cannot distinguish between the following two scenarios: the answer to q (i) is being protected; and (ii) cannot be inferred from $\mathcal{K}$.

One simple solution to the SPQA problem is to maintain a history that, for each agent, logs the sequence of queries and the corresponding answers. When a new query q is posed by an agent M_i, the reasoner tests whether the truthful answer to q together with answers to previous queries that M_i has received directly from the KB $\mathcal{K}$ or indirectly from other querying agents (its predecessors in the communication graph) compromises a secret that $\mathcal{K}$ is obliged to protect against any of the querying agents in $\mathcal{M}$. If it does, M_i receives the answer "Unknown" in response to the query q. Otherwise, q will be truthfully answered. A "Yes" or "No" answer can be shared by M_i with its successors in the communication graph. Because this approach, which we call *lazy evaluation*, requires checking the answer to each query posed by each querying agent against a query history, the time it takes to answer a query degrades with increase in the size of the history over time. Hence, we propose a different approach: we precompute a *secrecy envelope* (or simply *envelope*) $\mathbb{E} = \{E_1, ..., E_m\}$ (restricted to an appropriate finite subset of formulae that are entailed by the KB) such that $S_i \subseteq E_i$ and a query α posed by agent M_i can be answered truthfully whenever $\alpha \notin E_i$ and $\neg\alpha \notin E_i$. The envelope is updated as needed as new queries are answered. It is easy to show that an envelope always exists. The challenge is to construct an envelope that is guaranteed to protect secrets while allowing queries to be answered as informatively as possible (feigning ignorance only when doing so is necessary to protect a secret). This requires constructing an envelope that is as small as possible. Unfortunately, in general, computing the smallest envelope is NP-hard (see Section 4.1). Hence, we settle on computing and maintaining a *tight envelope*, i.e., an envelope that is *minimal* in that no formula can be removed from it without risk of a secret being compromised. When an envelope is finite, a tight envelope can be obtained by checking every formula in the envelope to see whether removing it compromises secrecy and remove it if it does not reveal any secret. In practice, we always build an initial finite partial envelope and update it as needed (see Section 3.2).

To illustrate our framework and approach, we first take a simple example with the KB being specified by the Propositional Horn logic and queries being specified by the facts. We show how to design rules for constructing an envelope for a Propositional Horn Clause KB utilizing forward chaining which is sound and complete for Horn logic w.r.t. the usual semantics of Propositional Logic (see Section 4.1). We then study a more expressive language, the Description Logic $\mathcal{AL}$, which offers atomic concept, the top and bottom concepts, atomic negation, concept constructors conjunction, value restriction, and unqualified existential restriction. For query answering, one would like to have a sound and complete inference system such that given a KB $\mathcal{K}$, a query α can be deduced from $\mathcal{K}$ ($\mathcal{K} \vdash \alpha$) if and only if α semantically follows from $\mathcal{K}$ ($\mathcal{K} \vDash \alpha$). For a language with full negation such as $\mathcal{ALC}$, checking $\mathcal{K} \vDash \alpha$ can be reduced to checking whether $\mathcal{K} \cup \{\neg\alpha\}$ is satisfiable. However, due to syntactic restrictions, this cannot be done in $\mathcal{AL}$. This led us to define a more general semantics that is particularly well-suited to deal with OWA by incorporating the "Unknown" value into the semantics. This facilitates the proofs of the soundness and completeness theorems for our $\mathcal{AL}$ tableau proof system. It turns out that our semantics is a notational variant of a more general approach to Description Logics over lattices, see [Straccia 2006; Borgwardt and Peñaloza 2011]. The idea is that the interpretation $\mathcal{I}$ of each concept name A is a weak 3-partition[1] $A^{\mathcal{I}} = (A_N^{\mathcal{I}}, A_U^{\mathcal{I}}, A_Y^{\mathcal{I}})$, where $A_N^{\mathcal{I}}$ is the collection of all elements (in the domain) which $\mathcal{I}$ specifies as belonging to $\neg A$, $A_Y^{\mathcal{I}}$ is the collection of all elements (in the domain) which $\mathcal{I}$ specifies as belonging to A, and $A_U^{\mathcal{I}}$ consists of all the other elements. Interestingly, this OW-semantics affords the flexibility of interpreting some concepts

[1] It is called a weak 3-partition because we allow a part to be empty. As usual, the union of the three parts is the whole domain and any two parts are disjoint.

(roles) classically, and others using our OW-approach. We show how an envelope can be constructed using our $\mathcal{AL}$ tableau proof system and how secrecy is preserved (see Section 4.2).

The rest of the paper is organized as follows: Section 2 introduces a general framework for SPQA with multiple querying agents under OWA. In Section 2.1 we prove some properties of envelopes. We also explain how envelopes can be used to answer queries and consider an interesting special case of communication graphs (inverted forests). Section 3 deals with applications where the query space is finite and shows how the multiagent secrecy preserving system can be developed in these applications. Section 4 illustrates applications of the framework and results of preceding sections to solve the SPQA problem in the simple cases of Propositional Horn KBs and the Description Logic $\mathcal{AL}$. This choice is motivated by our attempt to make the exposition of the basics of our secrecy-preserving framework more transparant and easy to understand.

2. MULTIAGENT SECRECY-PRESERVING KB - FRAMEWORK

In this section we introduce a general framework for secrecy-preserving reasoning in a multiagent environment. We denote by $\mathcal{L}$ a formal language which we may also view as a set of well-formed formulas. We will leave $\mathcal{L}$ unspecified but shall assume that it is equipped with formal semantics, allowing for the notion of models, and thereby inducing the concept of entailment, denoted by $\vDash_{\mathcal{L}}$ (or just $\vDash$) and defined in the usual fashion: for $\Gamma \subseteq \mathcal{L}$ and $\phi \in \mathcal{L}$, $\Gamma \vDash \phi$ iff every model of Γ is a model of ϕ. Concrete examples of such a formal language are description logics [Baader et al. 2003] and fragments of first order logic [Hodkinson et al. 2000]. We also assume that $\mathcal{L}$ allows for a deductive apparatus in the form of an inference system, denoted by $\vdash_{\mathcal{L}}$ (or just $\vdash$) which is sound and complete with respect to $\vDash$, i.e., for $\Gamma \subseteq \mathcal{L}$ and $\phi \in \mathcal{L}$, $\Gamma \vDash \phi$ iff $\Gamma \vdash \phi$[2]. For $\Gamma \subseteq \mathcal{L}$, we write $\Gamma^+ = \{\alpha \mid \Gamma \vdash_{\mathcal{L}} \alpha\}$ for the *inferential closure* of a set of formulas Γ. Clearly, $\Gamma \subseteq \Gamma^+$ for any $\Gamma \subseteq \mathcal{L}$. We say that Γ is *inferentially closed* if $\Gamma^+ = \Gamma$. Γ is *consistent* if $\Gamma^+ \neq \mathcal{L}$. A formula $\alpha \in \mathcal{L}$ is a *tautology* if $\vDash \alpha$. The set of all tautologies will be denoted by $\mathfrak{T}$.

Definition 2.1. A *knowledge base* (abbreviated, KB) over $\mathcal{L}$ is a triple $\mathcal{K} = \langle K, \mathcal{Q}, \Omega \rangle$ where

— K is a consistent finite subset $K \subseteq \mathcal{L}$.
— $\mathcal{Q}$, the *query space* of $\mathcal{K}$, is a subset: $K^+ \subseteq \mathcal{Q} \subseteq \mathcal{L}$.
— Ω, is the *answer space*. In most cases $\Omega = \{Y, N, U\}$ (for "Yes", "No" and "Unknown", respectively). The "classical" answer space is $\Omega = \{Y, N\}$.

K represents the information that is explicitly stored in $\mathcal{K}$. This should include not only information about specific objects that querying agents might be interested in, but also all the knowledge that may be relevant and needed to formalize a given application domain; moreover, any knowledge that may be required to protect secrecy should also be made available explicitly in the knowledge base. For instance, if individuals and their SSNs are represented in the knowledge base, then the KB must explicitly specify that an SSN uniquely identifies a person. K^+ represents all the information ("knowledge") that the KB $\mathcal{K}$ can infer. In the sequel, we shall refer to both $\mathcal{K}$ and K as a knowledge base (KB). $\mathcal{Q}$ represents the set of all queries that can be "legally" posed against $\mathcal{K}$. We assume that the signature of a KB is publicly available and queries that use symbols not in the signature are illegal. Moreover, we do not insist that $\mathcal{Q} = \mathcal{L}$ which allows one to account for possible restrictions to be imposed on the queries that

[2] We assume, of course, that all proofs in the inference system $\vdash_{\mathcal{L}}$ are finite.

querying agents may pose. Queries and knowledge bases are sometimes expressed in different languages. Yet, queries are answered based on the knowledge that the KB provides and in this paper, we aim at building a fundamental approach that results in a conceptual framework for answering queries while preserving secrecy. With this goal in mind, and for the sake of simplicity, we restrict the queries and the KB formulas to be expressed in the same language.

Let $\mathcal{K} = \langle K, \mathcal{Q}, \Omega \rangle$ be a KB and let $\mathcal{M} = \{M_1, M_2, ..., M_m\}$ be a set of *querying agents* who may pose queries to the KB. For each querying agent M_i there is a corresponding *secrecy set* consisting of non-tautological statements which the KB is supposed to protect against agent M_i. The querying agents may share the answers they obtain from $\mathcal{K}$ with other querying agents. The sharing is constrained by means of a *communication graph* $(\mathcal{M}, \mathcal{E})$ which is a directed graph with self-loops such that an edge $(M_i, M_j) \in \mathcal{E}$ is interpreted to mean that querying agent M_i shares with agent M_j all the non-U answers he receives from the KB [3]. The self-loops are assumed to make each node both a successor and a predecessor of itself.

Definition 2.2. A *secrecy structure on* a KB $\mathcal{K}$ is a triple $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G} \rangle$ where

- $\mathcal{M} = \{M_1, ..., M_m\}$ is a set of querying agents who may pose queries to $\mathcal{K}$,
- $\mathbb{S} = \{S_1, ..., S_m\}$ is a collection of secrecy sets, one for each querying agent, where for all $1 \leq i \leq m$, $S_i \subseteq K^+ \setminus \mathfrak{T}$, and
- $\mathcal{G} = (\mathcal{M}, \mathcal{E})$ is a communication graph.

Given a KB $\mathcal{K} = \langle K, \mathcal{Q}, \Omega \rangle$ and a secrecy structure $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G} \rangle$, let $\mathcal{R} : \mathcal{Q} \times \mathcal{M} \to \Omega$ be a total function. For $M_i \in \mathcal{M}$, define $\mathcal{Q}^i_B = \{\alpha \in \mathcal{Q} \mid \mathcal{R}(\alpha, M_i) = B\}$ for each $B \in \Omega = \{Y, N, U\}$ and $\mathcal{P}^i_B = \bigcup_{j:(M_j, M_i) \in \mathcal{E}} \mathcal{Q}^j_B$ for each $B \in \{Y, N\}$. Since $\mathcal{Q}^i_B$ contains all the B-answers that agent M_i obtained from $\mathcal{K}$ and $\mathcal{P}^i_B$ contains all the B-answers that agent M_i obtained from its predecessors (including itself), it is clear that $\mathcal{Q}^i_B \subseteq \mathcal{P}^i_B$ for $B \in \{Y, N\}$. We say that $\mathcal{R}$ is an *$\mathcal{L}$-reasoner* if $\mathcal{R}$ is *negation consistent* in the sense that $\mathcal{Q}^i_N = \{\neg\alpha \mid \ \alpha \in \mathcal{Q}^i_Y\}$ [4]. The requirement enforces a match between the Y-queries and the N-queries and implies that $\mathcal{Q}^i_U$ is closed under negation: $\neg\mathcal{Q}^i_U = \mathcal{Q}^i_U$. Thus, given a query $q \in \mathcal{Q}$ and a querying agent $M_i \in \mathcal{M}$, $\mathcal{R}$ provides an answer $\mathcal{R}(q, M_i) \in \Omega$ back to M_i. Note that $\mathcal{R}(q, M_i)$ may be different from $\mathcal{R}(q, M_j)$ when $i \neq j$. The set $\mathcal{Q}^i_B$ contains all B-queries that agent M_i obtains from the KB and $\mathcal{P}^i_B$ contains all B-queries that agent M_i obtains from its predecessors. Note that an agent M_i can pass to its successors only answers to queries in $\mathcal{Q}^i_Y$ or $\mathcal{Q}^i_N$, but not in $\mathcal{Q}^i_U$. Passing the answer to a query in $\mathcal{Q}^i_U$ to M_j will not disclose any secrets that the KB is required to protect against M_j. However, if a querying agent M_i gets an "Unknown" answer of a query q from its predecessor $M_j (i \neq j)$ while he gets a "Yes" answer from the KB, M_i would infer that q is protected against M_j, either because the truthful answer to q leads to the conclusion of some secret that needs to be protected against M_j or against one of M_j's sucessors that is not M_i. Choosing not to pass "Unknown" answers is "safer".

The following definition attempts to capture and formalize the whole secrecy framework as discussed above. It specifies conditions that must be satisfied by an $\mathcal{L}$-reasoner for it to be secrecy-preserving.

Definition 2.3. A *multi-agent secrecy-preserving query-answering (MSQ)* system is a triple $\langle \mathcal{K}, \mathcal{S}, \mathcal{R} \rangle$ where

[3] The case when agents are allowed to share query-answers obtained from other agents rather than just answers obtained from the KB can be reduced to the current problem by using the transitive closure of the communication graph.

[4] Note that if the language $\mathcal{L}$ does not have negation, then an $\mathcal{L}$-reasoner is negation consistent by default.

— $\mathcal{K} = \langle K, \mathcal{Q}, \Omega \rangle$ is a KB,
— $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G} \rangle$ is a secrecy structure on $\mathcal{K}$, and
— $\mathcal{R}$ is an $\mathcal{L}$-reasoner satisfying the following properties: for all $1 \leq i \leq m$,

 — **[Yes Property]** $\mathfrak{T} \subseteq \mathcal{Q}_Y^i \subseteq K^+$;
 — **[Closure Property]** $(\mathcal{Q}_Y^i)^+ = \mathcal{Q}_Y^i$;
 — **[Secrecy Property]** $(\mathcal{P}_Y^i)^+ \cap S_i = \emptyset$,

where $K^+ = \{\alpha \mid K \vdash \alpha\}$, $(\mathcal{Q}_Y^i)^+ = \{\alpha \mid \mathcal{Q}_Y^i \vdash \alpha\}$ and $(\mathcal{P}_Y^i)^+ = \{\alpha \mid \mathcal{P}_Y^i \vdash \alpha\}$ are inference closures obtained by the inference system $\vdash$. An $\mathcal{L}$-reasoner satisfying the above properties is termed a *secrecy-preserving reasoner*. We say that $\mathcal{R}$ and $\vdash$ are *associated*.

The Yes Property ensures that every Y-query is provable from K. The Closure Property requires that any consequence of a set of Y-queries that a querying agent obtains from the KB be a Y-query. Finally, the Secrecy Property ensures that any combination of Y-answers that agent M_i obtains from its predecessors does not compromise any secrets that need to be protected against it.

Example 2.4. Two $\mathcal{L}$-reasoners $\mathcal{R}_0$ and $\mathcal{R}_1$ are defined as follows.

$$\mathcal{R}_0(\alpha, M_i) = \begin{cases} Y & \text{if } \alpha \in \mathfrak{T}, \\ N & \text{if } \neg\alpha \in \mathfrak{T}, \\ U & \text{otherwise.} \end{cases} \qquad \mathcal{R}_1(\alpha, M_i) = \begin{cases} Y & \text{if } K \vdash \alpha \wedge \alpha \notin S_i, \\ N & \text{if } K \vdash \neg\alpha \wedge \neg\alpha \notin S_i, \\ U & \text{if } \alpha \in S_i \end{cases}$$

$\mathcal{R}_0$ is a trivial secrecy-preserving reasoner which hides all the information except tautologies. At the other extreme, the $\mathcal{L}$-reasoner $\mathcal{R}_1$ answers truthfully all queries except for $\alpha \in S_i$. It may fail to satisfy the Closure and/or Secrecy Properties and hence is not a secrecy-preserving reasoner.

Thus, a secrecy-preserving reasoner specifies a deductive apparatus, similar to the syntactic notion of a proof system in classical logics. On the other hand, the semantic notion of an envelope which we introduce next is essentially an entailment-blocking mechanism[5].

Definition 2.5. Let $\mathcal{K} = \langle K, \mathcal{Q}, \Omega \rangle$ be a KB and $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G} \rangle$ a secrecy structure on $\mathcal{K}$. A collection $\mathbb{E} = \{E_1, E_2, ..., E_m\}$, where for $1 \leq i \leq m$, $S_i \subseteq E_i \subseteq K^+ \setminus \mathfrak{T}$, is called a *(secrecy) envelope for* $\mathcal{S}$ if the following two properties are satisfied for every $1 \leq i \leq m$:

— **[E1]** for every $\alpha \in E_i$, $K^+ \setminus E_i \nvDash \alpha$;
— **[E2]** for every $\alpha \in S_i$, $\bigcup_{j:(M_j, M_i) \in \mathcal{E}} (K^+ \setminus E_j) \nvDash \alpha$.

The collection $\mathbb{E}$ is called a *weak envelope for* $\mathcal{S}$ if it only satisfies Property E2. A (weak) envelope $\mathbb{E}$ is said to be *tight* if it satisfies an extra minimality property:

— **[TE]** for every $M_i \in \mathcal{M}$ and every $\alpha \in E_i$, there exists an edge $(M_i, M_j) \in \mathcal{E}$ and $\beta \in S_j$ such that $\bigcup_{k:(M_k, M_j) \in \mathcal{E}} (K^+ \setminus E_k) \cup \{\alpha\} \vDash \beta$.

Note that every envelope is a weak envelope. Given an envelope $\mathbb{E} = \{E_1, E_2, ..., E_m\}$, we say that E_i is an *envelope for the secrecy set* S_i. Property E1 requires that no information in the envelope E_i is entailed from $K^+ \setminus E_i$. Property E2 ensures that no combination of query answers obtained from an agent's predecessors entails any secrets to be protected against this agent. Property TE requires that none of the assertions in any of the envelopes in $\mathbb{E}$ can be removed without compromising the overall secrecy (not necessarily of its own secrecy set). Specifically, answering M_i's query

[5] As observed by an astute reviewer.

$\alpha \in E_i$ with Y (instead of U) would allow one of M_i's successors to conclude some of its own secrets using the information passed to it from its own predecessors.

LEMMA 2.6. *If a weak envelope $\mathbb{E} = \{E_1, E_2, ..., E_m\}$ for $\mathcal{S}$ is tight, then $\mathbb{E}$ is a tight envelope.*

PROOF. We need to show that $\mathbb{E}$ satisfies Property E1. Suppose that there is $\alpha \in E_i$ s.t. $K^+ \setminus E_i \vDash \alpha$. Since $\mathbb{E}$ satisfies Property TE, there exist an edge $(M_i, M_j) \in \mathcal{E}$ and $\beta \in S_j$ s.t. $\bigcup_{k:(M_k,M_j)\in\mathcal{E}}(K^+ \setminus E_k) \cup \{\alpha\} \vDash \beta$. Since $K^+ \setminus E_i \vDash \alpha$, we have $\bigcup_{k:(M_k,M_j)\in\mathcal{E}}(K^+ \setminus E_k) \vDash \alpha$, and so $\bigcup_{k:(M_k,M_j)\in\mathcal{E}}(K^+ \setminus E_k) \vDash \beta$. This contradicts the fact that $\mathbb{E}$ satisfies Property E2. Hence, $\mathbb{E}$ satisfies Property E1. It follows that $\mathbb{E}$ is a tight envelope for $\mathcal{S}$. □

Note that secrecy envelopes (as well as tight envelopes) are not unique and, for example, $\mathbb{E} = \{K^+ \setminus \mathfrak{T}, ..., K^+ \setminus \mathfrak{T}\}$ is always a secrecy envelope. The next corollary lists two useful properties of envelopes; part (1) follows from Property E1 and part (2) follows from Property E2.

COROLLARY 2.7. *Let $\mathcal{K} = \langle K, \mathcal{Q}, \Omega\rangle$ be a KB, $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G}\rangle$ a secrecy structure on $\mathcal{K}$, and $\mathbb{E} = \{E_1, E_2, ..., E_m\}$ a secrecy envelope for $\mathcal{S}$. Then (1) for each $M_i \in \mathcal{M}$, $K^+ \setminus E_i \vDash \alpha$ implies $\alpha \in K^+ \setminus E_i$; (2) for each $(M_j, M_i) \in \mathcal{E}$, $S_i \subseteq E_j$. In particular, $S_i \subseteq E_i$.*

For a secrecy structure $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G}\rangle$ on a KB $\mathcal{K} = \langle K, \mathcal{Q}, \Omega\rangle$, define a set of *induced single-agent secrecy structures (projections)*, one for each $M_i \in \mathcal{M}$: $\mathcal{S}_i = \langle\{M_i\}, \{S_i\}, \langle\{M_i\}, \{(M_i, M_i)\}\rangle\rangle$, $1 \le i \le m$. Let E'_i be a secrecy envelope for $\mathcal{S}_i$ (as per Definition 2.5) and for each $1 \le i \le m$, define $E^*_i = \bigcup_{j:(M_i,M_j)\in\mathcal{E}} E'_j$. Even though $\mathbb{E}' = \{E'_1, ..., E'_m\}$ need not be a weak envelope for $\mathcal{S}$, we have the following result which shows that an envelope for a secrecy structure $\mathcal{S}$ can be constructed from the set of its projections, a "structure theorem" of sorts.

THEOREM 2.8. *$\mathbb{E}^* = \{E^*_1, ..., E^*_m\}$ is an envelope for $\mathcal{S}$.*

PROOF. Since for each $1 \le i \le m$, E'_i is a secrecy envelope for $\mathcal{S}_i$, we have $S_i \subseteq E'_i \subseteq K^+ \setminus \mathfrak{T}$. Therefore, $S_i \subseteq E^*_i \subseteq K^+ \setminus \mathfrak{T}$. We need to verify that $\mathbb{E}^*$ satisfies properties E1 and E2.

— **[E1]**: Suppose that for some i and $\alpha \in E^*_i$, $K^+ \setminus E^*_i \vDash \alpha$. Then $\alpha \in E'_j$ for some j with $(M_i, M_j) \in \mathcal{E}$. Since $K^+ \setminus E^*_i \subseteq K^+ \setminus E'_j$, every model of $K^+ \setminus E'_j$ is a model of $K^+ \setminus E^*_i$, and so $K^+ \setminus E'_j \vDash \alpha$. This contradicts the definition of E'_j.

— **[E2]**: Suppose that for some i and $\alpha \in S_i$, we have $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}(K^+ \setminus E^*_j) \vDash \alpha$. This is equivalent to $K^+ \setminus (\bigcap_{j:(M_j,M_i)\in\mathcal{E}} E^*_j) \vDash \alpha$. By the definition of $\mathbb{E}^*$, we have $E'_i \subseteq \bigcap_{j:(M_j,M_i)\in\mathcal{E}} E^*_j$. It follows that $K^+ \setminus (\bigcap_{j:(M_j,M_i)\in\mathcal{E}} E^*_j) \subseteq K^+ \setminus E'_i$, and so every model of $K^+ \setminus E'_i$ is a model of $K^+ \setminus (\bigcap_{j:(M_j,M_i)\in\mathcal{E}} E^*_j)$. This implies that $K^+ \setminus E'_i \vDash \alpha$, contradicting the definition of E'_i. □

Let MSQ1 and MSQ2 be two MSQs with disjoint sets of querying agents over the same language. If communication between agents M_i from MSQ1 and L_j from MSQ2 is needed, the two MSQs can be merged by adding the corresponding edges between their communication graphs and locally recomputing envelopes as per Theorem 2.8. Thus, by performing only local changes, Theorem 2.8 can be used to integrate existing MSQs into one.

We have defined secrecy-preserving reasoners and envelopes. The former concept is purely syntactic and can be used to construct MSQ systems. The latter, as mentioned previously, is an entailment-blocking device. The two notions are equivalent in the sense that given a secrecy-preserving reasoner, there is a natural and rather obvious way to define a corresponding secrecy envelope, and vice versa. This is shown in Theorems 2.9 and 2.10.

THEOREM 2.9. *Let $\langle \mathcal{K}, \mathcal{S}, \mathcal{R} \rangle$ be an MSQ system. Define a set $\mathbb{E}' = \{E'_1, E'_2, ..., E'_m\}$ where $E'_i = K^+ \setminus \mathcal{Q}^i_Y$ $(1 \leq i \leq m)$. Then $\mathbb{E}'$ is a secrecy envelope for $\mathcal{S}$.*

PROOF. Since $\mathcal{R}$ is a secrecy-preserving reasoner, by the Yes Property, $\mathfrak{T} \subseteq \mathcal{Q}^i_Y$ and so $E'_i = K^+ \setminus \mathcal{Q}^i_Y \subseteq K^+ \setminus \mathfrak{T}$; the Secrecy Property implies that $\mathcal{Q}^i_Y \cap S_i = \emptyset$ and hence $S_i \subseteq E'_i$. We next show that $\mathbb{E}'$ satisfies Properties E1 and E2.

— **[E1]**: Suppose that there exist $M_i \in \mathcal{M}$ and $\alpha \in E'_i$ such that $K^+ \setminus E'_i \vDash \alpha$. Since the inference system $\vdash$ is complete w.r.t. $\vDash$, we have $\mathcal{Q}^i_Y = (K^+ \setminus E'_i) \vdash \alpha$. It follows from the Closure Property that $\alpha \in \mathcal{Q}^i_Y$, i.e., $\alpha \in K^+ \setminus E'_i$. This contradicts $\alpha \in E'_i$.
— **[E2]**: Suppose that there exists $\alpha \in S_i$ such that $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}(K^+ \setminus E'_j) \vDash \alpha$. It follows from the definition of E'_i that $\bigcup_{j:(M_j,M_i)\in\mathcal{E}} \mathcal{Q}^j_Y \vDash \alpha$. Since the inference system $\vdash$ is complete w.r.t. $\vDash$, we have $\bigcup_{j:(M_j,M_i)\in\mathcal{E}} \mathcal{Q}^j_Y \vdash \alpha$, i.e., $\mathcal{P}^i_Y \vdash \alpha$. This contradicts the Secrecy Property. □

The following theorem gives the opposite direction: given a KB and an envelope $\mathbb{E}$, a corresponding secrecy-preserving reasoner can be defined and the answer to a query α can be obtained by checking whether α can be deduced from the KB and its membership status w.r.t. $\mathbb{E}$.

THEOREM 2.10. *Let $\mathcal{K} = \langle K, \mathcal{Q}, \Omega \rangle$ be a KB, $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G} \rangle$ a secrecy structure on $\mathcal{K}$ and $\mathbb{E} = \{E_1, E_2, ..., E_m\}$ a secrecy envelope for $\mathcal{S}$. Define a function $\mathcal{R}$:$\mathcal{Q} \times \mathcal{M} \to \Omega$, by*

$$\mathcal{R}(\alpha, M_i) = \begin{cases} Y & \textit{if } \alpha \in K^+ \setminus E_i, \\ N & \textit{if } \neg\alpha \in K^+ \setminus E_i, \\ U & \textit{otherwise.} \end{cases}$$

Then $\mathcal{R}$ is a secrecy-preserving reasoner.

PROOF. By the definition of $\mathcal{R}$, $\mathcal{Q}^i_N = \{\neg\alpha \mid \alpha \in K^+ \setminus E_i\} = \neg\mathcal{Q}^i_Y$, and so $\mathcal{R}$ is negation consistent. We need to show that $\mathcal{R}$ satisfies the three properties.

— Yes Property: We need to show that $\mathfrak{T} \subseteq \mathcal{Q}^i_Y \subseteq K^+$. By definition of $\mathcal{R}$, $\mathcal{Q}^i_Y = \{\alpha \mid \alpha \in K^+ \setminus E_i\} = K^+ \setminus E_i \subseteq K^+$. Since $\vdash$ is complete w.r.t. $\vDash$, $\mathfrak{T} \subseteq K^+$. It follows from the definition of E_i that $E_i \cap \mathfrak{T} = \emptyset$ and so $\mathfrak{T} \subseteq K^+ \setminus E_i = \mathcal{Q}^i_Y$.
— Closure Property: It suffices to show that $(\mathcal{Q}^i_Y)^+ \subseteq \mathcal{Q}^i_Y$. By definition of $\mathcal{R}$, $\mathcal{Q}^i_Y = K^+ \setminus E_i$. Suppose that $\mathcal{Q}^i_Y \vdash \alpha$ and $\alpha \notin \mathcal{Q}^i_Y = K^+ \setminus E_i$. By the soundness of $\vdash$, $\mathcal{Q}^i_Y \vDash \alpha$ and $\alpha \in E_i$, contradicting Property E1.
— Secrecy Property: Suppose that $(\mathcal{P}^i_Y)^+ \cap S_i \neq \emptyset$. Let $\alpha \in S_i$ s.t. $\mathcal{P}^i_Y \vdash \alpha$. Then $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}(K^+ \setminus E_j) \vdash \alpha$ and by the soundness of $\vdash$, we obtain $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}(K^+ \setminus E_j) \vDash \alpha$. This contradicts our assumption that $\mathbb{E}$ is an envelope. □

2.1. Properties of Envelopes

In this section, we prove some general properties of envelopes. When tractability conditions are satisfied, see Section 3, algorithms may be designed to construct envelopes utilizing these properties. Given a KB $\mathcal{K}$ and a secrecy structure $\mathcal{S}$, as indicated in Theorem 2.8 (and the paragraph before it), the basic task is to construct an envelope for a single secrecy set. Our idea is to find a set of proof-disrupting assertions (of secrets) and put these in an envelope. In Section 4, we utilize the normal inference rules (that are native to the underlying language) and look for such disrupting formulas by inverting the inference rules. Next we formalize these ideas.

For a given KB $\mathcal{K} = \langle K, \mathcal{Q}, \Omega \rangle$ and a formula $\alpha \in K^+$, we say that a finite set $\Gamma \subseteq K^+$ is *α-minimal* if $\Gamma \models \alpha$ and for every $\beta \in \Gamma$, $\Gamma \setminus \{\beta\} \not\models \alpha$. Let $\mathcal{F}_\alpha = \{\Gamma \mid \Gamma \text{ is } \alpha\text{-minimal}\}$. If α needs to be protected, then at least one element in each $\Gamma \in \mathcal{F}_\alpha$ has to be protected so that α cannot be entailed. Note that when $\alpha \in \mathfrak{T}$, $\mathcal{F}_\alpha = \{\emptyset\}$, and so α cannot be protected anyway. Denote by ϕ_Γ an arbitrary but fixed element of a given set Γ.

Remark. The finite sets $\Gamma \in \mathcal{F}_\alpha$ and formulas ϕ_Γ are defined here in a non-constructive way. In fact, we are only interested in ϕ_Γ in so far as it is used to disrupt the proofs for α since these ϕ_Γ will be members of an envelope. In Section 4, we illustrate that, given a KB represented in a specific language, it is possible to compute ϕ_Γ. The following theorem indicates a general way for obtaining an envelope for a given secrecy structure.

THEOREM 2.11. *Given a secrecy structure $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G} \rangle$ where $\mathbb{S} = \{S_1, S_2, ..., S_m\}$, for each $1 \leq i \leq m$, define a sequence of sets where $E_i^0 = S_i$ and $E_i^{k+1} = \{\phi_\Gamma \mid \Gamma \in \mathcal{F}_\alpha$ for some $\alpha \in E_i^k\}$. Let $E_i = \bigcup_{k=0}^{\infty} E_i^k$ and $E_i^* = \bigcup_{j:(M_i, M_j) \in \mathcal{E}} E_j$. Then $\mathbb{E}^* = \{E_1^*, ..., E_m^*\}$ is an envelope for $\mathcal{S}$.*

PROOF. For the given secrecy structure $\mathcal{S}$, define a set of induced single-agent secrecy structures, one for each $M_i \in \mathcal{M}$: $\mathcal{S}_i = \langle \{M_i\}, \{S_i\}, \langle \{M_i\}, \{(M_i, M_i)\} \rangle \rangle$, $1 \leq i \leq m$. By Theorem 2.8, it suffices to show that for each $1 \leq i \leq m$, E_i is an envelope for $\mathcal{S}_i$. Suppose that for some $\alpha \in E_i$, $K^+ \setminus E_i \models \alpha$. Then there is a finite set $\Gamma \subseteq K^+ \setminus E_i$ such that $\Gamma \models \alpha$ and Γ is α-minimal. Hence, $\Gamma \in \mathcal{F}_\alpha$. According to the definition of E_i, there exists k such that $\alpha \in E_i^k$. It follows that $\phi_\Gamma \in \Gamma \cap E_i^{k+1}$ and so $\Gamma \cap E_i \neq \emptyset$. This contradicts the fact that $\Gamma \subseteq K^+ \setminus E_i$. Therefore, E_i is an envelope for $\mathcal{S}_i$. □

In principle, once an envelope is available, queries can be answered according to Theorem 2.10 without compromising secrecy, which is the basic goal of solving the SPQA problem. Since we want our reasoner to be as informative as possible, and because deciding a minimum envelope may be NP-hard (see Section 4.1), we aim at computing tight envelopes. In general, given a finite envelope, we could obtain a tight envelope by checking every formula in the envelope to see whether removing it compromises any secrets: keeping it if it does and removing it if it doesn't. When an envelope is infinite, we may not obtain a tight envelope by removing assertions from it one by one.

Given two (weak) envelopes $\mathbb{E} = \{E_1, ..., E_m\}$ and $\mathbb{E}' = \{E_1', ..., E_m'\}$ for $\mathcal{S}$, we say that $\mathbb{E}'$ is a *(weak) sub-envelope of* $\mathbb{E}$, denoted by $\mathbb{E}' \subseteq \mathbb{E}$, if for each $1 \leq i \leq m$, $E_i' \subseteq E_i$. A (weak) sub-envelope $\mathbb{E}'$ of $\mathbb{E}$ is *proper* if there exists $1 \leq i \leq m$ such that $E_i' \subset E_i$. We next show that every envelope contains a tight sub-envelope.

LEMMA 2.12. *Consider a KB $\mathcal{K}$ and a secrecy structure $\mathcal{S}$ on $\mathcal{K}$. For every weak envelope $\mathbb{E} = \{E_1, ..., E_m\}$ for $\mathcal{S}$, if $\mathbb{E}$ does not contain a proper weak sub-envelope, then $\mathbb{E}$ is a tight envelope for $\mathcal{S}$.*

PROOF. Since $\mathbb{E}$ does not contain a proper weak sub-envelope, for any $M_i \in \mathcal{M}$ and $\alpha \in E_i$, there is an edge $(M_i, M_j) \in \mathcal{E}$ and $\beta \in S_j$ s.t. $\bigcup_{k \neq i:(M_k, M_j) \in \mathcal{E}} (K^+ \setminus E_k) \cup (K^+ \setminus (E_i \setminus \{\alpha\})) \models \beta$, i.e., $\bigcup_{k:(M_k, M_j) \in \mathcal{E}} (K^+ \setminus E_k) \cup \{\alpha\} \models \beta$. Therefore, $\mathbb{E}$ satisfies Property TE. By Lemma 2.6, $\mathbb{E}$ is a tight envelope for $\mathcal{S}$. □

Given a KB $\mathcal{K}$ and a secrecy structure $\mathcal{S}$ on $\mathcal{K}$, for every weak envelope $\mathbb{E}$ for $\mathcal{S}$, either it has a sub-envelope $\mathbb{E}'$, or it does not. In the latter case, $\mathbb{E}$ is a tight envelope for $\mathcal{S}$ by Lemma 2.12. Given a weak envelope $\mathbb{E}$ for $\mathcal{S}$, let $\mathbb{E} = \mathbb{E}^0 \supseteq \mathbb{E}^1 \supseteq \cdots \supseteq \mathbb{E}^n \supseteq \cdots$ be a descending chain of weak envelopes for $\mathcal{S}$ where $\mathbb{E}^k = \{E_1^k, ..., E_m^k\}$, $k \geq 0$.

CLAIM 1. *$\mathbb{E}^\infty = \{\bigcap_{k=0}^{\infty} E_1^k, ..., \bigcap_{k=0}^{\infty} E_m^k\}$ is a weak envelope for $\mathcal{S}$.*

PROOF. Suppose that $\mathbb{E}^\infty$ is not a weak envelope for $\mathcal{S}$. Then for some $M_i \in \mathcal{M}$ and some $\alpha \in S_i$, $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}(K^+ \setminus \bigcap_{k=0}^{\infty} E_j^k) \models \alpha$. This means that there is a finite subset $\Phi \subseteq \bigcup_{j:(M_j,M_i)\in\mathcal{E}}(K^+ \setminus \bigcap_{k=0}^{\infty} E_j^k)$ such that $\Phi \models \alpha$. Since Φ is finite, for some (large enough) n, $\Phi \subseteq \bigcup_{j:(M_j,M_i)\in\mathcal{E}}(K^+ \setminus E_j^n)$, implying $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}(K^+ \setminus E_j^n) \models \alpha$. This contradicts the assumption that $\mathbb{E}^n$ is a weak envelope for $\mathcal{S}$. □

A weak envelope $\mathbb{E}$ is *minimal* if it does not contain a proper weak sub-envelope. Let $\mathbb{E}$ be a weak envelope for $\mathcal{S}$ and $\mathcal{C}_\mathbb{E}$ be the collection of all weak sub-envelopes of $\mathbb{E}$. Since the binary relation $\subseteq$ between weak envelopes of a given $\mathbb{E}$ is a partial order on $\mathcal{C}_\mathbb{E}$, by Claim 1 and (the dual of) Zorn's Lemma, $\mathcal{C}_\mathbb{E}$ contains a minimal weak sub-envelope $\mathbb{E}'$. By Lemma 2.12, $\mathbb{E}'$ is a tight envelope for $\mathcal{S}$. Since every envelope is a weak envelope, this implies that every envelope has a tight sub-envelope (for the same secrecy set $\mathcal{S}$).

Depending on the native inference system $\vdash$ for the language, and/or properties of the communication graph, with some appropriate (probably rather strict) tractability assumptions, a strategy could be designed to guide the computation so that the resulting envelope is tight.

In the remainder of this section, we assume that a tight envelope for a single-agent secrecy structure is available (or easily computable). We show that when the communication graph is an inverted forest (with self-loops), a tight envelope for a multi-agent secrecy structure can be constructed in a single bottom-up sweep.

Definition 2.13. Given a formula α and a set D of formulas, for each $\Gamma \in \mathcal{F}_\alpha$, let $\Gamma_D = \Gamma$ if $\Gamma \cap D = \emptyset$ and $\Gamma_D = \Gamma \cap D$ otherwise. Define $\mathcal{F}_{\alpha,D} = \{\Gamma_D \mid \Gamma \in \mathcal{F}_\alpha\}$. For each $\Gamma_D \in \mathcal{F}_{\alpha,D}$, let ϕ_{Γ_D} be an arbitrary but fixed element in Γ_D. Given a secrecy structure $\mathcal{S} = \langle \mathcal{M} = \{M_1, ..., M_m\}, \{S_1, ..., S_m\}, (\mathcal{M}, \mathcal{E})\rangle$ and a set D, define $\mathcal{H}_i[D] = \{\phi_{\Gamma_D} \mid \Gamma_D \in \mathcal{F}_{\alpha,D}$ for some $\alpha \in S_i\}$.

First note that since $\{\alpha\} \in \mathcal{F}_\alpha$, $S_i \subseteq \mathcal{H}_i[D]$. We claim that $\mathcal{H}_i[D]$ is a weak envelope for the induced single-agent secrecy structure $\mathcal{S}_i$. If not, then there is $\alpha \in S_i$ s.t. $K^+ \setminus \mathcal{H}_i[D] \models \alpha$, and hence there is a finite subset $\Gamma \subseteq K^+ \setminus \mathcal{H}_i[D]$ s.t. Γ is α-minimal. By Definition 2.13, $\phi_{\Gamma_D} \in \Gamma \cap \mathcal{H}_i[D]$, implying $\Gamma \cap \mathcal{H}_i[D] \neq \emptyset$. This is contrary to $\Gamma \subseteq K^+ \setminus \mathcal{H}_i[D]$.

Let $\mathcal{K}$ be a KB and $\mathcal{S} = \langle \mathcal{M}, \{S_1, ..., S_m\}, \mathcal{G} = (\mathcal{M}, \mathcal{E})\rangle$ be a secrecy structure on $\mathcal{K}$ where $\mathcal{G}$ is an inverted forest with self-loops, i.e., $\mathcal{G}$ is a DAG such that each node has at most one successor beside itself. A node that has no successor is called a *leaf*. For every node M_i in $\mathcal{G}$, let $\mathcal{S}_i = \langle \{M_i\}, \{S_i\}, \langle \{M_i\}, \{(M_i, M_i)\}\rangle\rangle$ be the induced single-agent secrecy structure. For every leaf node M_i, let E_i be a tight envelope for $\mathcal{S}_i$. For every non-leaf node M_i, in a bottom-up fashion according to $\mathcal{G}$, define a weak envelope $\mathcal{H}_i[E_j]$ for $\mathcal{S}_i$ as per Definition 2.13 where $(M_i, M_j) \in \mathcal{E}$ and E_j is a tight envelope for $\mathcal{S}_j$. By the discussion following Claim 1, $\mathcal{H}_i[E_j]$ contains a tight sub-envelope for $\mathcal{S}_i$ which we denote by E_i. Let $\mathbb{E}_m^* = \{E_1^*, ..., E_m^*\}$ where $E_i^* = E_i \cup E_j$ where $(M_i, M_j) \in \mathcal{E}$ and $i \neq j$.

Before we show that $\mathbb{E}_m^*$ is a tight envelope for $\mathcal{S}$, we first prove an auxiliary lemma which takes a communication graph to be an edge with two self-loops, i.e., $\mathcal{S} = \langle \mathcal{M} = \{M_1, M_2\}, \{S_1, S_2\}, \langle \mathcal{M}, \mathcal{E}\rangle\rangle$ where $\mathcal{E} = \{(M_1, M_1), (M_2, M_2), (M_1, M_2)\}$.

LEMMA 2.14. *$\mathbb{E}_2^* = \{E_1^*, E_2^*\}$ is a tight envelope for $\mathcal{S}$.*

PROOF. By Theorem 2.8, $\mathbb{E}_2^*$ is an envelope for $\mathcal{S}$. It suffices to show that $\mathbb{E}_2^*$ satisfies Property TE.

For M_2: We have to show that for each $\alpha \in E_2^* = E_2$, there is $\beta \in S_2$ s.t. $(K^+ \setminus E_2) \cup \{\alpha\} \models \beta$. This is true because E_2 is a tight envelope for $\mathcal{S}_2$.

For M_1: (i) Consider $\alpha \in E_1^* \cap E_2 = E_2$. Since E_2 is a tight envelope for $\mathcal{S}_2$, there is $\beta \in S_2$ s.t. $(K^+ \setminus E_2) \cup \{\alpha\} \vDash \beta$. Then $(\bigcup_{k:(M_k,M_2)\in\mathcal{E}}(K^+ \setminus E_k^*)) \cup \{\alpha\} = (K^+ \setminus (E_1^* \cap E_2^*)) \cup \{\alpha\} = (K^+ \setminus E_2) \cup \{\alpha\} \vDash \beta$. (ii) Consider $\alpha \in E_1^* \setminus E_2 = E_1 \setminus E_2$. Since E_1 is a tight envelope for $\mathcal{S}_1$, there is $\beta \in S_1$ s.t. $(K^+ \setminus E_1) \cup \{\alpha\} \vDash \beta$ and $K^+ \setminus E_1 \nvDash \beta$. Since $\alpha \in E_1 \subseteq \mathcal{H}_1[E_2]$, by Definition 2.13, there is a finite set $\Gamma \subseteq (K^+ \setminus E_1) \cup \{\alpha\}$ such that $\alpha = \phi_{\Gamma_{E_2}}$, $\Gamma \vDash \beta$ and $\Gamma_{E_2} \in \mathcal{F}_{\beta,E_2}$. If $\Gamma \cap E_2 \neq \emptyset$, then $\alpha = \phi_{\Gamma_{E_2}} \in E_2$ by the construction of $\mathcal{H}_1[E_2]$ according to Definition 2.13, contradicting the assumption that $\alpha \in E_1 \setminus E_2$. Therefore $\Gamma \cap E_2 = \emptyset$ and hence $((K^+ \setminus E_1) \setminus E_2) \cup \{\alpha\} \vDash \beta$. Moreover, $\bigcup_{k:(M_k,M_1)\in\mathcal{E}}(K^+ \setminus E_k^*) = K^+ \setminus E_1^* = (K^+ \setminus E_1) \setminus E_2$. So, the Property TE holds for M_1. □

Now we consider the general case.

THEOREM 2.15. *$\mathbb{E}_m^* = \{E_1^*, ..., E_m^*\}$ is a tight envelope for $\mathcal{S}$.*

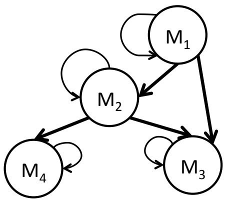
PROOF. By Theorem 2.8, it suffices to show that $\mathbb{E}_m^*$ satisfies Property TE. Suppose, by contradiction, that there are $M_i \in \mathcal{M}$ and $\alpha \in E_i^*$ s.t. for every $(M_i, M_j) \in \mathcal{E}$ and every $\beta \in S_j$, $\bigcup_{k:(M_k,M_j)\in\mathcal{E}}(K^+ \setminus E_k^*) \cup \{\alpha\} \nvDash \beta$. Consider $(M_i, M_j) \in \mathcal{E}$ where $i \neq j$ (when $i = j$ the argument is easy). There are two cases: (1) $E_i^* \setminus E_j \neq \emptyset$. By the proof of Lemma 2.14 (M_1 case (ii)), for every $\alpha' \in E_i^* \setminus E_j$, there is $\gamma \in S_i$ s.t. $(K^+ \setminus E_i^*) \cup \{\alpha'\} \vDash \gamma$. Since $\bigcap_{k:(M_k,M_j)\in\mathcal{E}} E_k^* \subseteq E_i^*$, $(K^+ \setminus \bigcap_{k:(M_k,M_j)\in\mathcal{E}} E_k^*) \cup \{\alpha'\} \vDash \gamma$. (2) $E_i^* \setminus E_j = \emptyset$, i.e., $E_i^* = E_j$. By Lemma 2.14, $\{E_i^*, E_j\}$ is a tight envelope for the induced two-agent secrecy structure: $\mathcal{S}_{ij} = \langle \{M_i, M_j\}, \{S_i, S_j\}, \langle \{M_i, M_j\}, \{(M_i, M_i), (M_j, M_j), (M_i, M_j)\} \rangle$. Hence, for every $\alpha' \in E_i^*$, there is $\gamma \in S_i \cup S_j$ s.t. $(K^+ \setminus E_i^*) \cup \{\alpha'\} \vDash \gamma$ or $(K^+ \setminus E_i^*) \cup (K^+ \setminus E_j) \cup \{\alpha'\} \vDash \gamma$, i.e. $(K^+ \setminus E_i^*) \cup \{\alpha'\} \vDash \gamma$. Hence, $\bigcup_{k:(M_k,M_j)\in\mathcal{E}}(K^+ \setminus E_k^*) \cup \{\alpha'\} \vDash \gamma$. Both cases (1) and (2) yield a contradiction and so Property TE holds. □

Note that in both Lemma 2.14 and Theorem 2.15, for any node M_i, the computation of its envelope E_i is governed by the envelope E_j where $(M_i, M_j) \in \mathcal{E}$ so that as much information as possible in E_j is reused for E_i.

When the communication graph is not an inverted forest, a single bottom-up sweep may not be sufficient to construct a tight envelope, even if the communication graph is a DAG. Here is an example.[6]

Example 2.16. Let α and β be propositional variables and let $\mathcal{K} = \langle K, \mathcal{S}, \mathcal{R} \rangle$ be a propositional KB where: $K = \{\alpha \wedge \beta, \alpha, \beta\}$, $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G} \rangle$, $\mathcal{M} = \{M_1, M_2, M_3, M_4\}$, $\mathbb{S} = \{S_1, S_2, S_3, S_4\}$, $S_1 = S_2 = S_3 = S_4 = \{\alpha \wedge \beta\}$. Suppose that the following choices are made:

(1) $E_3 = \{\alpha \wedge \beta, \alpha\}$ – tight for $\mathcal{S}_3$. $E_3^* = E_3$.
(2) $E_4 = \{\alpha \wedge \beta, \alpha\}$ – tight for $\mathcal{S}_4$. $E_4^* = E_4$.
(3) $E_2 = \{\alpha \wedge \beta, \alpha\}$ – tight for $\mathcal{S}_2$.
(4) $E_2^* = \{\alpha \wedge \beta, \alpha, \beta\}$.
(5) $E_1 = \{\alpha \wedge \beta, \beta\}$ – tight for $\mathcal{S}_1$.
(6) $E_1^* = \{\alpha \wedge \beta, \alpha, \beta\}$.



Note that $\{E_2^*, E_3^*, E_4^*\}$ is a tight envelope for the subgraph induced by the nodes $\{M_2, M_3, M_4\}$. However, the Property TE is not satisfied for M_1 because $\beta \in E_1^*$ is redundant: since $\{\alpha \wedge \beta, \alpha\} \nsubseteq \bigcup_{k:(M_k,M_i)\in\mathcal{E}}(K^+ \setminus E_k^*)$ where $i \in \{1, 2, 3\}$, we have

(1) For M_1's successor M_2, $\bigcup_{k:(M_k,M_2)\in\mathcal{E}}(K^+ \setminus E_k^*) \cup \{\beta\} \nvDash \alpha \wedge \beta \in S_2$.

[6] When constructing and writing down particular envelopes in examples, we will list only a (maximal) subset of mutually inequivalent formulas. For instance, if $\alpha \wedge \beta$ is in a set, then $\beta \wedge \alpha$ is not. In fact, we will call these subsets envelope as well.

(2) For M_1's successor M_3, $\bigcup_{k:(M_k,M_3)\in\mathcal{E}}(K^+ \setminus E_k^*) \cup \{\beta\} \nvDash \alpha \wedge \beta \in S_3$.
(3) For M_1 itself, $\bigcup_{k:(M_k,M_1)\in\mathcal{E}}(K^+ \setminus E_k^*) \cup \{\beta\} \nvDash \alpha \wedge \beta \in S_1$.

Hence, $\mathbb{E}^* = \{E_1^*, E_2^*, E_3^*, E_4^*\}$ is not a tight envelope for $\mathcal{S}$. □

3. MSQ SYSTEMS IN PRACTICE

Let $\mathcal{K} = \langle K, \mathcal{Q}, \Omega\rangle$ be a KB and $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G}\rangle$ a secrecy structure on $\mathcal{K}$ where $\mathcal{G} = (\mathcal{M}, \mathcal{E})$ is the communication graph, see Section 2. We have defined an associated *$\mathcal{L}$-reasoner* as a function $\mathcal{R} : \mathcal{Q} \times \mathcal{M} \to \Omega$ and have specified what it means for such a function to be a secrecy-preserving reasoner, see Definition 2.3. In Section 2, we have also shown that secrecy-preserving reasoners and secrecy envelopes are equivalent concepts, in the sense of the statements of Theorems 2.9 and 2.10.

In most applications we expect that neither the reasoners nor the envelopes are "finished" entities, but rather are constructed piecemeal. Starting at the pre-query stage with a finite reasoner or a finite envelope, they evolve as the queries are presented, depending on secrecy considerations, but are kept finite at all times. In this section, we pursue this idea further.

3.1. A Simple MSQ Algorithm - Lazy Evaluation

For a finite subset $\mathcal{D} \subseteq \mathcal{Q} \times \mathcal{M}$ and a (finite) function $\rho : \mathcal{D} \to \Omega$, analogously to the notation used in Section 2, we define the sets of queries: $\mathcal{Q}_{i,B}^{\rho} = \{\alpha \in \mathcal{Q} \mid \rho(\alpha, M_i) = B\}$ with $B \in \Omega$, and $\mathcal{P}_{i,B}^{\rho} = \bigcup_{j:(M_i,M_j)\in A} \mathcal{Q}_{j,B}^{\rho}$ for $B \in \{Y, N\}$. We require that ρ be *negation consistent*, i.e., $\mathcal{Q}_{i,N}^{\rho} = \{\neg\alpha \mid \ \alpha \in \mathcal{Q}_{i,Y}^{\rho}\}$. The function ρ will be called a *$\mathcal{D}$-answer* (or just an *answer*). Thus ρ represents answers (perhaps, to be) given to queries in $\mathcal{D}$. In the pre-query stage $\mathcal{D} = \emptyset$ and we initialize the answer by: $Init(\alpha, M_i) = U$ iff for some j, $M_j \in \mathcal{M}$ with $(M_i, M_j) \in \mathcal{E}$ and $\alpha \in S_j$. For any $\mathcal{D} \neq \emptyset$, any $\mathcal{D}$-answer is an extension of $Init$.

We say that $\rho : \mathcal{D} \to \Omega$ is *secrecy-preserving (w.r.t. $\mathcal{D}$)* if it satisfies the following properties:

— **[Yes Property]** $\mathcal{Q}_{i,Y}^{\rho} \subseteq K^+$;
— **[Closure Property]** $(\mathcal{Q}_{i,Y}^{\rho})^+ \cap (\mathcal{Q}_{i,N}^{\rho} \cup \mathcal{Q}_{i,U}^{\rho}) = \emptyset$;
— **[Secrecy Property]** $(\mathcal{P}_{i,Y}^{\rho})^+ \cap S_i = \emptyset$,

Recall that for any set of assertions Γ, $\Gamma^+ = \{\alpha \mid \Gamma \vdash \alpha\}$ where $\vdash$ denotes inference in the deductive apparatus associated with the formal language $\mathcal{L}$, see Section 2. Observe also that the initial answer $Init$ is secrecy-preserving w.r.t. $\emptyset$.

Now suppose that a $\mathcal{D}$-answer $\rho : \mathcal{D} \to \Omega$ is given (together with the corresponding query sets $\mathcal{Q}_{i,B}^{\rho}$, $B \in \Omega$, and $P_{i,B}^{\rho}$, $B \in \{Y, N\}$) and a query (α, M_i) comes along. Denote by $\rho' : \mathcal{D}' \to \Omega$ be the new resulting answer. If $(\alpha, M_i) \in \mathcal{D}$, then $\mathcal{D}' = \mathcal{D}$ and $\rho' = \rho$. Otherwise, we have to extend $\mathcal{D}$ by adding (α, M_i) and $(\neg\alpha, M_i)$, and update the answer function: For each $(\alpha, M_i) \in \mathcal{D}$, $\rho'(\alpha, M_i) := \rho(\alpha, M_i)$; otherwise, $\rho'(\alpha, M_i)$, along with the relevant query sets, is computed as indicated in Algorithm 1.

LEMMA 3.1. *Given a knowledge base $\mathcal{K} = \langle K, \mathcal{Q}, \Omega\rangle$, a secrecy structure $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G}\rangle$, a finite subset $\mathcal{D} \subseteq \mathcal{Q} \times \mathcal{M}$, and a secrecy-preserving $\mathcal{D}$-answer ρ, let ρ' be the new answer resulting from Algorithm 1 with the input (α, M_i). Then ρ' is also secrecy-preserving.*

PROOF. Let $\mathcal{D}' = \mathcal{D} \cup \{(\alpha, M_i)\}$. We need to show that ρ' is negation consistent and that it satisfies the three secrecy-preserving properties given above. Since ρ is secrecy-

```
input : $\alpha \in \mathcal{Q}$, $M_i \in \mathcal{M}$
1  if $(\alpha, M_i) \notin \mathcal{D}$ then
2      $\mathcal{D} := \mathcal{D} \cup \{(\alpha, M_i), (\neg\alpha, M_i)\}$
3      if $K \nvdash \alpha$ and $K \nvdash \neg\alpha$ then
4          $\mathcal{Q}^{\rho}_{i,U} := \mathcal{Q}^{\rho}_{i,U} \cup \{\alpha, \neg\alpha\}$ and $\rho(\alpha, M_i) := U$
5      else
6          let $\bar{\alpha} \in \{\alpha, \neg\alpha\}$ such that $K \vdash \bar{\alpha}$
7          if there exists $j$ where $(M_i, M_j) \in \mathcal{E}$ and $\beta \in S_j$ s.t. $\mathcal{P}^{\rho}_{j,Y} \cup \{\bar{\alpha}\} \vdash \beta$ then
8              $\mathcal{Q}^{\rho}_{i,U} := \mathcal{Q}^{\rho}_{i,U} \cup \{\alpha, \neg\alpha\}$
9          else
10             $\mathcal{Q}^{\rho}_{i,Y} := \mathcal{Q}^{\rho}_{i,Y} \cup \{\bar{\alpha}\}$ and $\rho(\bar{\alpha}, M_i) := Y$
11             $\mathcal{Q}^{\rho}_{i,N} := \mathcal{Q}^{\rho}_{i,N} \cup \{\neg\bar{\alpha}\}$ and $\rho(\neg\bar{\alpha}, M_i) := N$
12             forall the $j$ where $(M_i, M_j) \in \mathcal{E}$ do
13                 $\mathcal{P}^{\rho}_{j,Y} := \mathcal{P}^{\rho}_{j,Y} \cup \{\bar{\alpha}\}$ and $\mathcal{P}^{\rho}_{j,N} := \mathcal{P}^{\rho}_{j,N} \cup \{\neg\bar{\alpha}\}$
14             end
15         end
16     end
17 end
18 return $\rho(\alpha, M_i)$ to $M_i$
```

Algorithm 1: Lazy Evaluation $Lazy(\alpha, M_i)$

preserving, it follows from Lines 6, 10 and 11 that ρ' is negation consistent and that the Yes Property is satisfied.

We next show that the Secrecy Property is satisfied. Since ρ is secrecy-preserving, we have $(\mathcal{P}^{\rho}_{i,Y})^+ \cap S_i = \emptyset$, and so $\mathcal{P}^{\rho}_{i,Y} \cap S_i = \emptyset$. If $\alpha \notin \mathcal{P}^{\rho'}_{i,Y}$, then $\mathcal{P}^{\rho'}_{i,Y} = \mathcal{P}^{\rho}_{i,Y}$ and the Secrecy Property is satisfied. Otherwise, α was added to $\mathcal{P}^{\rho'}_{i,Y}$ in Line 13. Therefore, the condition in Line 7 was not satisfied. It follows that for every j where $(M_i, M_j) \in \mathcal{E}$ and every $\beta \in S_j$, $\mathcal{P}^{\rho'}_{j,Y} = \mathcal{P}^{\rho}_{j,Y} \cup \{\alpha\} \nvdash \beta$. In particular, for every $\beta \in S_i$, $\mathcal{P}^{\rho'}_{i,Y} \nvdash \beta$. Hence, $(\mathcal{P}^{\rho'}_{i,Y})^+ \cap S_i = \emptyset$.

To show that ρ' satisfies the Closure Property, we assume that $\mathcal{Q}^{\rho'}_{i,Y} \vdash \alpha$. There are three cases:

- If $\alpha \in \mathcal{Q}^{\rho}_{i,Y}$, then $(\alpha, M_i) \in \mathcal{D}$, $\mathcal{D}' = \mathcal{D}$ and $\rho' = \rho$. Since ρ is secrecy-preserving, so is ρ'.
- If $\alpha \in \mathcal{Q}^{\rho'}_{i,Y} \setminus \mathcal{Q}^{\rho}_{i,Y}$, then α was added to $\mathcal{Q}^{\rho'}_{i,Y}$ in Line 10. From Lines 6-11, we see that $\alpha \notin \mathcal{Q}^{\rho'}_{i,N}$ and $\alpha \notin \mathcal{Q}^{\rho'}_{i,U}$.
- If $\alpha \in (\mathcal{Q}^{\rho'}_{i,Y})^+ \setminus \mathcal{Q}^{\rho'}_{i,Y}$, then we have $\alpha \in \mathcal{Q}^{\rho'}_{i,U} \cup \mathcal{Q}^{\rho'}_{i,N}$. However, since $K \vdash \gamma$ for every $\gamma \in \mathcal{Q}^{\rho'}_{i,Y}$, $(\mathcal{Q}^{\rho'}_{i,Y})^+ \subseteq K^+$. In particular, $\alpha \in K^+$, i.e., $K \vdash \alpha$. Therefore, from Lines 6-11, $\alpha \notin \mathcal{Q}^{\rho'}_{i,N}$ and so $\alpha \in \mathcal{Q}^{\rho'}_{i,U}$, which means that the condition in Line 7 was satisfied: there exists j where $(M_i, M_j) \in \mathcal{E}$ and $\beta \in S_j$ s.t. $\mathcal{P}^{\rho'}_{j,Y} \cup \{\alpha\} \vdash \beta$. However, since $\mathcal{Q}^{\rho'}_{i,Y} \vdash \alpha$ and $\mathcal{Q}^{\rho'}_{i,Y} \subseteq \mathcal{P}^{\rho'}_{j,Y}$ (see Lines 10-14), we have $\mathcal{P}^{\rho'}_{j,Y} \vdash \beta$, contradicting the Secrecy Property for ρ'. Hence, $\alpha \notin \mathcal{Q}^{\rho'}_{i,U}$.

It follows that the Closure Property is satisfied. □

Note that in Algorithm 1, $\mathcal{Q}^{\rho}_{i,Y}$, $\mathcal{Q}^{\rho}_{i,N}$ and $\mathcal{Q}^{\rho}_{i,U}$ are disjoint sets at all times: when $\mathcal{D} \subseteq \mathcal{Q} \times \mathcal{M}$ gets larger, these sets also get larger, but never overlap. Moreover, since (i) $\mathcal{S}$ is assumed to contain no tautologies, (ii) the inference system $\vdash$ is complete, and (iii) for any $\beta \in S_i$, $\mathcal{P}^{\rho}_{i,Y} \nvDash \beta$ (by the Secrecy Property), if α is a tautology, then $\alpha \in \mathcal{Q}^{\rho}_{i,Y}$ (see Lines 6, 7 and 10). Therefore, when $\mathcal{D}$ is really large, the $\mathcal{D}$-answer ρ approximates the secrecy-preserving $\mathcal{L}$-reasoner $\mathcal{R}$ given in Definition 2.3.

3.2. Envelope Maintenance

The lazy evaluation approach is rather simple, but as the number of queries increases, the sets $\mathcal{P}^{\rho}_{i,Y}$ get larger and checking condition in Line 7 in Algorithm 1 takes longer time. Thus, answering queries will tend to be more time consuming as the KB continues to operate. In this section, we propose an alternative solution to the SPQA problem that involves precomputing finite parts of an envelope.

Definition 3.2. Let $\mathcal{K} = \langle K, \mathcal{Q}, \Omega \rangle$ be a KB and $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G} \rangle$ a secrecy structure on $\mathcal{K}$. Given a finite set $\mathcal{Q}'$ such that $K \cup \bigcup_{i=1}^{m} S_i \subseteq \mathcal{Q}' \subseteq \mathcal{Q}$, a collection $\mathbb{E} = \{E_1, E_2, ..., E_m\}$, where for $1 \leq i \leq m$, $S_i \subseteq E_i \subseteq (\mathcal{Q}' \cap K^+) \setminus \mathfrak{T}$, is called a *(partial) envelope for $\mathcal{S}$ w.r.t.* $\mathcal{Q}'$ if the following two properties are satisfied for every $1 \leq i \leq m$:

— **[E1']** for every $\alpha \in E_i$, $(K^+ \cap \mathcal{Q}') \setminus E_i \nvDash \alpha$;
— **[E2']** for every $\alpha \in S_i$, $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}((K^+ \cap \mathcal{Q}') \setminus E_j) \nvDash \alpha$.

$\mathbb{E}$ is said to be *tight* w.r.t. $\mathcal{Q}'$ if it satisfies an extra minimality property:

— **[TE']** for every $M_i \in \mathcal{M}$ and every $\alpha \in E_i$, there exists an edge $(M_i, M_j) \in \mathcal{E}$ and $\beta \in S_j$ such that $\bigcup_{k:(M_k,M_j)\in\mathcal{E}}((K^+ \cap \mathcal{Q}') \setminus E_k) \cup \{\alpha\} \vDash \beta$.

Recall that given a KB $\mathcal{K} = \langle K, \mathcal{Q}, \Omega \rangle$ and a formula $\alpha \in K^+$, a finite set $\Gamma \subseteq K^+$ is α-*minimal* if $\Gamma \vDash \alpha$ and for every $\beta \in \Gamma$, $\Gamma \setminus \{\beta\} \nvDash \alpha$, and that we have defined $\mathcal{F}_\alpha = \{\Gamma \mid \Gamma$ is α-minimal$\}$. Given a finite set of formulas $\mathcal{Q}'$ such that $K \cup \bigcup_{i=1}^{m} S_i \subseteq \mathcal{Q}' \subseteq \mathcal{Q}$, let $\mathcal{F}_\alpha\big|_{\mathcal{Q}'} = \{\Gamma \in \mathcal{F}_\alpha \mid \Gamma \subseteq \mathcal{Q}'\}$. Note that $\mathcal{F}_\alpha\big|_{\mathcal{Q}'}$ is finite. Algorithm 2 provides an approach for obtaining a partial envelope w.r.t. $\mathcal{Q}'$. Lemma 3.3 shows that Algorithm 2 is correct.

```
input  : $K$, $\mathcal{S} = \langle \mathcal{M}, \mathbb{S} = \{S_1, ..., S_m\}, \mathcal{G} \rangle$, $\mathcal{Q}'$
for $1 \leq i \leq m$ do
    $E_i := S_i$
    while there exist $\gamma \in E_i$ and $\Gamma \in \mathcal{F}_\gamma\big|_{\mathcal{Q}'}$ such that $\Gamma \cap E_i = \emptyset$ do
        Let $\phi_\Gamma$ be an arbitrary but fixed element of $\Gamma$
        $E_i := E_i \cup \{\phi_\Gamma\}$
    end
end
for $1 \leq i \leq m$ do
    $E'_i := \bigcup_{j:(M_i,M_j)\in\mathcal{E}} E_j$
end
return $E_{\mathcal{Q}'} = \{E'_1, ..., E'_m\}$
```

Algorithm 2: Partial Envelope Computation

LEMMA 3.3. *Given a KB $\mathcal{K}$, a secrecy structure $\mathcal{S}$ and a finite set of formulas $\mathcal{Q}'$ such that $K \cup \bigcup_{i=1}^{m} S_i \subseteq \mathcal{Q}' \subseteq \mathcal{Q}$, Algorithm 2 computes a partial envelope for $\mathcal{S}$ w.r.t. $\mathcal{Q}'$.*

PROOF. From Lines 2 and 9, it is clear that $S_i \subseteq E'_i$. Since $\mathcal{S}$ does not contain tautologies, by the definition of $\mathcal{F}_\alpha\big|_{\mathcal{Q}'}$, $E_{\mathcal{Q}'}$ does not contain any tautology. Moreover, $E'_i \subseteq (\mathcal{Q}' \cap K^+) \setminus \mathfrak{T}$.

We first show that for a given i, after Line 7, E_i is an envelope w.r.t. $\mathcal{Q}'$ for the induced single-agent secrecy structure $\mathcal{S}_i$. Suppose that for some $\alpha \in E_i$, $(K^+ \cap \mathcal{Q}') \setminus E_i \models \alpha$. Then there is a finite set $\Gamma \subseteq (K^+ \cap \mathcal{Q}') \setminus E_i$ such that $\Gamma \models \alpha$ and Γ is α-minimal. Hence, $\Gamma \in \mathcal{F}_\alpha\big|_{\mathcal{Q}'}$. Since $\Gamma \cap E_i = \emptyset$, the condition in Line 3 is satisfied, and so there exists $\phi_\Gamma \in \Gamma$ such that $\phi_\Gamma \in E_i$. This contradicts the assumption $\Gamma \cap E_i = \emptyset$ and implies that E_i is an envelope for $\mathcal{S}_i$ w.r.t. $\mathcal{Q}'$.

Next we show that $E_{\mathcal{Q}'} = \{E'_1, ..., E'_m\}$ is an envelope for $\mathcal{S}$ w.r.t. $\mathcal{Q}'$. We need to verify that $E_{\mathcal{Q}'}$ satisfies Properties E1' and E2'.

— **[E1']**: Suppose that for some i and $\alpha \in E'_i$, $(K^+ \cap \mathcal{Q}') \setminus E'_i \models \alpha$. Then $\alpha \in E_j$ for some j with $(M_i, M_j) \in \mathcal{E}$. Since $(K^+ \cap \mathcal{Q}') \setminus E'_i \subseteq (K^+ \cap \mathcal{Q}') \setminus E_j$, every model of $(K^+ \cap \mathcal{Q}') \setminus E_j$ is a model of $(K^+ \cap \mathcal{Q}') \setminus E'_i$, and so $(K^+ \cap \mathcal{Q}') \setminus E_j \models \alpha$. This contradicts the fact shown above that E_j is an envelope for $\mathcal{S}_j$ w.r.t. $\mathcal{Q}'$.

— **[E2']**: Suppose that for some i and $\alpha \in S_i$, we have $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}((K^+ \cap \mathcal{Q}') \setminus E'_j) \models \alpha$. This is equivalent to $(K^+ \cap \mathcal{Q}') \setminus (\bigcap_{j:(M_j,M_i)\in\mathcal{E}} E'_j) \models \alpha$. By the definition of $E_{\mathcal{Q}'}$, we have $E_i \subseteq \bigcap_{j:(M_j,M_i)\in\mathcal{E}} E'_j$. It follows that $(K^+ \cap \mathcal{Q}') \setminus E_i \models \alpha$, contradicting the fact that E_i is an envelope for $\mathcal{S}_i$ w.r.t. $\mathcal{Q}'$. □

Note that $E_{\mathcal{Q}'} = \{E'_1, ..., E'_m\}$ may not be tight. One can obtain a tight partial envelope by examining each element in each E'_i in $E_{\mathcal{Q}'}$, removing the formula if we still have a partial envelope without it, and keeping it otherwise.

Since $\mathcal{Q}'$ is finite, E_i is finite and so is E'_i. If a query $q \in \mathcal{Q}'$ is posed by the querying agent M_i, it will be answered "Yes" if $q \in (K^+ \cap \mathcal{Q}') \setminus E'_i$, "No" if $\neg q \in (K^+ \cap \mathcal{Q}') \setminus E'_i$, and "Unknown" otherwise. If a query $q' \notin \mathcal{Q}'$ is posed, before answering q, the envelope will be updated by taking E'_i as the new secrecy set for each $1 \leq i \leq m$ and computed according to Algorithm 2.

COROLLARY 3.4. *Given a KB $\mathcal{K} = \langle K, \mathcal{Q}, \Omega\rangle$, a secrecy structure $\mathcal{S} = \langle \mathcal{M}, \mathbb{S} = \{S_1, S_2, ..., S_m\}, \mathcal{G}\rangle$, and two finite subsets of $\mathcal{Q}$: $\mathcal{Q}_1 \subseteq \mathcal{Q}_2 \subseteq \mathcal{Q}$, let $E_{\mathcal{Q}_1} = \{E^1_1, ..., E^1_m\}$ be obtained from Algorithm 2 with the input K, $\mathcal{S} = \langle \mathcal{M}, \mathbb{S} = \{S_1, ..., S_m\}, \mathcal{G}\rangle$ and $\mathcal{Q}_1$. Then $E_{\mathcal{Q}_2} = \{E^2_1, ..., E^2_m\}$ obtained from Algorithm 2 with the input K, $\mathcal{S}' = \langle \mathcal{M}, \{E^1_1, ..., E^1_m\}, \mathcal{G}\rangle$ and $E_{\mathcal{Q}_2}$ is a partial envelope for $\mathcal{S}$ w.r.t. $\mathcal{Q}_2$.*

Corollary 3.4 shows that when the system evolves with new queries, a new envelope could be obtained from the old one, and so queries can be safely answered by considering the new envelope.

Next we show how to answer queries given a partial envelope. Given a KB $\mathcal{K}$, a secrecy structure $\mathcal{S}$ and a finite set of formulas $\mathcal{Q}'$ such that $K \cup \bigcup_{i=1}^m S_i \subseteq \mathcal{Q}' \subseteq \mathcal{Q}$, and a partial envelope $E = \{E_1, ..., E_m\}$ for $\mathcal{S}$ w.r.t. $\mathcal{Q}'$, let $\mathcal{D} = \mathcal{Q}' \times \mathcal{M}$. Define a function $\rho : \mathcal{D} \rightarrow \Omega$, by

$$\rho(\alpha, M_i) = \begin{cases} Y & \text{if } \alpha \in (K^+ \cap \mathcal{Q}') \setminus E_i, \\ N & \text{if } \neg\alpha \in (K^+ \cap \mathcal{Q}') \setminus E_i, \\ U & \text{otherwise.} \end{cases}$$

The following theorem shows that for any query $\alpha \in \mathcal{Q}'$, if α is not in the current partial envelope, it can be truthfully answered without compromising secrecy.

THEOREM 3.5. *ρ is secrecy-preserving w.r.t. $\mathcal{D}$.*

PROOF. By definition, ρ is negation consistent. We need to show that ρ satisfies the three secrecy-preservation properties in Section 3.1.

— Yes Property: We need to show that $\mathcal{Q}^{\rho}_{i,Y} \subseteq K^{+}$. By definition of ρ, $\mathcal{Q}^{\rho}_{i,Y} = \{\alpha \mid \rho(\alpha, M_i) = Y\} = (K^{+} \cap \mathcal{Q}') \setminus E_i \subseteq (K^{+} \cap \mathcal{Q}') \subseteq K^{+}$.

— Closure Property: We need to show that $(\mathcal{Q}^{\rho}_{i,Y})^{+} \cap (\mathcal{Q}^{\rho}_{i,N} \cup \mathcal{Q}^{\rho}_{i,U}) = \emptyset$. Suppose that there is α such that $\alpha \in (\mathcal{Q}^{\rho}_{i,Y})^{+}$ and $\alpha \in \mathcal{Q}^{\rho}_{i,N} \cup \mathcal{Q}^{\rho}_{i,U}$. Since $\alpha \in (\mathcal{Q}^{\rho}_{i,Y})^{+}$ and $\mathcal{Q}^{\rho}_{i,Y}$ is disjoint from $\mathcal{Q}^{\rho}_{i,N} \cup \mathcal{Q}^{\rho}_{i,U}$, we have $\alpha \in (\mathcal{Q}^{\rho}_{i,Y})^{+} \setminus \mathcal{Q}^{\rho}_{i,Y}$ and $\mathcal{Q}^{\rho}_{i,Y} \vdash \alpha$. By definition of ρ, $(K^{+} \cap \mathcal{Q}') \setminus E_i \vdash \alpha$, implying $K \vdash \alpha$. If $\alpha \in \mathcal{Q}^{\rho}_{i,N}$, then, by definition of ρ, $\neg\alpha \in (K^{+} \cap \mathcal{Q}') \setminus E_i$, implying $K \vdash \neg\alpha$. However, this contradicts the assumption that K is consistent. Therefore, $\alpha \in \mathcal{Q}^{\rho}_{i,U}$. Obviously, when $\alpha \in \mathcal{Q}'$ and $K \vdash \alpha$, then $\alpha \in K^{+} \cap \mathcal{Q}'$, which implies that $\alpha \in E_i$. Since $(K^{+} \cap \mathcal{Q}') \setminus E_i \vdash \alpha$, from the soundness of the inference system, $(K^{+} \cap \mathcal{Q}') \setminus E_i \vDash \alpha$. This contradicts the fact that E is a partial envelope (see property E1'). Hence, ρ satisfies the Closure Property.

— Secrecy Property: We need to show that $(\mathcal{P}^{\rho}_{i,Y})^{+} \cap S_i = \emptyset$. Suppose that $(\mathcal{P}^{\rho}_{i,Y})^{+} \cap S_i \neq \emptyset$. Let $\alpha \in S_i$ such that $(\mathcal{P}^{\rho}_{i,Y})^{+} \vdash \alpha$. Then $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}((K^{+} \cap \mathcal{Q}') \setminus E_j) \vdash \alpha$ and by the soundness of $\vdash$, we obtain $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}((K^{+} \cap \mathcal{Q}') \setminus E_j) \vDash \alpha$. This contradicts our assumption that E is a partial envelope (see property E2'). Therefore, ρ satisfies the Secrecy Property. □

3.3. Correspondance between $\mathcal{D}$-answers and Partial Envelopes

As shown in Theorems 2.9 and 2.10, secrecy-preserving reasoners and secrecy envelopes are equivalent concepts. In Theorem 3.5, we show how to obtain a secrecy-preserving $\mathcal{D}$-answer from a partial envelope. Here we show how a secrecy-preserving $\mathcal{D}$-answer corresponds to a partial envelope.

Given a KB $\mathcal{K} = \langle K, \mathcal{Q}, \Omega\rangle$, a secrecy structure $\mathcal{S} = \langle \mathcal{M}, \mathbb{S} = \{S_1, S_2, ..., S_m\}, \mathcal{G}\rangle$, a finite set $\mathcal{D} \subseteq \mathcal{Q} \times \mathcal{M}$ and a secrecy-preserving $\mathcal{D}$-answer ρ w.r.t. $\mathcal{D}$, let $\mathcal{Q}' = \bigcup_{i\in\{1,...,m\}} \bigcup_{B\in\Omega} \mathcal{Q}^{\rho}_{i,B}$. Then $\mathcal{Q}'$ contains all the queries that have been evaluated for some querying agent(s). Since different querying agents may ask different queries, a query $q \in \mathcal{Q}'$ that has been evaluated for some M_i such that $\rho(q, M_i) \in \Omega$ may not be defined for another M_j where $i \neq j$. To obtain an answer such that all querying agents share the same query space, we extend the $\mathcal{D}$-answer as per Algorithm 3. Let $\mathcal{D}' = \mathcal{Q}' \times \mathcal{M}$. After an execution of Algorithm 3, we obtain a $\mathcal{D}'$-answer $\rho' : \mathcal{D}' \to \Omega$ where ρ' is a total function. It follows from Lemma 3.1 that ρ' is secrecy-preserving.

```
input : $\mathcal{D}$, $\mathcal{Q}'$
for $1 \leq i \leq m$ do
    for $\alpha \in \mathcal{Q}'$ do
        if $(\alpha, M_i) \notin \mathcal{D}$ then
            call Lazy Evaluation $Lazy(\alpha, M_i)$ (see Algorithm 1)
        end
    end
end
```

Algorithm 3: Extending a $\mathcal{D}$-Answer

THEOREM 3.6. *Let $\mathbb{E} = \{E_1, ..., E_m\}$ where $E_i = (K^{+} \cap \mathcal{Q}') \setminus (\mathcal{Q}')^{\rho'}_{i,Y}$. Then $\mathbb{E}$ is a partial envelope for $\mathcal{S}$ w.r.t. $\mathcal{Q}'$.*

PROOF. Since $(\mathcal{Q}')^{\rho'}_{i,U}$ is initialized to be $\bigcup_{j:(M_i,M_j)\in\mathcal{E}} S_j$ and $(\mathcal{Q}')^{\rho'}_{i,Y} \cap (\mathcal{Q}')^{\rho'}_{i,U} = \emptyset$ (see Section 3.1), we have $S_i \subseteq E_i$. Moreover, by Lazy Evaluation (Algorithm 1 lines 3-16) and the assumption that $S_i \subseteq K^+ \setminus \mathfrak{T}$, we have $E_i \subseteq (K^+ \cap \mathcal{Q}') \setminus \mathfrak{T}$. Next we verify that $\mathbb{E}$ satisfies properties E1' and E2' in Section 3.2.

— **[E1']**: Suppose that for some i and $\alpha \in E_i$, $(K^+ \cap \mathcal{Q}') \setminus E_i \vDash \alpha$. Since the inference system $\vdash$ is complete w.r.t. $\vDash$, we have $(K^+ \cap \mathcal{Q}') \setminus E_i \vdash \alpha$, i.e., $(\mathcal{Q}')^{\rho'}_{i,Y} \vdash \alpha$. Therefore, $\alpha \in ((\mathcal{Q}')^{\rho'}_{i,Y})^+$. By definition of E_i, $E_i \subseteq (K^+ \cap \mathcal{Q}')$ and so $\alpha \in (K^+ \cap \mathcal{Q}')$. Since $E_i \cap (\mathcal{Q}')^{\rho'}_{i,Y} = \emptyset$, it follows from Algorithms 3 and 1 that $\alpha \in (\mathcal{Q}')^{\rho'}_{i,U} \subseteq (\mathcal{Q}')^{\rho'}_{i,N} \cup (\mathcal{Q}')^{\rho'}_{i,U}$. However, this contradicts the Closure Property and so contradicts the assumption that ρ' is secrecy-preserving. Therefore, E1' is satisfied.

— **[E2']**: Suppose that for some i and $\alpha \in S_i$, we have $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}((K^+ \cap \mathcal{Q}') \setminus E'_j) \vDash \alpha$. It follows from the definition of E_i that $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}(\mathcal{Q}')^{\rho'}_{i,Y} \vDash \alpha$. Since the inference system $\vdash$ is complete w.r.t $\vDash$, we have $\bigcup_{j:(M_j,M_i)\in\mathcal{E}}(\mathcal{Q}')^{\rho'}_{i,Y} \vdash \alpha$, i.e., $(\mathcal{P}')^{\rho'}_{i,Y} \vdash \alpha$. This contradicts the Secrecy Property of ρ'. Therefore, E2' is satisfied. □

Note that depending on the order of chosen elements in the set $\{1, ..., m\}$ and $\mathcal{Q}'$ in Algorithm 3, different executions of Algorithm 3 may result different answers ρ', and hence, different partial envelopes may be obtained. However, once an order is fixed, the resulting answer ρ' and the corresponding partial envelope are both secrecy-preserving.

One may have realized that the complexity of Lazy Evaluation depends heavily on the condition checked in line 7 of Algorithm 1. As history grows larger, $\mathcal{P}^{\rho}_{j,Y}$ gets larger and checking this condition on the fly takes more and more time. On the other hand, the complexity of computing a partial envelope mainly depends on the condition checking at line 3 in Algorithm 2. Given a specific language and a well-formed formula γ in the language, breaking its proof can often be done by looking at the formula itself and checking subformulas of γ as well as the operators that connect the subformulas. Such an example will be given in Section 4.2. Compared to the Lazy Evaluation, this is obviously more efficient. With the partial envelope in hand, queries posed by agent M_i can be answered simply by checking membership of $(K^+ \cap \mathcal{Q}') \setminus E_i$ where both $K^+ \cap \mathcal{Q}'$ and E_i have been logged.

Remark regarding the relationship between CQE and our MSQ System. In the literature of controlled query evaluation [Biskup 2011; Biskup and Weibert 2008; Biskup et al. 2010; Biskup et al. 2008; Biskup and Tadros 2012], a secret is preserved if for any sequence of queries there are two different models that are indistinguishable in the sense that they produce the same answers to the queries; one being a model of the secret and the other a model of the negation of the secret. Our framework also respects this property of secrecy-preservation. More specifically, given a finite set of queries $\mathcal{Q}' \subseteq \mathcal{Q}$ and a partial envelope $\mathbb{E} = \{E_1, ..., E_m\}$, by property E1', for any $\alpha \in E_i$, $(K^+ \cap \mathcal{Q}') \setminus E_i \nvDash \alpha$, and so $(K^+ \cap \mathcal{Q}') \setminus E_i \cup \{\neg\alpha\}$ is consistent. Moreover, since $\alpha \in E_i \subseteq K^+$, $K^+ \nvDash \neg\alpha$, and therefore $(K^+ \cap \mathcal{Q}') \cup \{\alpha\}$ is also consistent. By Theorem 3.5, any query $q \in \mathcal{Q}'$ can be truthfully answered to agent M_i if $q \in (K^+ \cap \mathcal{Q}') \setminus E_i$ or $\neg q \in (K^+ \cap \mathcal{Q}') \setminus E_i$, and unknown otherwise. This means that our MSQ system produces the same answers to a sequence of queries posed by any querying agent M_i in the both models: one being a model of $(K^+ \cap \mathcal{Q}') \cup \{\alpha\}$ and the other a model of $(K^+ \cap \mathcal{Q}') \setminus E_i \cup \{\neg\alpha\}$.

4. ILLUSTRATION OF MSQ SYSTEMS

In this section we aim at illustrating the construction of MSQ systems. As the first example, we take a simple language of Propositional Horn logic. Propositional Horn theories are widely used in computer science [Makowsky 1987; Dowling and Gallier 1984]. For the second illustration, we take the Description Logic $\mathcal{AL}$ which extends $\mathcal{FL}^-$ by allowing atomic negation [Baader et al. 2003]. It is also a sub-language of the Description Logic $\mathcal{ALC}$ which is extensively researched in the Semantic Web community [Baader et al. 2003] and serves as a foundation of the Web Ontology Language (OWL) [Staab and Studer 2009]. In both cases, we have not dealt with the tightness of envelopes. A simple approach to obtaining a tight partial envelope from a partial envelope is by checking whether removing each assertion in the partial envelope compromises the secrecy and removing the assertion from the partial envelope if it doesn't. Depending on the properties of a language, algorithms for computing a partial envelope may be optimized to create a tight partial envelope during the construction.

From Sections 2 and 3, in particular, Theorems 2.11 and Lemma 3.3, we see that one way to build an envelope for a secrecy set S_i is to hide, for each $\alpha \in S_i$, and every $\Gamma \in \mathcal{F}_\alpha$, some formula $\gamma \in \Gamma$. Since we have assumed that the inference system $\vdash$ is complete w.r.t. $\models$, this means that for every inference rule $\Gamma \vdash \alpha$ if α needs to be protected, we also protect one element of Γ, i.e. ϕ_Γ. In Section 2.1, we have commented that such ϕ_Γ could be computed given a KB represented in a specific language. The idea is to invert the inference rules into new rules that enforce the intuitively obvious requirement: *whenever the conclusion of an inference rule is to be secret so must be at least one of its premises*. This methodology was developed by the authors in [Tao et al. 2010]. We illustrate this approach with the following two cases and show how the inverted rules (together with some additional rules) help construct an envelope. Once an envelope is constructed, secrecy can be preserved while answering queries according to our framework, see Sections 2 and 3.

4.1. Propositional Horn MSQ System

Recall that a propositional *Horn Clause* is a clause containing at most one positive literal, i.e., generally, it is of the form: $x_1 \wedge \cdots \wedge x_k \rightarrow \eta$ where x_1, x_2, ..., x_k are propositional names and η is either a propositional name, in which case the Horn clause is called a *rule*, or it is $\bot$, in which case it is called a *constraint*. In this section we shall have no further use of constraints. A Horn clause is called a *fact* if $k = 0$ and $\eta \neq \bot$, i.e. it consists of a single positive literal. We assume a single underlying inference rule, *forward chaining*, which is known to be sound and complete for Horn logic w.r.t. the usual semantics of propositional logic,

$$\textsc{Forward Chaining } (FC)\text{:}\quad \frac{l_1 \wedge l_2 \wedge \cdots \wedge l_k \rightarrow p, \qquad l_1, \qquad l_2, \qquad ..., \qquad l_k}{p}$$

where $p, l_1, l_2, ..., l_k$ are all propositional names.

A *Horn KB* is a triple $\mathcal{K} = \langle K, \mathcal{Q}, \Omega \rangle$ where K is a finite set of Horn clauses, $\mathcal{Q}$ is the set of all (relevant) facts (the query space), and $\Omega = \{Y, N, U\}$ is the answer space. The set K can be further partitioned $K = R \cup F$ where R, the TBox, contains a set of rules in K and F, the ABox, is the set of facts in K. By F^+ we denote the set of all facts derivable by applying the FC-rule with assumptions in the ABox F and rules in the TBox R: $F^+ = \{p \mid K \vdash_{FC} p \text{ and } p \text{ is a fact}\}$. Obviously, if K is finite, so is F^+.

Given a collection of querying agents $\mathcal{M} = \{M_1, M_2, ..., M_m\}$, a corresponding collection of secrecy sets $\mathbb{S} = \{S_1, S_2, ..., S_m\}$, and a communication graph $\mathcal{G}$, we have a secrecy structure $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G} \rangle$. We assume $S_i \subseteq F^+$. To compute an envelope for $\mathcal{S}$, we can use the approach suggested in Theorem 2.8: for $1 \leq i \leq m$, compute an envelope E_i^1 for the single-agent secrecy structure $\mathcal{S}_i = \langle \{M_i\}, \{S_i\}, \langle \{M_i\}, \{(M_i, M_i)\}\rangle\rangle$. Letting $E_i^* = \bigcup_{j:(M_i,M_j)\in\mathcal{E}} E_j^1$, $\mathbb{E}^* = \{E_1^*, ..., E_m^*\}$ is an envelope for $\mathcal{S}$.

It remains to show how to compute an envelope for the single-agent secrecy structure $\mathcal{S}_i$. We invert the FC-rule into a new rule, denoted by FC^I and it is formulated as follows for each $1 \leq i \leq m$:

$$FC^I\text{-RULE:}\quad \frac{p \in E_i', \qquad l_1 \wedge l_2 \wedge ... \wedge l_k \rightarrow p \in R, \qquad l_1, l_2, ..., l_k \in F^+ \setminus E_i'}{E_i' := E_i' \cup \{l\},\ \text{for some } l \in \{l_1, l_2, ..., l_k\}}$$

The actual computation of the envelopes proceeds by initializing $\mathbb{E}' = \{E_1', E_2', ..., E_m'\}$ with $E_i' = S_i$ $(1 \leq i \leq m)$. The FC^I-rule is then applied repeatedly until it is no longer applicable. Denote by $\mathbb{E}^1 = \{E_1^1, E_2^1, ..., E_m^1\}$ the resulting collection of sets. To show the correctness of our procedure we must prove that for each $1 \leq i \leq m$,

THEOREM 4.1. *E_i^1 is a finite secrecy envelope for $\mathcal{S}_i$.*

PROOF. It suffices to show that E_i^1 satisfies Axiom E1: for every $\alpha \in E_i^1$, $F^+ \setminus E_i^1 \nvDash \alpha$. Since $\vdash_{FC}$ is complete w.r.t. $\vDash$ (for Horn KBs), we argue instead that for every $\alpha \in E_i^1$, $F^+ \setminus E_i^1 \nvdash_{FC} \alpha$. Note that for a fixed i, once a rule in R is used in an application of FC^I-rule for computing E_i^1, it is no longer applicable (for that fixed i). Thus, after at most $|R|$ applications of the FC^I-rule (for that i) the computation of the set E_i^1 is complete. Hence, for any $\alpha \in E_i^1$, for any rule $l_1 \wedge ... \wedge l_k \rightarrow \alpha \in R$, we have $\{l_1, ..., l_k\} \cap E_i^1 \neq \emptyset$ and so $F^+ \setminus E_i^1 \nvdash_{FC} \alpha$. The theorem follows. □

Since $\mathbb{E}^*$ is an envelope, by Theorem 2.10, a query can be safely answered by checking whether it is provable from the given KB and its membership status w.r.t. $\mathbb{E}^*$. The envelope $\mathbb{E}^*$ resulting from the single-agent "slices" in a manner indicated above need not be tight.

Example 4.2. Given a Horn KB $\mathcal{K} = \langle K, \mathcal{S}, \mathcal{R}\rangle$ where $K = \langle F = \{l_1, l_2, s\}, R = \{l_1 \wedge l_2 \rightarrow s\}\rangle$, $\mathcal{S} = \langle \{M_1, M_2\}, \{S_1, S_2\}, \langle \{M_1, M_2\}, \{(M_1, M_1), (M_2, M_2), (M_1, M_2)\}\rangle\rangle$ and $S_1 = S_2 = \{s\}$. Suppose that $E_1^1 = \{s, l_1\}$ and $E_2^1 = \{s, l_2\}$. Obviously, E_1^1 is an envelope for S_1 and E_2^1 is an envelope for S_2. In fact, E_1^1 is a tight envelope for S_1 and E_2^1 is a tight envelope for S_2. Let $E_1^* = E_1^1 \cup E_2^1 = \{s, l_1, l_2\}$ and $E_2^* = E_2^1$. Then we have $\mathbb{E}^* = \{E_1^*, E_2^*\}$ is an envelope for $\mathcal{S}$. However, $\mathbb{E}^*$ is not tight because we could remove l_1 from E_1^* and still result an envelope for $\mathcal{S}$. In fact, there are two tight envelopes for $\mathcal{S}$: $\mathbb{E}_1^* = \{\{s, l_1\}, \{s, l_1\}\}$ and $\mathbb{E}_2^* = \{\{s, l_2\}, \{s, l_2\}\}$. □

We next show that computing minimum size envelopes is NP-hard for Horn KBs. We specify the *Minimum Envelope* problem (*ME*) by the pair $\langle\langle \mathcal{K}, \mathcal{S}\rangle, N\rangle$ where $\mathcal{K} = \langle K, \mathcal{Q}, \Omega\rangle$ is a Horn KB, $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G}\rangle$ is a secrecy structure for $\mathcal{K}$ and N is a positive integer. The decision problem is to determine whether $\mathcal{S}$ has a secrecy envelope $\mathbb{E} = \{E_1, ..., E_m\}$ satisfying $|\bigcup_{1\leq i\leq m} E_i| \leq N$. It is easy to see that the problem is in NP as this only involves checking that $\mathbb{E}$ satisfies the axioms E1 and E2. The NP hardness of this problem can be shown by reduction from the Hitting Set (HS) problem: Given a finite set X, a finite collection of non-empty sets $\mathcal{C} = \{C_1, ..., C_k\} \subseteq \mathbb{P}(X)$ and an integer $0 \leq N \leq |X|$, the problem is to determine whether or not there is a subset $Y \subseteq X$ such that $|Y| \leq N$ and for every $C \in \mathcal{C}$, $C \cap Y \neq \emptyset$. Given such an instance of HS, we construct an instance of ME as follows:

— $\mathcal{M} = \{M_1\}$, a single querying agent;
— the communication graph consists of a single self-loop on M_1;

— $\mathbb{S} = \{S_1\}$, with $S_1 = \{s_i \mid C_i \in \mathcal{C}\}$, where s_i's are new symbols;
— $N' = N + |\mathcal{C}|$;
— $K = F \cup R$ where $F = X \cup S_1$ and $R = \{l_1 \wedge ... \wedge l_r \rightarrow s_i \mid C_i = \{l_1, ..., l_r\} \in \mathcal{C}\}$.

CLAIM 2. *$\mathcal{C}$ has a hitting set $Y \subseteq X$ with $|Y| \leq N$ if and only if $\langle \mathcal{K}, \mathcal{S} \rangle$ has a secrecy envelope $\mathbb{E} = \{E_1\}$ such that $S_1 \subseteq E_1 \subseteq F^+$ and $|E_1| \leq N'$.*

PROOF. Suppose that $Y \subseteq X$ is a hitting set for $\mathcal{C}$ with $|Y| \leq N$. Define the set $E_1 := Y \cup S_1$. Since $Y \cap S_1 = \emptyset$, $|E_1| = |Y| + |S_1| \leq N + |\mathcal{C}|$. Moreover, for every $C \in \mathcal{C}$, $C \cap Y \neq \emptyset$. Therefore, none of the rules in K can be used in applying the FC-rule to $F^+ \setminus E_1$. It follows that E_1 is a secrecy envelope for $\mathcal{S}$.

Conversely, let $\mathbb{E} = \{E_1\}$ be a secrecy envelope for $\mathcal{S}$ such that $S_1 \subseteq E_1 \subseteq F^+$ and $|E_1 \backslash S_1| \leq N$. By Axiom E1 and the soundness of the FC-rule, this implies that for every $\alpha \in E_1 : F^+ \backslash E_1 \nvdash_{FC} \alpha$. We show that the HS instance $\mathcal{C} = \{C_1, ..., C_k\} \subseteq \mathbb{P}(X)$, together with an integer $0 < N \leq |X|$, has a hitting set of size at most N. Define $Y := E_1 \setminus S_1$. It now suffices to show that for every $C_i \in \mathcal{C}$, $Y \cap C_i \neq \emptyset$. Let $C_i = \{l_1, ..., l_r\}$; by the definition of the reduction, this implies that $l_1 \wedge ... \wedge l_r \rightarrow s_i$ belongs to R. If none of the $l_j (1 \leq j \leq r)$ belongs to Y, then they all belong to $F^+ \setminus Y$ and hence also to $F^+ \setminus E_1$ because $C_i \cap S_1 = \emptyset$. Therefore, $F^+ \setminus E_1 \vdash_{FC} s_i \in E_1$, which yields a contradiction. □

In light of the preceding result, it is not feasible to compute a minimal cardinality envelope. However, because of the finiteness of Propositional Horn KB envelopes, a tight envelope can be obtained by first constructing an envelope and then repeatedly eliminating (from the envelope) those assertions that are not essential for protecting secrets.

4.2. MSQ System for Description Logic $\mathcal{AL}$

Recall from Section 1 that the idea behind our approach to secrecy-preservation is to place secrets as well as assertions from which secrets are deducible into an envelope so that under OWA a reasoner can feign ignorance without resorting to outright lying. In Section 4.1 we have seen how this idea was applied to the simple case of Propositional Horn Logic where all the facts we need to protect are propositional facts. In this section we illustrate how to apply our MSQ framework to a considerably more expressive language, the Description Logic $\mathcal{AL}$ [Baader et al. 2003], where we need to protect compound assertions which may contain quantifiers. Usually, a KB consists of both an Abox and a TBox [Baader et al. 2003]; however, since our goal is to illustrate the MSQ framework, considering the subsumption problem for the sublanguage $\mathcal{FL}_0$ of $\mathcal{AL}$ is coNP-complete w.r.t. an acyclic TBox and PSpace-complete w.r.t. cyclic TBoxes [Baader 2009], for the sake of simplicity, we shall assume that the TBox is empty. Moreover, in addition to its added expressive power, it has very good computational characteristics, e.g., concept satisfiability (without TBoxes) can be checked in linear time [Schmidt-Schauß and Smolka 1991].

4.2.1. Syntax and Semantics. The non-logical signature of $\mathcal{AL}$ consists of three mutually disjoint sets: a set of *concept names* $N_{\mathcal{C}}$, a set of *role names* $N_{\mathcal{R}}$ and a set of *individuals* (or *objects*) $N_{\mathcal{O}}$. The set of $\mathcal{AL}$ *expressions* consists of the set of role names $N_{\mathcal{R}}$ and the set of *concept expressions* $\mathcal{C}$ recursively defined as follows:

$$C, D \longrightarrow A \mid \top \mid \bot \mid \neg A \mid C \sqcap D \mid \forall R.C \mid \exists R$$

where $A \in N_{\mathcal{C}}$, $\top$ is the *top symbol*, $\bot$ is the *bottom symbol*, $C, D \in \mathcal{C}$ and $R \in N_{\mathcal{R}}$. An $\mathcal{AL}$ *assertion* is an expression of the form $C(a)$ or $R(a, b)$ where $C \in \mathcal{C}$, $R \in N_{\mathcal{R}}$ and $a, b \in N_{\mathcal{O}}$. An ABox is a finite set of assertions.

Under the classical interpretation, a formula is interpreted as either "true" or "false". However, under the OWA, given a KB K, the answer to a query α can be "Yes" (if

$K \vDash \alpha$), "No" (if $K \vDash \neg\alpha$), or "Unknown" (otherwise). To test $K \vDash \alpha$ one usually runs the tableau algorithm to check the satisfiability of $K \cup \{\neg\alpha\}$, see [Baader et al. 2003]. This is what we may think of as an algorithmic approach to OWA. In $\mathcal{ALC}$, any formula can be transformed to its negation normal form using De Morgan's laws to push negation inside and then check for satisfiability. Such an algorithmic approach does not work for $\mathcal{AL}$ because negation is allowed only in front of concept names and existential restriction is unqualified; hence, the usual tableau algorithm is not directly applicable. In this section we incorporate the "Unknown" truth value into the semantics of $\mathcal{AL}$ and transform the tableau algorithm for satisfiability into a sound and complete inference system for answering queries. As one shall see, our semantics is a generalization of the classical two-valued semantics. If an application only requires two-valued answers such as "Yes" and "No", or maybe "Yes" and "Unknown", this three-valued semantics is easily adapted into the two-valued case.

With the soundness and completeness of the inference system for answering queries, we further show how to invert the rules of the inference system for $\mathcal{AL}$ and thus obtain an algorithm for constructing envelopes. In more detail, the semantics of $\mathcal{AL}$ is provided by means of an *OW-interpretation*[7] $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ where Δ is a *non-empty domain* and $\cdot^{\mathcal{I}}$ is an interpretation function such that

— for each individual $a \in N_{\mathcal{O}}$, $a^{\mathcal{I}} \in \Delta$;
— for each concept name $A \in N_{\mathcal{C}}$, $A^{\mathcal{I}}$ is a *weak 3-partition* of Δ, $A^{\mathcal{I}} = (A^{\mathcal{I}}_N, A^{\mathcal{I}}_U, A^{\mathcal{I}}_Y)$; this means that $A^{\mathcal{I}}_N, A^{\mathcal{I}}_U$ and $A^{\mathcal{I}}_Y$ are mutually disjoint and their union is Δ (but they can be empty);
— for each role $R \in N_{\mathcal{R}}$, $R^{\mathcal{I}}$ is a weak 3-partition of $\Delta \times \Delta$ of the following special form $R^{\mathcal{I}} = (\emptyset, R^{\mathcal{I}}_U, R^{\mathcal{I}}_Y)$; the reason for $R^{\mathcal{I}}_N = \emptyset$ is that in $\mathcal{AL}$, role negation is not allowed.

The function $\cdot^{\mathcal{I}}$ is extended to compound $\mathcal{AL}$ expressions for all $a \in N_{\mathcal{O}}$, $A \in N_{\mathcal{C}}$, $R \in N_{\mathcal{R}}$ and $C, D \in \mathcal{C}$ as follows:

(1) $\top^{\mathcal{I}} = (\emptyset, \emptyset, \Delta)$ and $\bot^{\mathcal{I}} = (\Delta, \emptyset, \emptyset)$;
(2) $(\neg A)^{\mathcal{I}} = (A^{\mathcal{I}}_Y, A^{\mathcal{I}}_U, A^{\mathcal{I}}_N)$;
(3) $(C \sqcap D)^{\mathcal{I}} = ((C \sqcap D)^{\mathcal{I}}_N, (C \sqcap D)^{\mathcal{I}}_U, (C \sqcap D)^{\mathcal{I}}_Y)$ where
— $(C \sqcap D)^{\mathcal{I}}_Y = C^{\mathcal{I}}_Y \cap D^{\mathcal{I}}_Y$, and
— $(C \sqcap D)^{\mathcal{I}}_N = C^{\mathcal{I}}_N \cup D^{\mathcal{I}}_N$,
— $(C \sqcap D)^{\mathcal{I}}_U = \Delta \setminus ((C \sqcap D)^{\mathcal{I}}_Y \cup (C \sqcap D)^{\mathcal{I}}_N)$;
(4) $(\exists R)^{\mathcal{I}} = (\emptyset, (\exists R)^{\mathcal{I}}_U, (\exists R)^{\mathcal{I}}_Y)$ where
— $(\exists R)^{\mathcal{I}}_Y = \{d \in \Delta \mid \exists b \in \Delta : (d, b) \in R^{\mathcal{I}}_Y\}$ and
— $(\exists R)^{\mathcal{I}}_U = \Delta \setminus (\exists R)^{\mathcal{I}}_Y = \{d \in \Delta \mid \forall b \in \Delta : (d, b) \in R^{\mathcal{I}}_U\}$;
(5) $(\forall R.C)^{\mathcal{I}} = ((\forall R.C)^{\mathcal{I}}_N, (\forall R.C)^{\mathcal{I}}_U, (\forall R.C)^{\mathcal{I}}_Y)$ where
— $(\forall R.C)^{\mathcal{I}}_Y = \{d \in \Delta \mid \forall b \in \Delta : (d, b) \in R^{\mathcal{I}}_Y \rightarrow b \in C^{\mathcal{I}}_Y\}$,
— $(\forall R.C)^{\mathcal{I}}_N = \{d \in \Delta \mid \exists b \in \Delta[(d, b) \in R^{\mathcal{I}}_Y \wedge b \in C^{\mathcal{I}}_N]\}$ and
— $(\forall R.C)^{\mathcal{I}}_U = \Delta \setminus ((\forall R.C)^{\mathcal{I}}_Y \cup (\forall R.C)^{\mathcal{I}}_N)$.

An OW-interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ *satisfies* assertion $C(a)$ (resp. $R(a, b)$), if $a^{\mathcal{I}} \in C^{\mathcal{I}}_Y$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}_Y$); this is denoted by $\mathcal{I} \vDash C(a)$ (resp. $\mathcal{I} \vDash R(a, b)$). An ABox $\mathcal{A}$ is *satisfiable* if there is an OW-interpretation $\mathcal{I}$ satisfying all the assertions in $\mathcal{A}$; we then say that $\mathcal{I}$ is a *model* of $\mathcal{A}$. An individual $a \in N_{\mathcal{O}}$ is an *instance* of a concept C w.r.t $\mathcal{A}$, denoted by $\mathcal{A} \vDash C(a)$, if $\mathcal{I} \vDash C(a)$ for all models $\mathcal{I}$ of $\mathcal{A}$.

An $\mathcal{AL}$ knowledge base is a triple $\mathcal{K} = \langle \mathcal{A}, \mathcal{Q}, \Omega \rangle$ where $\mathcal{A}$ is an $\mathcal{AL}$ ABox, $\mathcal{Q}$, the query space, is the set of all assertions over the vocabulary of $\mathcal{A}$, and $\Omega = \{Y, N, U\}$

[7] OW stands for Open World.

is the answer space. Given a collection of querying agents $\mathcal{M} = \{M_1, M_2, ..., M_m\}$ with respective secrecy sets $\mathbb{S} = \{S_1, S_2, ..., S_m\}$ and a communication graph $\mathcal{G}$, we define a secrecy structure $\mathcal{S} = \langle \mathcal{M}, \mathbb{S}, \mathcal{G} \rangle$ (cf. Definition 2.2). We say that an $\mathcal{AL}$ ABox is in *normal form* if universal restriction does not contain a conjunction, i.e., for every concept expression of the form $\forall R.C$ occurring in the ABox, C is not in the form of $C_1 \sqcap C_2$. It is easy to verify that $\forall R.(C_1 \sqcap C_2)$ is equivalent to $\forall R.C_1 \sqcap \forall R.C_2$.

We make the following simplifying assumptions

(1) the ABox $\mathcal{A}$ and the secrecy sets $\mathbb{S}$ are satisfiable and in normal form,
(2) all individual names occurring in $\mathbb{S}$ occur in $\mathcal{A}$,
(3) for $1 \leq i \leq m$ and every $\alpha \in S_i$, $\mathcal{A} \vDash \alpha$, and
(4) querying agents ask queries only about individuals occurring in $\mathcal{A}$.

Recall that an MSQ system is a triple $\langle \mathcal{K}, \mathcal{S}, \mathcal{R} \rangle$ (See Definition 2.3). By Theorems 2.8 and 2.10, in order to define a secrecy-preserving reasoner $\mathcal{R}$, it suffices to show how to construct an envelope for each induced single-agent secrecy structure, which for notational simplicity, we denote by $\mathcal{S} = \langle \{M\}, \{S\}, \langle \{M\}, \{(M, M)\} \rangle \rangle$. Henceforth, we focus on the single-agent case.

4.2.2. A Sound and Complete Inference System for $\mathcal{AL}$. As indicated in Sections 2 and 3, in order to construct an envelope, we need an underlying sound and complete inference system for $\mathcal{AL}$. We fashion our inference system after the tableau algorithm for the satisfiability problem for the Description Logic $\mathcal{ALCQ}$ [Baader and Sattler 2001]. Since the query space for $\mathcal{AL}$ knowledge bases is potentially infinite, as suggested in Section 3.2, at the pre-query stage, we restrict the inferences (and, correspondingly, the envelope) to a finite set of sub-expressions of assertions occurring in $\mathcal{A} \cup S$.

Definition 4.3. Given a set of $\mathcal{AL}$-assertions $\mathcal{B}$, the set of sub-expressions of roles and concept expressions occurring in $\mathcal{B}$, denoted by $Sub(\mathcal{B})$, is defined as follows:

— $C(a) \in \mathcal{B} \Rightarrow C \in Sub(\mathcal{B})$ and $R(a, b) \in \mathcal{B} \Rightarrow R \in Sub(\mathcal{B})$;
— $C \sqcap D \in Sub(\mathcal{B}) \Rightarrow \{C, D\} \subseteq Sub(\mathcal{B})$;
— $\forall R.C \in Sub(\mathcal{B}) \Rightarrow \{R, C\} \subseteq Sub(\mathcal{B})$;
— $\exists R \in Sub(\mathcal{B}) \Rightarrow R \in Sub(\mathcal{B})$.

We now fix $\Phi = Sub(\mathcal{A} \cup S)$. The inference system for $\mathcal{AL}$ is presented in Figure 1, in the form of tableau completion rules in which $\mathcal{B}$ is initialized as the ABox $\mathcal{A}$. The tableau algorithm, denoted by Λ, non-deterministically applies the completion rules until no further applications are possible.

$\sqcap_1$**-rule:** If $(C_1 \sqcap C_2)(a) \in \mathcal{B}$ and $\{C_1(a), C_2(a)\} \not\subseteq \mathcal{B}$,
then $\mathcal{B} := \mathcal{B} \cup \{C_1(a), C_2(a)\}$.
$\exists_1$**-rule:** If $\exists R(a) \in \mathcal{B}$ and there is no $b \in N_{\mathcal{O}}$ s.t. $R(a, b) \in \mathcal{B}$,
then $\mathcal{B} := \mathcal{B} \cup \{R(a, c)\}$ where $c \in N_{\mathcal{O}}$ is fresh.
$\forall$**-rule:** If $\{\forall R.C(a), R(a, b)\} \subseteq \mathcal{B}$ and $C(b) \notin \mathcal{B}$,
then $\mathcal{B} := \mathcal{B} \cup \{C(b)\}$.
$\sqcap_2$**-rule:** If $C_1 \sqcap C_2 \in \Phi$, $\{C_1(a), C_2(a)\} \subseteq \mathcal{B}$ and $(C_1 \sqcap C_2)(a) \notin \mathcal{B}$,
then $\mathcal{B} := \mathcal{B} \cup \{(C_1 \sqcap C_2)(a)\}$.
$\exists_2$**-rule:** If $\exists R \in \Phi$, there is $b \in N_{\mathcal{O}}$ s.t. $R(a, b) \in \mathcal{B}$ and $\exists R(a) \notin \mathcal{B}$,
then $\mathcal{B} := \mathcal{B} \cup \{\exists R(a)\}$.

Fig. 1. Completion rules for $\mathcal{AL}$

The first three rules break down assertions into smaller constituents whereas the last two rules construct compound assertions, all restricted to Φ. In reference to the $\exists_1$-rule, an individual $a \in N_{\mathcal{O}}$ is *fresh* if a has not been used before during the execution of Λ. An ABox is *closed* if it contains a *clash*, i.e., it contains $\{A(a), \neg A(a)\}$ or $\bot(a)$ for some $A \in N_{\mathcal{C}}$ and $a \in N_{\mathcal{O}}$. An ABox that is not closed is *open* and it is *completed* if none of the completion rules (in Figure 1) is applicable. With Φ defined as above, the ABox resulting from the application of Λ is denoted by $\mathcal{A}^{\Lambda}_{\Phi}$ and it is unique up to renaming of "fresh" individuals introduced by the $\exists_1$-rule.

THEOREM 4.4. *(Soundness of Λ) For all individuals a, b occurring in $\mathcal{A}$, $C(a) \in \mathcal{A}^{\Lambda}_{\Phi} \Rightarrow \mathcal{A} \vDash C(a)$ and $R(a,b) \in \mathcal{A}^{\Lambda}_{\Phi} \Rightarrow \mathcal{A} \vDash R(a,b)$.*

PROOF. Let $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ be an arbitrary model of $\mathcal{A}$. We prove the claim by induction on the construction of $\mathcal{A}^{\Lambda}_{\Phi}$. The base case is when no rule was applied yet. Since $\mathcal{I}$ is a model of $\mathcal{A}$, for every $C(a), R(a,b) \in \mathcal{A}$, $\mathcal{I} \vDash C(a)$ and $\mathcal{I} \vDash R(a,b)$. For the induction step, denote by $\mathcal{A}'$ ($\mathcal{A}''$) the ABox before (after) the application of a completion rule.

— If the $\sqcap_1$-rule is applied, then $(C_1 \sqcap C_2)(a) \in \mathcal{A}'$, $\{C_1(a), C_2(a)\} \not\subseteq \mathcal{A}'$ and $\{C_1(a), C_2(a)\} \subseteq \mathcal{A}''$. By IH, $\mathcal{I} \vDash (C_1 \sqcap C_2)(a)$ means $a^{\mathcal{I}} \in (C_1 \sqcap C_2)^{\mathcal{I}}_Y = (C_1)^{\mathcal{I}}_Y \cap (C_2)^{\mathcal{I}}_Y$, and so $\mathcal{I} \vDash C_1(a)$ and $\mathcal{I} \vDash C_2(a)$.
— Since the application of the $\exists_1$-rule always creates a fresh individual (that does not appear in $\mathcal{A}$, the claim holds true vacuously.
— If the $\forall$-rule is applied, then $\{\forall R.C(a), R(a,b)\} \subseteq \mathcal{A}'$, $C(b) \notin \mathcal{A}'$ and $C(b) \in \mathcal{A}''$. If a or b does not occur in $\mathcal{A}$, the claim holds. If a, b occur in $\mathcal{A}$, by IH, $\mathcal{I} \vDash \forall R.C(a)$ and $\mathcal{I} \vDash R(a,b)$, i.e., we have $a^{\mathcal{I}} \in \{d \in \Delta \mid \forall c : (d,c) \in R^{\mathcal{I}}_Y \to c \in C^{\mathcal{I}}_Y\}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}_Y$. It follows that $b^{\mathcal{I}} \in C^{\mathcal{I}}_Y$ and so $\mathcal{I} \vDash C(b)$.
— If the $\sqcap_2$-rule is applied, then $C_1 \sqcap C_2 \in \Phi$, $\{C_1(a), C_2(a)\} \subseteq \mathcal{A}'$, $(C_1 \sqcap C_2)(a) \notin \mathcal{A}'$ and $(C_1 \sqcap C_2)(a) \in \mathcal{A}''$. By IH, $\mathcal{I} \vDash C_1(a)$ and $\mathcal{I} \vDash C_2(a)$, i.e., $a^{\mathcal{I}} \in (C_1)^{\mathcal{I}}_Y$ and $a^{\mathcal{I}} \in (C_2)^{\mathcal{I}}_Y$ implying $a^{\mathcal{I}} \in (C_1)^{\mathcal{I}}_Y \cap (C_2)^{\mathcal{I}}_Y = (C_1 \sqcap C_2)^{\mathcal{I}}_Y$. It follows that $\mathcal{I} \vDash (C_1 \sqcap C_2)(a)$.
— If the $\exists_2$-rule is applied, then $\exists R \in \Phi$, $R(a,b) \in \mathcal{A}'$, $\exists R(a) \notin \mathcal{A}'$ and $\exists R(a) \in \mathcal{A}''$. If a does not occur in $\mathcal{A}$, the claim holds vacuously. If (a does and) b doesn't occur in $\mathcal{A}$, then b was freshly introduced by applying the $\exists_1$-rule either to the assertion $\exists R(a)$ or to an assertion $\exists R(c)$ for some $c \neq a$. The former case contradicts the assumption $\exists R(a) \notin \mathcal{A}'$; in the latter case, the $R(a,b)$ could not have been added to $\mathcal{A}'$, contradicting $R(a,b) \in \mathcal{A}'$. Finally, suppose that both a and b occur in $\mathcal{A}$. By IH, $\mathcal{I} \vDash R(a,b)$ and so $a^{\mathcal{I}} \in \{d \in \Delta \mid \exists b : (d,b) \in R^{\mathcal{I}}_Y\}$. Therefore, $a^{\mathcal{I}} \in (\exists R)^{\mathcal{I}}_Y$. i,e., $\mathcal{I} \vDash \exists R(a)$. □

To prove the completeness of Λ, we define a *canonical OW-interpretation* $\mathcal{J} = \langle \Delta, \cdot^{\mathcal{J}} \rangle$ for an open and completed ABox $\mathcal{A}^{\Lambda}_{\Phi}$ as follows:

— $\Delta := \{a \in N_{\mathcal{O}} \mid a \text{ occurs in } \mathcal{A}^{\Lambda}_{\Phi}\}$;
— $a^{\mathcal{J}} := a$ for all a occurring in $\mathcal{A}^{\Lambda}_{\Phi}$;
— for $A \in N_{\mathcal{C}} \cap \Phi$, $A^{\mathcal{J}} = (A^{\mathcal{J}}_N, A^{\mathcal{J}}_U, A^{\mathcal{J}}_Y)$ with $A^{\mathcal{J}}_U = (\Delta \setminus A^{\mathcal{J}}_Y) \setminus A^{\mathcal{J}}_N$ where $A^{\mathcal{J}}_Y := \{a \in \Delta \mid A(a) \in \mathcal{A}^{\Lambda}_{\Phi}\}$ and $A^{\mathcal{J}}_N := \{a \in \Delta \mid \neg A(a) \in \mathcal{A}^{\Lambda}_{\Phi}\}$;
— for $R \in N_{\mathcal{R}} \cap \Phi$, $R^{\mathcal{J}} = (\emptyset, \Delta \times \Delta \setminus R^{\mathcal{J}}_Y, R^{\mathcal{J}}_Y)$ where $R^{\mathcal{J}}_Y := \{(a,b) \in \Delta \times \Delta \mid R(a,b) \in \mathcal{A}^{\Lambda}_{\Phi}\}$;
— $\mathcal{J}$ is extended to Φ as indicated in Section 4.2.1.

COROLLARY 4.5. *For every role assertion $R(a,b)$ where $R \in \Phi$ and a, b occur in $\mathcal{A}^{\Lambda}_{\Phi}$, $R(a,b) \in \mathcal{A}^{\Lambda}_{\Phi} \Leftrightarrow \mathcal{J} \vDash R(a,b)$.*

The following lemma points out a relationship between the canonical OW-interpretation $\mathcal{J}$ and the completed ABox $\mathcal{A}^{\Lambda}_{\Phi}$. In fact, together with Corollary 4.5, it shows that $\mathcal{J}$ is a model of $\mathcal{A}^{\Lambda}_{\Phi}$, and hence a model of $\mathcal{A}$.

LEMMA 4.6. $C(a) \in \mathcal{A}_{\Phi}^{\Lambda} \Rightarrow \mathcal{J} \vDash C(a)$.

PROOF. The proof is by induction on the structure of concept expressions. The basis is when C is A or $\neg A$ where $A \in \Phi$. In these cases the implications follow from the definition of $\mathcal{J}$. The induction step includes the following cases:

— $C = C_1 \sqcap C_2$. Since $\mathcal{A}_{\Phi}^{\Lambda}$ is completed, by the $\sqcap_1$-rules, $C_1(a) \in \mathcal{A}_{\Phi}^{\Lambda}$ and $C_2(a) \in \mathcal{A}_{\Phi}^{\Lambda}$. By IH, $\mathcal{J} \vDash C_1(a)$ and $\mathcal{J} \vDash C_2(a)$. Hence, $a \in (C_1)_Y^{\mathcal{J}} \cap (C_2)_Y^{\mathcal{J}} = (C_1 \sqcap C_2)_Y^{\mathcal{J}}$. So $\mathcal{J} \vDash C_1 \sqcap C_2(a)$ and the claim holds.

— $C = \exists R$. Since $\mathcal{A}_{\Phi}^{\Lambda}$ is completed, by the $\exists_1$-rule, $\exists R(a) \in \mathcal{A}_{\Phi}^{\Lambda}$ implies that there is $b \in \Delta$ such that $R(a, b) \in \mathcal{A}_{\Phi}^{\Lambda}$. By Corollary 4.5, $R(a, b) \in \mathcal{A}_{\Phi}^{\Lambda} \Leftrightarrow \mathcal{J} \vDash R(a, b)$. Moreover, with $b \in \Delta$, $\mathcal{J} \vDash R(a, b) \Leftrightarrow (a^{\mathcal{J}}, b^{\mathcal{J}}) \in R_Y^{\mathcal{J}} \Leftrightarrow a^{\mathcal{J}} \in (\exists R)_Y^{\mathcal{J}} \Leftrightarrow \mathcal{J} \vDash \exists R(a)$. Therefore, the claim holds.

— $C = \forall R.C_1$. If $\forall R.C_1(a) \in \mathcal{A}_{\Phi}^{\Lambda}$, since $\mathcal{A}_{\Phi}^{\Lambda}$ is completed, by the $\forall$-rule, for every $b \in \Delta$, $R(a, b) \in \mathcal{A}_{\Phi}^{\Lambda}$ implies $C_1(b) \in \mathcal{A}_{\Phi}^{\Lambda}$. By Corollary 4.5, $R(a, b) \in \mathcal{A}_{\Phi}^{\Lambda} \Leftrightarrow \mathcal{J} \vDash R(a, b)$. By IH, we have $\mathcal{J} \vDash C_1(b)$. That is, for every $b \in \Delta$, $(a^{\mathcal{J}}, b) \in R_Y^{\mathcal{J}} \rightarrow b \in (C_1)_Y^{\mathcal{J}}$. Hence, $\mathcal{J} \vDash \forall R.C_1(a)$. □

For the proof of the completeness theorem, we will need the following auxiliary Lemma.

LEMMA 4.7. *For any* $\forall R.C \in \Phi$ *and any* a *that occurs in* $\mathcal{A}$*,* $\mathcal{A} \vDash \forall R.C(a) \Rightarrow \forall R.C(a) \in \mathcal{A}_{\Phi}^{\Lambda}$.

PROOF. Suppose $\psi = \forall R.C(a) \notin \mathcal{A}_{\Phi}^{\Lambda}$. From ψ and the canonical OW-interpretation $\mathcal{J} = \langle \Delta, \cdot^{\mathcal{J}} \rangle$ we will construct a new OW-interpretation $\mathcal{J}_{\psi}$ which, as we will show, is a model of $\mathcal{A}$ that does not satisfy ψ. The construction proceeds as follows:

Step 1 *Initialization*. We define a domain Δ' and a function $\mathcal{J}'$ as follows

— $\Delta' := \Delta \cup \{x\}$ where $x \notin \Delta$,

— $b^{\mathcal{J}'} := b^{\mathcal{J}}$ for all b occurring in $\mathcal{A}_{\Phi}^{\Lambda}$,

— $A^{\mathcal{J}'} := A^{\mathcal{J}}$ for all $A \in N_{\mathcal{C}} \cap \Phi$,

— $P^{\mathcal{J}'} := P^{\mathcal{J}}$ for all $P \in N_{\mathcal{R}} \cap \Phi$ and $P \neq R$,

— $R^{\mathcal{J}'} := (\emptyset, \Delta' \times \Delta' \setminus R_Y^{\mathcal{J}'}, R_Y^{\mathcal{J}'})$ where $= R_Y^{\mathcal{J}} \cup \{(a^{\mathcal{J}}, x)\}$.

Step 2 *Construction I*. For all $\forall R.D(a) \in \mathcal{A}_{\Phi}^{\Lambda}$ where $D \neq C$ and D is not of the form $\forall S.D'$,

— if $D \in N_{\mathcal{C}}$, then $D_Y^{\mathcal{J}'} := D_Y^{\mathcal{J}'} \cup \{x\}$;

— if $D = \neg A$ where $A \in N_{\mathcal{C}}$, then $A_N^{\mathcal{J}'} := A_N^{\mathcal{J}'} \cup \{x\}$;

— if D is of the form $\exists S$, then $\Delta' := \Delta' \cup \{x_S\}$ where x_S is new and let $S_Y^{\mathcal{J}'} := S_Y^{\mathcal{J}'} \cup \{(x, x_S)\}$.

Step 3 *Construction II*. Let $\Gamma = \{\forall R.\forall S.E(a) \in \mathcal{A}_{\Phi}^{\Lambda} \mid \forall S.E \neq C\}$. For every $\gamma \in \Gamma$, let $dep(\gamma)$ be the number of universal quantifiers in the prefix of γ. For example, $dep(F(b)) = dep(\neg F(b) = dep(\exists P(b)) = 0$ and $dep(\forall P.F(b)) = 1$ if $F \in N_{\mathcal{C}}$. Then extend the construction of $\mathcal{J}'$ using Algorithm 4.

Step 4 *Completion*. Let Δ_{ψ} be the Δ' after the Step 3.

— For every $A \in N_{\mathcal{C}} \cap \Phi$ and $z \in \Delta_{\psi} \setminus \Delta$, if $z \notin A_Y^{\mathcal{J}'} \cup A_U^{\mathcal{J}'} \cup A_N^{\mathcal{J}'}$, then $A_U^{\mathcal{J}'} := A_U^{\mathcal{J}'} \cup \{z\}$.

— For every $P \in N_{\mathcal{R}} \cap \Phi$, $P_U^{\mathcal{J}'} = \Delta_{\psi} \times \Delta_{\psi} \setminus P_Y^{\mathcal{J}'}$.

Note that in Step 2, by the definition of Φ, $\forall R.D \in \Phi \Rightarrow R, D \in \Phi$. In Step 3, at all times during the construction using Algorithm 4, Γ contains only assertions whose concept expressions are universally restricted. By the definition of $dep(\gamma)$ for $\gamma \in \Gamma$, an assertion $\forall P_1.\forall P_2.A(z)$ will be checked after an assertion $\forall P_1.\exists P_2(z)$. This guarantees that if z

```
while $\Gamma \neq \emptyset$ do
    Let $k$ be the current minimum $dep(\gamma)$ for $\gamma \in \Gamma$
    Let $\Gamma_k := \{\gamma \in \Gamma \mid dep(\gamma) = k\}$
    for $\forall P.F(y) \in \Gamma_k$ do
        if there is $y'$ such that $(y, y') \in P_Y^{\mathcal{J}'}$ then
            if $F \in N_\mathcal{C}$ then
                Let $F_Y^{\mathcal{J}'} := F_Y^{\mathcal{J}'} \cup \{y'\}$
            else if $F = \neg A$ where $A \in N_\mathcal{C}$ then
                Let $A_N^{\mathcal{J}'} := A_N^{\mathcal{J}'} \cup \{y'\}$
            else if $F = \exists Q$ then
                Let $\Delta' := \Delta' \cup \{y''\}$ where $y''$ is new
                Let $Q_Y^{\mathcal{J}'} := Q_Y^{\mathcal{J}'} \cup \{(y', y'')\}$
            else if $F = \forall Q.G$ then
                Let $\Gamma := \Gamma \cup \{F(y')\}$
            end
        end
        Let $\Gamma := \Gamma \setminus \{\forall P.F(y)\}$
    end
end
```

Algorithm 4: $\mathcal{J}_\psi$ construction Step 3

has a P_1-successor z_1, then a P_2-successor, say z_2, of z_1 will be created before checking the assertion $\forall P_1.\forall P_2.A(z)$ so that z_2 will be correctly put into the concept name A's interpretation. Let $\mathcal{J}_\psi$ be $\mathcal{J}'$ after completion of Step 4. It is easy to see that Δ_ψ is finite and that for every $A \in N_\mathcal{C} \cap \Phi$ and every $P \in N_\mathcal{R} \cap \Phi$, $A_Y^\mathcal{J} \subseteq A_Y^{\mathcal{J}_\psi}$, $A_N^\mathcal{J} \subseteq A_N^{\mathcal{J}_\psi}$ and $P_Y^\mathcal{J} \subseteq P_Y^{\mathcal{J}_\psi}$. Define $\mathcal{J}_\psi = \langle \Delta_\psi, \cdot^{\mathcal{J}_\psi} \rangle$. To show that $\mathcal{J}_\psi$ is an OW-interpretation, we need to show that for all $A \in N_\mathcal{C} \cap \Phi$, $A_Y^{\mathcal{J}_\psi} \cap A_N^{\mathcal{J}_\psi} = \emptyset$. Suppose that there is $y \in A_Y^{\mathcal{J}_\psi} \cap A_N^{\mathcal{J}_\psi}$ for some $A \in N_\mathcal{C} \cap \Phi$. Since $\mathcal{J}$ is a model, we have $y \in \Delta' \backslash \Delta$. Then from the above procedure, there are two assertions $\forall S_1. \cdots .\forall S_j.A(b) \in \mathcal{A}_\Phi^\Lambda$ and $\forall S_1. \cdots .\forall S_j.\neg A(b) \in \mathcal{A}_\Phi^\Lambda$, which contradicts the fact that $\mathcal{J}$ satisfies $\mathcal{A}_\Phi^\Lambda$ (see Corollary 4.5 and Lemma 4.6).

CLAIM 3. *$\mathcal{J}_\psi$ is a model of $\mathcal{A}$.*

Proof of the claim. We need to show that for every $E(b) \in \mathcal{A}$, $\mathcal{J}_\psi \vDash E(b)$ and for every $P(b, c) \in \mathcal{A}$, $\mathcal{J}_\psi \vDash P(b, c)$. Since $\mathcal{J}$ is a model of $\mathcal{A}$, for every $P(b, c) \in \mathcal{A}$, we have $(b^{\mathcal{J}_\psi}, c^{\mathcal{J}_\psi}) = (b^\mathcal{J}, c^\mathcal{J}) \in P_Y^\mathcal{J} \subseteq P_Y^{\mathcal{J}_\psi}$, and so $\mathcal{J}_\psi \vDash P(b, c)$. For $E(b) \in \mathcal{A}$, we have the following cases:

— $E = A \in N_\mathcal{C}$. Since $\mathcal{J}$ is a model of $\mathcal{A}$, $b^{\mathcal{J}_\psi} = b^\mathcal{J} \in A_Y^\mathcal{J} \subseteq A_Y^{\mathcal{J}_\psi}$, and so $\mathcal{J}_\psi \vDash A(b)$.
— $E = \neg A$ where $A \in N_\mathcal{C}$. Since $\mathcal{J}$ is a model of $\mathcal{A}$, $b^{\mathcal{J}_\psi} = b^\mathcal{J} \in (\neg A)_Y^\mathcal{J} = A_N^\mathcal{J} \subseteq A_N^{\mathcal{J}_\psi} = (\neg A)_Y^{\mathcal{J}_\psi}$. So $\mathcal{J}_\psi \vDash \neg A(b)$.
— $E = E_1 \sqcap E_2$. Since $\mathcal{J}$ is a model of $\mathcal{A}$, $b^{\mathcal{J}_\psi} = b^\mathcal{J} \in (E_1 \sqcap E_2)_Y^\mathcal{J} = (E_1)_Y^\mathcal{J} \cap (E_2)_Y^\mathcal{J} \subseteq (E_1)_Y^{\mathcal{J}_\psi} \cap (E_2)_Y^{\mathcal{J}_\psi} = (E_1 \sqcap E_2)_Y^{\mathcal{J}_\psi}$. So $\mathcal{J}_\psi \vDash E_1 \sqcap E_2(b)$.
— $E = \exists P$. Since $\mathcal{J}$ is a model of $\mathcal{A}$, $b^{\mathcal{J}_\psi} = b^\mathcal{J} \in (\exists P)_Y^\mathcal{J}$ if and only if there is $c \in \Delta$ such that $(b^\mathcal{J}, c) \in P_Y^\mathcal{J}$. Since $P_Y^\mathcal{J} \subseteq P_Y^{\mathcal{J}_\psi}$, we have $(b^\mathcal{J}, c) \in P_Y^{\mathcal{J}_\psi}$, and so $\mathcal{J}_\psi \vDash \exists R(b)$.

— $E = \forall P.D$. Since $\mathcal{J}$ is a model of $\mathcal{A}$, $\mathcal{J} \vDash \forall P.D(b)$, i.e., for every $c \in \Delta$, $(b^{\mathcal{J}}, c) \in P_Y^{\mathcal{J}} \Rightarrow c \in D_Y^{\mathcal{J}}$. For every $z \in \Delta_\psi \setminus \Delta$, if $(b^{\mathcal{J}}, z) \in P_Y^{\mathcal{J}_\psi}$, then we have the following cases: by the definition of $\mathcal{J}_\psi$ (specifically, see Algorithm 4),
  - — if $D \in N_{\mathcal{C}}$, then $z \in D_Y^{\mathcal{J}_\psi}$.
  - — if $D = \neg A$ where $A \in N_{\mathcal{C}}$, then $z \in A_N^{\mathcal{J}_\psi} = (\neg A)_Y^{\mathcal{J}_\psi}$.
  - — if $D = \exists S$, there is $z' \in \Delta_\psi$ such that $(z, z') \in S_Y^{\mathcal{J}_\psi}$, and so $z \in (\exists S)_Y^{\mathcal{J}_\psi}$.
  - — if $D = \forall S.D'$, we have $z \in (\forall S.D')_Y^{\mathcal{J}_\psi}$.

  Note that due to our normal form assumption for assertions in the ABox, D is not of the form $D_1 \sqcap D_2$. It follows that $z \in D_Y^{\mathcal{J}_\psi}$. Therefore, for every $c \in \Delta_\psi$, we have $(b^{\mathcal{J}_\psi}, c) \in P_Y^{\mathcal{J}_\psi} \Rightarrow c \in D^{\mathcal{J}_\psi}$, and so $\mathcal{J}_\psi \vDash \forall P.D(b)$. □

Because of our normal form assumption, for the assertion $\psi = \forall R.C(a)$, C is not a conjunction and the construction of $\mathcal{J}_\psi$ guarantees that $\mathcal{J}_\psi$ is not a model of $\forall R.C(a)$. Since $\mathcal{J}_\psi$ is a model of $\mathcal{A}$, we have $\mathcal{A} \nvDash \forall R.C(a)$. Thus, the claim that for any $\forall R.C \in \Phi$ and any a that occurs in $\mathcal{A}$, $\mathcal{A} \vDash \forall R.C(a) \Rightarrow \forall R.C(a) \in \mathcal{A}_\Phi^\Lambda$ holds. □

The following example illustrates how we construct a model to show that the ABox need not entail an assertion of the form $\forall R.C(a)$ that is not in the ABox.

Example 4.8. Suppose that we have an ABox $\mathcal{A} = \{\forall R.\exists S_1(a), \forall R.\exists S_2(a), \forall R.\forall S_1.A(a), \forall R.\forall S_2.\neg A(a), \forall R.C(b)\}$ and that the secrecy set $S = \emptyset$. Then $\Phi = \{R, S_1, S_2, A, \neg A, \exists S_1, \exists S_2, \forall R.\exists S_1, \forall R.\exists S_2, \forall R.\forall S_1.A, \forall R.\forall S_2.\neg A, \forall S_1.A, \forall S_2.\neg A, \forall R.C, C\}$. Since no completion rule is applicable to $\mathcal{A}$, $\mathcal{A}_\Phi^\Lambda = \mathcal{A}$. Let $\psi = \forall R.C(a)$.

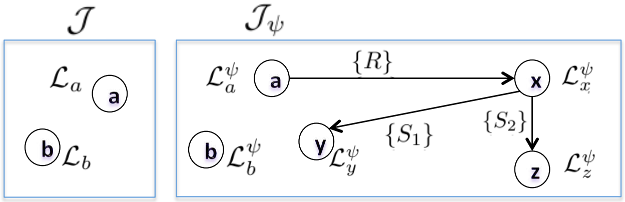


Fig. 2. $\mathcal{J}$ and $\mathcal{J}_\psi$ in Example 4.8

In Figure 2, we have the canonical model $\mathcal{J}$ for $\mathcal{A}_\Phi^\Lambda$ and the extension $\mathcal{J}_\psi$ of $\mathcal{J}$ where

- — $\mathcal{L}_a = \mathcal{L}_b = \{\forall R.\exists S_1, \forall R.\exists S_2, \forall R.\forall S_1.A, \forall R.\forall S_2.\neg A, \forall S_1.A, \forall S_2.\neg A, \forall R.C\}$,
- — $\mathcal{L}_a^\psi = \mathcal{L}_a \setminus \{\forall R.C\}$,
- — $\mathcal{L}_b^\psi = \mathcal{L}_b$,
- — $\mathcal{L}_x^\psi = \{\exists S_1, \exists S_2\} \cup \mathcal{L}_a$,
- — $\mathcal{L}_y^\psi = \{A\} \cup \mathcal{L}_a$,
- — $\mathcal{L}_z^\psi = \{\neg A\} \cup \mathcal{L}_a$.

We use the convention that for a concept expression $D \in \mathcal{C}$ and individual $c \in N_{\mathcal{O}}$, $D \in \mathcal{L}_c$ (respectively, $D \in \mathcal{L}_c^\psi$) means $c \in D_Y^{\mathcal{J}}$ (respectively, $c \in D_Y^{\mathcal{J}_\psi}$). It is easy to see that $\mathcal{J}_\psi$ is a model of $\mathcal{A}$. But since $C \notin \mathcal{L}_x^\psi$, $\mathcal{J}_\psi \nvDash \psi$. □

THEOREM 4.9. *(Completeness of Λ) Let $\mathcal{A}_\Phi^\Lambda$ be an open and completed ABox as defined above. Then for every $C, R \in \Phi$ and a, b occurring in $\mathcal{A}$, $\mathcal{A} \vDash C(a) \Rightarrow C(a) \in \mathcal{A}_\Phi^\Lambda$ and $\mathcal{A} \vDash R(a,b) \Rightarrow R(a,b) \in \mathcal{A}_\Phi^\Lambda$.*

PROOF. By Corollary 4.5, we have $\mathcal{A} \vDash R(a,b) \Rightarrow R(a,b) \in \mathcal{A}^{\Lambda}_{\Phi}$. Suppose that $\mathcal{A} \vDash C(a)$. We argue that $\mathcal{J} \vDash C(a) \Rightarrow C(a) \in \mathcal{A}^{\Lambda}_{\Phi}$ by induction on the structure of concept expressions. Since $\mathcal{J}$ is a model of $\mathcal{A}$, $\mathcal{J} \vDash C(a)$. The base case is when C is A or $\neg A$ where $A \in \Phi \cap N_C$. By the definition of the canonical model $\mathcal{J}$, we have $C(a) \in \mathcal{A}^{\Lambda}_{\Phi}$. The induction step includes the following cases.

- $C = C_1 \sqcap C_2$. Since $\mathcal{J} \vDash C_1 \sqcap C_2(a) \Leftrightarrow \mathcal{J} \vDash C_1(a) \wedge \mathcal{J} \vDash C_2(a)$, by IH, $\mathcal{J} \vDash C_i(a) \Rightarrow C_i(a) \in \mathcal{A}^{\Lambda}_{\Phi}$ for $i \in \{1,2\}$. Since $\mathcal{A}^{\Lambda}_{\Phi}$ is completed, by the $\sqcap_2$-rules, $(C_1 \sqcap C_2)(a) \in \mathcal{A}^{\Lambda}_{\Phi}$.
- $C = \exists R$. Since $\mathcal{J} \vDash \exists R(a) \Leftrightarrow$ there is $b \in \Delta$ such that $\mathcal{J} \vDash R(a,b)$. By Corollary 4.5, $R(a,b) \in \mathcal{A}^{\Lambda}_{\Phi}$. Since $\mathcal{A}^{\Lambda}_{\Phi}$ is completed, by the $\exists_2$-rule, $\exists R(a) \in \mathcal{A}^{\Lambda}_{\Phi}$.
- $C = \forall R.C_1$. Since $\mathcal{A} \vDash \forall R.C_1(a)$, it follows from Lemma 4.7 that $\forall R.C_1(a) \in \mathcal{A}^{\Lambda}_{\Phi}$. □

From Theorem 4.9 and the fact that there is no completion rule to generate $\forall R.C(a)$, we can see that one cannot conclude $\forall R.C(a)$ unless it is already in $\mathcal{A}^{\Lambda}_{\Phi}$. This is reasonable because, due to the OWA, the knowledge in the KB is incomplete, and so one cannot infer *all* the individuals that are R-successors of a.

4.2.3. Constructing Envelopes for $\mathcal{AL}$ Knowledge Bases. We have shown that the tableau algorithm Λ based on rules in Figure 1 is sound and complete. To construct an envelope, we invert these completion rules, similarly to Section 4.1. The resulting rules are listed in Figure 3. In this section, we assume that $\mathcal{A}^{\Lambda}_{\Phi}$ is open and completed.

$\sqcap^S_1$**-rule:** If $\{C_1(a), C_2(a)\} \cap E \neq \emptyset$ and $(C_1 \sqcap C_2)(a) \in \mathcal{A}^{\Lambda}_{\Phi} \setminus E$,
then $E := E \cup \{(C_1 \sqcap C_2)(a)\}$.
$\exists^S_1$**-rule:** If $R(a,b) \in E$ and $\exists R(a) \in \mathcal{A}^{\Lambda}_{\Phi} \setminus E$,
then $E := E \cup \{\exists R(a)\}$.
$\forall^S$**-rule:** If $C(b) \in E$ and $\{\forall R.C(a), R(a,b)\} \subseteq \mathcal{A}^{\Lambda}_{\Phi} \setminus E$,
then $E := E \cup \{\forall R.C(a)\}$.
$\sqcap^S_2$**-rule:** If $(C_1 \sqcap C_2)(a) \in E$ and $\{C_1(a), C_2(a)\} \cap E = \emptyset$,
then either $E := E \cup \{C_1(a)\}$ or $E := E \cup \{C_2(a)\}$.
$\exists^S_2$**-rule:** If $\exists R(a) \in E$ and $R(a,b) \in \mathcal{A}^{\Lambda}_{\Phi} \setminus E$,
then $E := E \cup \{R(a,b)\}$.

Fig. 3. Completion rules for computing envelopes

Note that the $\exists^S_1$-rule is used in the proof of Lemma 4.10. Also note that for the $\forall^S$-rule, we could have non-deterministically choosen $\forall R.C(a)$ or $R(a,b)$ for constructing E. However, we prefer $\forall R.C(a)$ since putting $R(a,b)$ into E may trigger an application of the $\exists^S_1$-rule which may potentially trigger application(s) of the $\exists^S_2$-rule and therefore leading to a larger envelope.

The algorithm that non-deterministically applies these rules is denoted by Λ_S. Due to the non-determinism in applying the $\sqcap^S_2$-rule, different executions of Λ_S may result in different sets. Given the set $\mathcal{A}^{\Lambda}_{\Phi}$ (obtained by applying Λ on $\mathcal{A}$ with $\Phi = Sub(\mathcal{A} \cup S)$), the set E is initialized as S and expanded with the execution of Λ_S. Since $\mathcal{A}^{\Lambda}_{\Phi}$ is finite, the computation of Λ_S terminates. Let E^1 be a resulting set (which will stay fixed for the remainder of this section). By the assumption that for any $\alpha \in S$, $\mathcal{A} \vDash \alpha$ (see Section 4.2.1), it follows from Theorem 4.9 that $S \subseteq \mathcal{A}^{\Lambda}_{\Phi}$, and hence, $E^1 \subseteq \mathcal{A}^{\Lambda}_{\Phi}$.

The next lemma shows that no assertion in the envelope is “logically reachable” from outside the envelope.

LEMMA 4.10. *The ABox $\mathcal{A}^{\Lambda}_{\Phi} \setminus E^1$ is open and completed.*

PROOF. We must show that no completion rule in Figure 1 is applicable to $\mathcal{A}^\Lambda_\Phi \setminus E^1$.

- If the $\sqcap_1$-rule is applicable, then there is an assertion $C \sqcap D(a) \in \mathcal{A}^\Lambda_\Phi \setminus E^1$ and $\{C(a), D(a)\} \nsubseteq \mathcal{A}^\Lambda_\Phi \setminus E^1$. Since $\mathcal{A}^\Lambda_\Phi$ is completed, $\{C(a), D(a)\} \subseteq \mathcal{A}^\Lambda_\Phi$. Hence, $\{C(a), D(a)\} \cap E^1 \neq \emptyset$. However, since none of the completion rules in Figure 3 is applicable to E^1, by the $\sqcap^S_1$-rule, $C \sqcap D(a) \in E^1$, yielding a contradiction.
- If the $\exists_1$-rule is applicable, then there exists $\exists R(a) \in \mathcal{A}^\Lambda_\Phi \setminus E^1$ and there is no $b \in N_\mathcal{O}$ such that $R(a,b) \in \mathcal{A}^\Lambda_\Phi \setminus E^1$. However, since $\mathcal{A}^\Lambda_\Phi$ is completed, there exists $c \in N_\mathcal{O}$ such that $R(a,c) \in \mathcal{A}^\Lambda_\Phi$, implying $R(a,c) \in E^1$. By the $\exists^S_1$-rule, $\exists R(a) \in E^1$, yielding a contradiction.
- If the $\forall$-rule is applicable, then $\{\forall R.C(a), R(a,b)\} \subseteq \mathcal{A}^\Lambda_\Phi \setminus E^1$ and $C(b) \notin \mathcal{A}^\Lambda_\Phi \setminus E^1$. Since $\mathcal{A}^\Lambda_\Phi$ is completed, we have $C(b) \in E^1$. However, since none of the completion rules in Figure 3 is applicable to E^1, by the $\forall^S$-rule, $\{\forall R.C(a), R(a,b)\} \cap E^1 \neq \emptyset$, yielding a contradiction.
- If the $\sqcap_2$-rule is applicable, then $C_1 \sqcap C_2 \in \Phi$, $\{C_1(a), C_2(a)\} \subseteq \mathcal{A}^\Lambda_\Phi \setminus E^1$ and $(C_1 \sqcap C_2)(a) \notin \mathcal{A}^\Lambda_\Phi \setminus E^1$. Since $\mathcal{A}^\Lambda_\Phi$ is completed, we have $(C_1 \sqcap C_2)(a) \in E^1$. However, since none of the completion rules in Figure 3 is applicable to E^1, by the $\sqcap^S_2$-rule, $\{C_1(a), C_2(a)\} \cap E^1 \neq \emptyset$, yielding a contradiction.
- If the $\exists_2$-rule is applicable, then $\exists R \in \Phi$, $R(a,b) \in \mathcal{A}^\Lambda_\Phi \setminus E^1$ and $\exists R(a) \notin \mathcal{A}^\Lambda_\Phi \setminus E^1$. Since $\mathcal{A}^\Lambda_\Phi$ is completed, we have $\exists R(a) \in E^1$. However, since none of the completion rules in Figure 3 is applicable to E^1, by the $\exists^S_2$-rule, $R(a,b) \in E^1$, yielding a contradiction.

Since no rule is applicable to $\mathcal{A}^\Lambda_\Phi \setminus E^1$, it is completed. Moreover, since $\mathcal{A}^\Lambda_\Phi$ is open, so is $\mathcal{A}^\Lambda_\Phi \setminus E^1$. □

The next theorem shows that the set E^1 resulting from Λ^S is a partial envelope for $\mathcal{S}$.

THEOREM 4.11. *E^1 is a partial envelope for $\mathcal{S}$ w.r.t. Φ.*

PROOF. We need to show that E^1 satisfies Axiom E1': for every $\alpha \in E^1 \subseteq \mathcal{A}^\Lambda_\Phi$, $\mathcal{A}^\Lambda_\Phi \setminus E^1 \not\models \alpha$. By Lemma 4.10, $\mathcal{A}^\Lambda_\Phi \setminus E^1$ is open and completed, so $(\mathcal{A}^\Lambda_\Phi \setminus E^1)^\Lambda_\Phi = \mathcal{A}^\Lambda_\Phi \setminus E^1$. By Theorem 4.9, for every $C, R \in \Phi$ and a, b occurring in $\mathcal{A}^\Lambda_\Phi \setminus E^1$, $\mathcal{A}^\Lambda_\Phi \setminus E^1 \models C(a) \Rightarrow C(a) \in \mathcal{A}^\Lambda_\Phi \setminus E^1$ and $\mathcal{A}^\Lambda_\Phi \setminus E^1 \models R(a,b) \Rightarrow R(a,b) \in \mathcal{A}^\Lambda_\Phi \setminus E^1$. Therefore, for every $\alpha \in E^1$, since $\alpha \notin \mathcal{A}^\Lambda_\Phi \setminus E^1$, $\mathcal{A}^\Lambda_\Phi \setminus E^1 \not\models \alpha$ and so Axiom E1' holds. □

4.2.4. Answering Queries. Recall that in Section 3.2, we discussed how a partial envelope can be developed and maintained. In Algorithm 2, for each querying agent $i \in \{1, ..., m\}$, E_i is initialized as S_i and for every assertion $\gamma \in E_i$, we check all proofs of γ, and for each such proof Γ, we "break" it by adding $\phi_\Gamma \in \Gamma$ to E_i. In the case of $\mathcal{AL}$, the completion rules for computing envelopes in Figure 3 in fact provide a procedure for breaking such proofs for any assertion in E_i. The query space $\mathcal{Q}'$ in Algorithm 2 corresponds to the set of subexpressions Φ that restricts the computation of both Λ and Λ_S.

After we have computed E^1_i for every $i \in \{1, ..., m\}$, let $E^*_i\big|_\Phi := \bigcup_{j:(M_i,M_j)\in\mathcal{E}} E^1_j$ for $1 \leq i \leq m$. By Lemma 3.3, $\mathbb{E}^*\big|_\Phi = \{E^*_1\big|_\Phi, ..., E^*_m\big|_\Phi\}$ is a partial envelope for $\mathcal{S}$ (w.r.t. Φ). When a querying agent M_i poses a query $C(a) \in \mathcal{A}^\Lambda_\Phi$ where $C \in \Phi$, it is answered "Yes" if $C(a) \in \mathcal{A}^\Lambda_\Phi \setminus (E^*_i\big|_\Phi)$, "No" if $\neg C(a) \in \mathcal{A}^\Lambda_\Phi \setminus (E^*_i\big|_\Phi)$ (only when $C \in N_\mathcal{C}$ in the case of $\mathcal{AL}$), and unknown otherwise.

However, if $C \notin \Phi$, the reasoner cannot conclude an answer with $\mathcal{A}^\Lambda_\Phi$ and $\mathbb{E}^*\big|_\Phi$ and more work is required. Let $\Phi' = \Phi \cup Sub(\{C(a)\})$ where $Sub(\{C(a)\})$ contains all the sub-expressions of C, see Definition 4.3. The set $\mathcal{A}^\Lambda_\Phi$ may not be completed w.r.t. Φ'.

Applying Λ on $\mathcal{A}^{\Lambda}_{\Phi}$, we obtain $\mathcal{A}^{\Lambda}_{\Phi'}$. Accordingly, some of the completion rules in Figure 3 may become applicable. For each $1 \leq i \leq m$, we expand $E_i^*\big|_{\Phi}$ until none of the completion rules in Figure 3 is applicable. The resulting set is denoted by E_i^2. By Corollary 3.4, E_i^2 is a partial envelope (w.r.t. Φ') for the induced single-agent secrecy structure S_i; by Lemma 3.3, $\mathbb{E}^*\big|_{\Phi'} = \{E_1^*\big|_{\Phi'}, ..., E_m^*\big|_{\Phi'}\}$ is a partial envelope for $\mathcal{S}$ (w.r.t. Φ') where $E_i^*\big|_{\Phi'} = \bigcup_{j:(M_i,M_j)\in\mathcal{E}} E_j^2$. With the sets $\mathcal{A}^{\Lambda}_{\Phi'}$ and $\mathbb{E}^*\big|_{\Phi'}$, queries over Φ' can be answered without revealing any secrets. Subsequent queries are treated similarly.

5. CONCLUSION

In this paper we have introduced a conceptual logic-based framework for secrecy-preserving reasoning for KBs in multiagent settings based on sound and complete proof systems. We have adapted the idea of secrecy envelopes (introduced in [Tao et al. 2010]) to this framework (Section 2) and have proved some interesting results about the structure of such envelopes (Section 2.1). We provided a single sweep bottom-up approach for constructing tight envelopes in the special case where the communication graph is an inverted forest, and have shown that this cannot be extended to DAGs. In practice, we build an initial partial envelope and update it as needed. We illustrated an application of this general approach in Propositional Horn KBs (Section 4.1) and Description Logic $\mathcal{AL}$ KBs (Section 4.2). In the case of $\mathcal{AL}$, the usual way of answering queries by checking satisfiability is not directly applicable since negation is allowed only in front of concept names and existential restriction is unqualified. Therefore, we utilized three-valued interpretation for answering queries under OWA. Such a semantics allows us to have a sound and complete proof system for answering queries directly rather than through satisfiability checking. Furthermore, since the rules for computing an envelope are obtained by inverting the rules of the sound and complete proof system, OW-interpretations are crucial in computing envelopes under OWA.

We have assumed that the secrecy sets are finite and given. It would be useful to consider cases where secrecy sets are (in principle) infinite, but have finite descriptions that can be expressed in a suitable policy language. We have also assumed that the queries and the KB are represented in the same language. It would be useful to consider query languages different from that of the KB, for example, allowing conjunctive queries. Another direction for future work is to study more general communication graphs. For instance, graphs that can place additional restrictions on answer-sharing, perhaps by attaching predicates to edges which could restrict the communication between querying agents.

REFERENCES

BAADER, F. 2009. Description logics. In *Reasoning Web: Semantic Technologies for Information Systems, 5th International Summer School 2009*. Lecture Notes in Computer Science Series, vol. 5689. Springer–Verlag, 1–39.

BAADER, F., CALVANESE, D., MCGUINNESS, D. L., NARDI, D., AND PATEL-SCHNEIDER, P. F., Eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

BAADER, F., KNECHTEL, M., AND PEÑALOZA, R. 2009. A generic approach for large-scale ontological reasoning in the presence of access restrictions to the ontology's axioms. In *International Semantic Web conference*, A. Bernstein, D. R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, and K. Thirunarayan, Eds. Lecture Notes in Computer Science Series, vol. 5823. Springer, 49–64.

BAADER, F. AND SATTLER, U. 2001. An overview of tableau algorithms for description logics. *Studia Logica 69,* 1, 5–40.

BAO, J., SLUTZKI, G., AND HONAVAR, V. 2007. Privacy-preserving reasoning on the semantic web. In *Web Intelligence*. IEEE Computer Society, 791–797.

BELL, D. E. AND LAPADULA, L. 1974a. Secure computer systems. Tech. Rep. ESD-TR-73-278, Vols. 1, 2, MITRE Corp., Bedford, MA.

BELL, D. E. AND LAPADULA, L. 1974b. Secure computer systems: A mathematical model. Tech. Rep. ESD-TR-73-278, Vol. 2, MITRE Corp., Bedford, MA.

BELL, D. E. AND LAPADULA, L. 1974c. Secure computer systems: Mathematical foundations. Tech. Rep. ESD-TR-73-278, Vol. 1, MITRE Corp., Bedford, MA.

BERTINO, E., KHAN, L. R., SANDHU, R. S., AND THURAISINGHAM, B. M. 2006. Secure knowledge management: confidentiality, trust, and privacy. *IEEE Transactions on Systems, Man, and Cybernetics, Part A 36,* 3, 429–438.

BISKUP, J. 2011. History-dependent inference control of queries by dynamic policy adaption. In *DBSec*. 106–121.

BISKUP, J., KERN-ISBERNER, G., AND THIMM, M. 2008. Towards enforcement of confidentiality in agent interactions. In *Proceedings of the 12th International Workshop on Non-Monotonic Reasoning (NMR08)*. 104–112.

BISKUP, J. AND TADROS, C. 2012. Revising belief without revealing secrets. *Foundations of Information and Knowledge Systems*, 51–70.

BISKUP, J., TADROS, C., AND WIESE, L. 2010. Towards controlled query evaluation for incomplete first-order databases. *Foundations of Information and Knowledge Systems*, 230–247.

BISKUP, J. AND WEIBERT, T. 2008. Keeping secrets in incomplete databases. *International Journal of Information Security 7,* 3, 199–217.

BONATTI, P. A., DUMA, C., FUCHS, N., NEJDL, W., OLMEDILLA, D., PEER, J., AND SHAHMEHRI, N. 2006. Semantic web policies - a discussion of requirements and research issues. In *ESWC*. 712–724.

BONATTI, P. A. AND OLMEDILLA, D. 2007. Rule-based policy representation and reasoning for the semantic web. In *Reasoning Web*. LNCS Series, vol. 4636. Springer, 240–268.

BONATTI, P. A. AND SAURO, L. 2013. A confidentiality model for ontologies. In *The Semantic Web–ISWC 2013*. Springer, 17–32.

BORGWARDT, S. AND PEÑALOZA, R. 2011. Description logics over lattices with multi-valued ontologies. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*. AAAI Press, 768–773.

DI VIMERCATI, S. D. C., SAMARATI, P., AND JAJODIA, S. 2005. Policies, models, and languages for access control. In *DNIS*, S. Bhalla, Ed. Lecture Notes in Computer Science Series, vol. 3433. Springer, 225–237.

DOWLING, W. F. AND GALLIER, J. H. 1984. Linear-time algorithms for testing the satisfiability of propositional horn formulae. *J. Log. Program. 1,* 3, 267–284.

GODIK, S. AND (ED.), T. M. 2002. Oasis extensible access control markup language (xacml). OASIS Committee Secification cs-xacml-specification-1.0, November 2002, http://www.oasis-open.org/committees/xacml/.

GOGUEN, J. A. AND MESEGUER, J. 1982. Security policies and security models. In *IEEE Symposium on Security and Privacy*. 11–20.

GRAY, J. W. AND SYVERSON, P. F. 1998. A logical approach to multilevel security of probabilistic systems. *Distributed Computing 11,* 2, 73–90.

HALPERN, J. Y. AND O'NEILL, K. R. 2008. Secrecy in multiagent systems. *ACM Trans. Inf. Syst. Secur. 12,* 1, 1–47.

HALPERN, J. Y. AND WEISSMAN, V. 2008. Using first-order logic to reason about policies. *ACM Trans. Inf. Syst. Secur. 11,* 4, 1–41.

HODKINSON, I., WOLTER, F., AND ZAKHARYASCHEV, M. 2000. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied logic 106,* 1, 85–134.

JAIN, A. AND FARKAS, C. 2006. Secure resource description framework: an access control model. In *SACMAT*. 121–129.

JAJODIA, S. 1996. Database security and privacy. *ACM Comput. Surv. 28,* 1, 129–131.

JAJODIA, S., SAMARATI, P., SAPINO, M. L., AND SUBRAHMANIAN, V. S. 2001. Flexible support for multiple access control policies. *ACM Trans. Database Syst. 26,* 2, 214–260.

KAGAL, L., BERNERS-LEE, T., CONNOLLY, D., AND WEITZNER, D. J. 2006. Using semantic web technologies for policy management on the web. In *AAAI*. AAAI Press.

KAGAL, L., FININ, T. W., AND JOSHI, A. 2003. A policy based approach to security for the semantic web. In *ISWC*. 402–418.

KAGAL, L., PAOLUCCI, M., SRINIVASAN, N., DENKER, G., FININ, T. W., AND SYCARA, K. P. 2004. Authorization and privacy for semantic web services. *IEEE Intelligent Systems 19,* 4, 50–56.

KOLOVSKI, V., HENDLER, J. A., AND PARSIA, B. 2007. Analyzing web access control policies. In *WWW*, C. L. Williamson, M. E. Zurko, P. F. Patel-Schneider, and P. J. Shenoy, Eds. ACM, 677–686.

MAKOWSKY, J. A. 1987. Why horn formulas matter in computer science: Initial structures and generic examples. *Journal of Computer and System Sciences 34,* 2-3, 266–292.

MCLEAN, J. 1992. Proving noninterference and functional correctness using traces. *Journal of Computer Security 1,* 1, 37–58.

OSBORN, S., SANDHU, R., AND MUNAWER, Q. 2000. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Trans. Inf. Syst. Secur. 3,* 2, 85–106.

SCHMIDT-SCHAUSS, M. AND SMOLKA, G. 1991. Attributive concept descriptions with complements. *Artif. Intell. 48,* 1, 1–26.

SHANNON, C. 1949. Communication theory of secrecy systems. *Bell Systems Technical Journal 28*, 656–715.

SICHERMAN, G. L., DE JONGE, W., AND VAN DE RIET, R. P. 1983. Answering queries without revealing secrets. *ACM Trans. Database Syst. 8,* 1, 41–59.

STAAB, S. AND STUDER, R. 2009. *Handbook on ontologies*. Springer.

STRACCIA, U. 2006. Description logics over lattices. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 14,* 01, 1–16.

SUTHERLAND, D. 1986. A model of information. In *Proceedings of the 9th National Computer Security Conference*. Vol. 247. 175–183.

TAO, J., SLUTZKI, G., AND HONAVAR, V. 2010. Secrecy-preserving query answering for instance checking in $\mathcal{EL}$. In *RR*. 195–203.

TONTI, G., BRADSHAW, J. M., JEFFERS, R., MONTANARI, R., SURI, N., AND USZOK, A. 2003. Semantic web languages for policy representation and reasoning: A comparison of kaos, rei, and ponder. In *ISWC*. 419–437.

WEITZNER, D. J., ABELSON, H., BERNERS-LEE, T., FEIGENBAUM, J., HENDLER, J. A., AND SUSSMAN, G. J. 2008. Information accountability. *Commun. ACM 51,* 6, 82–87.

WEITZNER, D. J., HENDLER, J., BERNERS-LEE, T., AND CONNOLLY, D. 2005. Creating the policy-aware web: Discretionary, rules-based access for the world wide web. In *Elena Ferrari and Bhavani Thuraisingham (editors), Web and Information Security, Hershey, PA: Idea Group Inc*.