Secrecy-Preserving Reasoning using Secrecy Envelopes

Giora Slutzki, George Voutsadakis and Vasant Honavar
Department of Computer Science
Iowa State University
Ames, IA 50011, USA

Abstract

In many applications of networked information systems, the need to share information often has to be balanced against the need to protect secret information from unintended disclosure, e.g., due to copyright, privacy, security, or commercial considerations. We study the problem of *secrecy-preserving reasoning*, that is, answering queries using secret information, whenever it is possible to do so, without compromising secret information. In the case of a knowledge base that is queried by a single querying agent, we introduce the notion of a secrecy envelope. This is a superset of the secret part of the knowledge base that needs to be concealed from the querying agent in order to ensure that the secret information is not compromised. We establish several important properties of secrecy envelopes and present an algorithm for computing minimal secrecy envelopes. We extend our analysis of secrecy preserving reasoning to the setting where different parts of the knowledge base need to be protected from different querying agents that are subject to certain restrictions on the sharing of answers supplied to them by the knowledge base.

1 Introduction

The rapid expansion of the Internet and the widespread adoption and use of distributed databases and networked information systems offer unprecedented opportunities for productive interaction and collaboration among autonomous individuals and across organizations in virtually every area of human endeavor. However, the need to share information (e.g., advance warning of an impending terrorist attack provided by FBI to a friendly nation) often has to be balanced against the need to protect sensitive or confidential information (e.g., the particular pieces of intelligence used to infer the likelihood of an attack on a specific target, the likely attackers, or the specific sources that were relied on to gather such information) from unintended disclosure. One can envision similar need for selective sharing of information arising from privacy, security, or commercial considerations in scenarios that involve interactions among different governmental agencies (e.g., intelligence, law enforcement, public policy), or independent nations acting on matters of global concern (e.g., counter-terrorism, international finance), and participants in business transactions (e.g., healthcare, insurance). Consequently, problems of trust, privacy and security in information systems in general, and networked information systems (e.g., the web), in particular, are topics of significant current interest.

Early work on information protection focused on access control mechanisms (see [Bertino *et al.*, 2006] for a survey). For, instance, work on *policy languages* for the web [Bonatti *et al.*, 2006; Kolovski *et al.*, 2007; Kagal *et al.*, 2006] involves specifying syntax-based restrictions on access to specific resources or operations on the web. Giereth [Giereth, 2005] has studied the hiding of a fragment of an RDF document by encrypting it while the rest of the document remains publicly readable. Farkas et al. [Farkas *et al.*, 2006; Jain and Farkas, 2006] have proposed a *privacy information flow model* to prevent unwanted inferences in data repositories. Jain and Farkas [Jain and Farkas, 2006] have proposed an RDF authorization model that can selectively control access to stored RDF triples using a pre-specified set of *syntactic* rules. In a recent paper [Cuenca Grau and Horrocks, 2008] Grau and Horrocks have introduced a framework that combines logic and probabilistic approaches to guarantee privacy preservation. A growing body of work on data linkage [O'Keefe *et al.*, 2004] addresses the problem of disclosure of personal data from aggregate information or from separately released, non-confidential information about an individual. Recent research on privacy preserving data mining [Clifton *et al.*, 2002] addresses the design of algorithms for constructing predictive models that describe shared characteristics of groups of individuals, e.g., patients in a clinical trial, without revealing information about specific individuals, e.g., clinical records of individual participants in the clinical trial.

Most of the existing methods for the protection of secret information rely on forbidding access to the sensitive parts of a knowledge base. Such approaches can be overly restrictive in scenarios where it is possible, and may be desirable, for a knowledge base to use secret knowledge to answer queries without risking disclosure of the secret knowledge [Bao *et al.*, 2007]. This calls for algorithms for *secrecy-preserving reasoning*, that is, answering queries using secret information, whenever it is possible to do so, without compromising secret information. Against this background, we introduce the notion of a secrecy envelope, that is, a superset of the secret part of the knowledge base that needs to be concealed by the knowledge base from the querying agent in order to ensure that the secret information is not compromised. We establish several important properties of secrecy envelopes and present an algorithm for computing minimal secrecy envelopes. We extend our analysis of secrecy preserving reasoning to the setting where different parts of the knowledge base need to be protected from different querying agents that are subject to certain restrictions on the sharing of answers supplied to them by the knowledge base.

2 The General Setting: Knowledge Bases and Reasoners

Let $\mathcal{E} = \langle X, \mathcal{R} \rangle$ be an entailment system[Voutsadakis *et al.*, 2008], e.g. a description logic, with consequence relation $\vdash_{\mathcal{E}}$. We denote by $Z^+ = \{x \in X : Z \vdash_{\mathcal{E}} x\}$, the $\mathcal{E}$-deductive closure of $Z \subseteq X$. $\mathcal{E}$ will be assumed fixed in what follows and, even though many of the concepts encountered will be relative to $\mathcal{E}$, this fact will not always be made explicit.

A **knowledge base** $\mathbf{K} = \langle K, B, Q, A \rangle$ **over** $\mathcal{E}$ consists of

1. A finite set $K \subseteq X$, which represents the knowledge contained in $\mathbf{K}$;
2. A finite subset B of K, representing the **browsable knowledge** that the querying agent has unrestricted access to;
3. A query set $Q \subseteq X$;
4. An answer set A, which is, usually either $\{Y, U\}$ or $\{Y, N, U\}$, for YES, NO and UNKNOWN.

Additionally, $\mathbf{K}$ has a subset $S \subseteq K^+$, the **secret** or **secrecy set**, which the knowledge base needs to keep secret from the querying agent. We assume that the querying agent has available a reasoner for the entailment system $\mathcal{E}$. Thus, since the agent can browse B, S has to satisfy the condition $B^+ \cap S = \emptyset$.

Let $\mathbf{K} = \langle K, B, Q, A \rangle$ be a knowledge base over an entailment system $\mathcal{E} = \langle X, \vdash_{\mathcal{E}} \rangle$. Given a function $R : Q \to \{Y, N, U\}$, we use the following notational conventions:

$$Q_Y = \{x \in Q : R(x) = Y\}$$

and, similarly, for Q_U (and for Q_N, in case the language has negation). A computable function $R : Q \to \{Y, N, U\}$ is a **reasoner** for $\mathbf{K} = \langle K, B, Q, A \rangle$ if it satisfies

1. **Interderivability Axiom:** If $x \vdash_{\mathcal{E}} y$ and $y \vdash_{\mathcal{E}} x$, then $R(x) = R(y)$, for all $x, y \in X$;
2. **Yes-Axiom:** $B^+ \subseteq Q_Y \subseteq K^+$;
3. **No-Axiom:** $Q_N = \{\neg x : x \in Q_Y\}$, in case X is a language that includes a logical negation.

A reasoner R for $\mathbf{K}$ is a **secrecy-preserving reasoner** if

$$Q_Y^+ \cap S = \emptyset. \quad (1)$$

The Interderivability Axiom ensures that two $\mathcal{E}$-equivalent queries are always answered the same way. The Yes-Axiom requires that all consequences of the browsable part in the knowledge base $\mathbf{K}$ are answered positively to the querying agent and that every positively answered query is a consequence of the information contained in the knowledge set K. On the other hand, the No-Axiom asserts that all negations of queries with positive answers have negative answers and, therefore, all negations of queries with negative answers have positive answers. Finally, Condition (1) asserts that the querying agent cannot discover information in its secret set given information in the set of queries with positive answers (including browsable sets, by the Yes-Axiom).

3 Security Envelopes

Let $\mathbf{K} = \langle K, B, Q, A \rangle$ be a knowledge base. Given any $Q' \subseteq Q$, we say that Q' is **inferentially closed** if $Q'^+ \cap Q = Q'$. Note that, assuming $Q = Q^+$, the inferential closure requirement reduces to $Q'^+ = Q'$. A $\mathbf{K}$-reasoner R is **inferentially closed** if Q_Y is inferentially closed, i.e. any consequence of a finite set of Y-queries is a Y-query. If a set $S \subseteq K^+ \backslash B^+$ is to be protected by a $\mathbf{K}$-reasoner R, we must have $S \subseteq Q_U$. This, however, may not be enough: It is likely that knowledge outside of S could be elicited by the querying agent and, in turn, be used to deduce information in S. To prevent this, the $\mathbf{K}$-reasoner must compute a possibly larger subset E_S, $S \subseteq E_S \subseteq K^+ \backslash B^+$, and this set should satisfy

Secrecy-Set Axiom: $(K^+ \backslash E_S)^+ \cap S = \emptyset$.

This Axiom is equivalent to Condition (1), if one takes $Q_Y = K^+ \backslash E_S$, i.e., every query not in E_S is answered positively (which is necessarily the case when $A = \{Y, U\}$). A reasoner satisfying this axiom is said to be a **secrecy-preserving K-reasoner** (w.r.t. the secrecy-set S). The set E_S, which is not necessarily unique, is called a **security** or **secrecy envelope** (or just an **envelope**) of S. Envelope sets E_S always exist (e.g., the set $K^+ \backslash B^+$ is one such set), but, in order for the reasoner to be as useful, i.e., as informative, as possible, the $\mathbf{K}$-reasoner should rather aim to have E_S as small as possible.

We say that an envelope E_S is a **tight envelope** if it is irredundant in the sense that removing any query from E_S (i.e., answering it with Y instead of U (and adapting the N answers accordingly, if the language has negation), would compromise the secrecy of S. In other words, tight envelopes must satisfy the following condition:

- **Tight Envelope Property**:

  $(\forall \alpha \in E_S)(\exists F \subseteq_f K^+ \backslash E_S)[(F \cup \{\alpha\})^+ \cap S \neq \emptyset]$,

  where $\subseteq_f$ denotes "finite subset".

Clearly, tight envelopes are precisely those that are minimal with respect to set inclusion. A tight envelope E_S of S is said to be **optimal** if it has the smallest cardinality among all tight envelopes of S. If the smallest possible (and hence optimal) envelope for a set S is S itself (which, of course, is not always the case), then S satisfies

- **Strong Secrecy–Set Axiom**: $(K^+ \backslash S)^+ \cap S = \emptyset$.

Given a knowledge base $\mathbf{K}$, let $\mathcal{F}_{\mathbf{K}}$ be the collection of all sets that satisfy the strong secrecy-set axiom, namely

$$\mathcal{F}_{\mathbf{K}} = \{S \subseteq K^+ \backslash B^+ : (K^+ \backslash S)^+ \cap S = \emptyset\}$$

and note that $\emptyset, K^+ \backslash B^+ \in \mathcal{F}_{\mathbf{K}}$. The following proposition gives a precise characterization of $\mathcal{F}_{\mathbf{K}}$.

Proposition 1 *$S \in \mathcal{F}_{\mathcal{K}}$ iff $K^+ \backslash S$ is inferentially closed.*

Proof: First suppose that $K^+ \backslash S$ is inferentially closed i.e. $(K^+ \backslash S)^+ = K^+ \backslash S$. Then,

$$\emptyset = (K^+ \backslash S) \cap S = (K^+ \backslash S)^+ \cap S$$

implying $S \in \mathcal{F}_{\mathbf{K}}$. Conversely, suppose that $S \in \mathcal{F}_{\mathbf{K}}$, i.e. $(K^+ \backslash S)^+ \cap S = \emptyset$, and let $\alpha \in (K^+ \backslash S)^+$. We need to show that $\alpha \in K^+ \backslash S$. Since clearly $\alpha \in K^+$, it suffices to show that $\alpha \notin S$. But $\alpha \in (K^+ \backslash S)^+$, so membership in S would lead to a contradiction. □

The following observation is interesting and easy to prove.

Lemma 2 *$\mathcal{F}_{\mathbf{K}}$ is closed under arbitrary unions.*

Proof: Let $A_i \in \subseteq \mathcal{F}_{\mathbf{K}}, i \in I$. Then $(K^+ \backslash A_i)^+ \cap A_i = \emptyset$, for all $i \in I$. Therefore

$$\begin{array}{ll} (K^+ \backslash \bigcup_{i \in I} A_i)^+ \cap \bigcup_{i \in I} A_i & \\ = & \bigcup_{i \in I} [(K^+ \backslash \bigcup_{i \in I} A_i)^+ \cap A_i] \\ \subseteq & \bigcup_{i \in I} ((K^+ \backslash A_i)^+ \cap A_i) \\ = & \emptyset, \end{array}$$

whence $\bigcup_{i \in I} A_i \in \mathcal{F}_{\mathbf{K}}$. □

The next lemma reveals a connection between tight envelopes and $\mathcal{F}_{\mathbf{K}}$. Note, however, that every $S \in \mathcal{F}_{\mathbf{K}}$ is its own, and hence optimal, envelope.

Lemma 3 *If $S \subseteq E_S \subseteq K^+ \backslash B^+$ is a tight envelope of S, then $E_S \in \mathcal{F}_{\mathbf{K}}$.*

Proof: By definition, the hypothesis implies $(K^+ \backslash E_S)^+ \cap S = \emptyset$. Our task is to show that $(K^+ \backslash E_S)^+ \cap E_S = \emptyset$. Suppose there is an $\alpha \in X$ which satisfies (i) $\alpha \in E_S \backslash S$ and (ii) $\alpha \in (K^+ \backslash E_S)^+$. From (ii) it follows that there is a subset $F \subseteq_f K^+ \backslash E_S$ such that $\alpha \in F^+$. On the other hand, since E_S is a tight envelope, there is a subset $G \subseteq_f K^+ \backslash E_S$ such that for some $\beta \in S$ and $\beta \in (G \cup \{\alpha\})^+$. Define the set $H = F \cup G$. Then $H \subseteq_f K^+ \backslash E_S$ and we have $\beta \in (G \cup \{\alpha\})^+ \subseteq H^+ \subseteq (K^+ \backslash E_S)^+$ which contradicts our hypothesis. □

4 Computing Security Envelopes

There are some important computational problems related to envelopes of secrecy-sets. Given a knowledge base $\mathbf{K} = \langle K, B, Q, A \rangle$ and $S \subseteq_f K^+$, a $\mathbf{K}$-reasoner has to calculate a security envelope for S, preferably a tight one or even optimal. Some questions relevant to these computational tasks are the following (all in the context of a fixed and given knowledge base $\mathbf{K}$ and secrecy set S): Does S satisfy the Strong Secrecy-Set Axiom? Is a given $\mathbf{K}$-reasoner secrecy preserving? Is a given $F \subseteq K^+$ a tight envelope? One of the more interesting questions is how to efficiently compute tight security envelopes for given secrecy sets, especially in restricted types of knowledge bases, e.g. hierarchical (i.e., partial order) or propositional knowledge bases.

Below we give a general "lazy" approach that a $\mathbf{K}$-reasoner R may adopt: wait for the queries and when one comes along, figure out how to answer it so that no information about the secrecy set S is revealed, taking into account R's answers to the queries asked up to this point. This is the greedy heuristic with the greediness criterion being the local concern of making sure that the secrecy set is not compromised at this point of time, without giving any consideration as to how the current response may constrain R's answers to future queries. Clearly, this approach produces a history-dependent $\mathbf{K}$-reasoner, i.e., different query histories may yield different $\mathbf{K}$-reasoners. We concentrate on the construction of the Y, N, U answer sets and the envelope E_S of S rather than on the equivalent task of providing the responses $R(\alpha)$ for an incoming query α.

Lazy Reasoner Algorithm (LRA)
input S
$X_Y \leftarrow B^+;\ X_U \leftarrow E_S \leftarrow S;\ X_N \leftarrow \neg X_Y$
while *TRUE* ***do***
  input α
  if $\alpha \notin Q$ ***then ERROR & ignore*** α
  else
    if $\alpha \notin X_Y \cup X_U \cup X_N$ ***then***
      if $\alpha \in K^+$ ***then***
        if $(X_Y \cup \{\alpha\})^+ \cap S \neq \varnothing$
        then $X_U \leftarrow X_U \cup \{\alpha, \neg\alpha\}; E_S \leftarrow E_S \cup \{\alpha\}$
        else $X_Y \leftarrow X_Y \cup \{\alpha\}; X_N \leftarrow X_N \cup \{\neg\alpha\}$
      else
        if $\alpha \notin \neg K^+$ ***then*** $X_U \leftarrow X_U \cup \{\alpha, \neg\alpha\}$
        else $X_Y \leftarrow X_Y \cup \{\neg\alpha\}; X_N \leftarrow X_N \cup \{\alpha\}$

It should be clear that LRA defines a $\mathbf{K}$-reasoner in that all relevant axioms are upheld at all times during the execution. The algorithm is equivalent to an algorithm presented in [Voutsadakis *et al.*, 2008], whose execution is guided by a fixed ordering of the set of queries. It was proved in [Voutsadakis *et al.*, 2008] that the algorithm produces a secrecy-preserving $\mathbf{K}$-reasoner for S, which is also maximally informative. This means that it provides a tight envelope E_S for S. This algorithm will be generalized in Section 5 to deal with the case of multiple agents querying a single knowledge base. Each agent has, in general, a different browsable set and a difference secrecy set from those of other agents.

4.1 Security Envelopes in Hierarchical Knowledge Bases

We now focus on the task of computing tight (or optimal) envelopes in the restricted, yet practically important case of hierarchical (i.e., partial order) knowledge bases. In this context the knowledge base is a finite directed (acyclic) graph $G = (V, E)$, where the vertex set V represents the elements of the given poset and the arcs in E represent a partial order. Of particular interest are hierarchical knowledge bases that are also transitive, i.e., transitive DAGs (TDAGs), e.g., the familiar "is-a" and "part-of" hierarchies.

The inferential closure of the given ontology G is its transitive closure $G^+ = (V, E^+)$. The set of queries is $Q =$

$V \times V$ and the browsable part of G is empty. Finally, the answer space is $A = \{Y, U\}$ because the underlying language does not have negation. A secrecy-set is a subset of edges $S \subseteq E^+$ and our goal is to find a tight security envelope: $S \subseteq E_S \subseteq E^+$ satisfying $(E^+ \backslash E_S)^+ \cap S = \emptyset$. Graph theoretically, this amounts to the following: Suppose $S = \{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$ are the secret edges in E^+. The goal is to find a superset of S, as small as possible, whose removal will disrupt all the paths from s_i to t_i, for $i = 1, 2, \ldots, k$. In the graph theory literature this problem is often referred to as the **multicut problem**. The decision version of this problem is the following

Directed Multicut

Instance: Directed graph $G = (V, E)$, list $S = \{(s_1, t_1), \ldots, (s_k, t_k)\} \subseteq V \times V$ and an integer $M \geq 0$.

Question: Is there $F \subseteq E$, of at most *M* edges, whose removal disconnects all the pairs in S?

Unfortunately, it turns out that this problem is NP-complete even for acyclic graphs [Calinescu *et al.*, 2003; Bentz, 2008] and, even worse, its optimization version of finding the smallest multicut is hard to approximate [Leighton and Rao, 1999; Garg *et al.*, 1997; Chawla *et al.*, 2005; Cheriyan *et al.*, 2001; Chuzhoy and Khanna, 2006]. The question as to whether there exists a polynomial time algorithm for computing an optimal security envelope for TDAG-structured knowledge bases remains open.

We give two simple heuristics, the first based on a Max Flow-Min Cut algorithm and the second on computing reachability.

Example 1: Let $G = (V, E)$ be a DAG and suppose that the set of secret edges is $S = \{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\} \subseteq E^+$. We first present an algorithms based on computing minimum cuts. The algorithm does compute a multicut, but it is not guaranteed to output a tight multicut.

1. **Simple MultiCut Algorithm (SMC)**
2. $H \leftarrow G$
3. ***for*** $i = 1$ ***to*** k ***do***
4. $\quad C_i \leftarrow$Min-Cut(H,s_i,t_i)
5. $\quad H \leftarrow H \backslash C_i$
6. $C \leftarrow C_1 \cup C_2 \cup \ldots \cup C_k$

Clearly, the algorithm runs in polynomial time and $H = G \backslash C$ at termination. It is also easy to see that the set *C* computed by the SMC algorithm is a multicut of *G* with respect to *S*. I.e., in the graph *H* (at termination) there is no path from s_i to t_i, $i = 1, 2, ..., k$. Thus, *C* is a security envelope of *S*. Unfortunately, the envelope *C* need not be tight; this is essentially because later cuts may have edges that make edges in previous cuts redundant. For instance, in the graph $G = (\{s_1, s_2, t_1, t_2\}, \{(s_1, s_2), (s_2, t_2), (t_2, t_1)\})$, the first cut could be $C_1 = \{(s_1, s_2)\}$ whereas the second cut must then be $C_2 = \{(s_2, t_2)\}$. As a result, we obtain the multicut $C = C_1 \cup C_2 = \{(s_1, s_2), (s_2, t_2)\}$ which clearly is not tight.

Consider next the transitive closure of *G*. In this case, the first min-cut C_1 will definitely not include the edge (s_2, t_2) and will have to have three edges (either all those leaving s_1 or all those entering t_1), say $C_1 = \{(s_1, t_1), (s_1, s_2), (s_1, t_2)\}$. The second cut will still have to be $C_2 = \{(s_2, t_2)\}$. The union of these two cuts is, in fact, a tight envelope; indeed, it is optimal. □

Example 2: The second algorithm is based on repeatedly computing the reachability sets for the vertices s_i within the given directed (acyclic) graph:

1. **Simple Reachability Algorithm** (**SRA**)
2. $H \leftarrow G$
3. ***for*** $i = 1$ ***to*** k ***do***
4. $\quad R_i \leftarrow \{(u, t_i) \in E \mid u \in V \text{ is reachable from } s_i\}$
5. $\quad H \leftarrow H \backslash R_i$
6. $R \leftarrow R_1 \cup R_2 \cup \ldots \cup R_k$

I.e., for each vertex *u* reachable from s_i we remove the edge (u, t_i) from the graph and place it in *R* (only if it does exist in the graph *H*). The set *R* resulting at termination is an envelope of *S*. Applying SRA to the graph *G* from the previous example results in the envelope $R = \{(t_2, t_1), (s_2, t_2)\}$ which is not tight. On the other hand, applying SRA to the transitive closure of *G* yields the envelope $R = \{(s_1, t_1), (s_2, t_1), (t_2, t_1), (s_2, t_2)\}$ which actually is optimal. The SRA algorithm is not guaranteed to produce an optimal envelope in the case of TDAG-structured knowledge bases. To see this let $G = (\{s_1, s_2, t_1, t_2\}, \{(s_1, s_2), (s_1, t_1), (s_1, t_2), (s_2, t_1), (s_2, t_2), (t_1, t_2)\})$. The algorithm will output the edges entering the terminals: $R_1 = \{(s_1, t_1), (s_2, t_1)\}$ and $R_2 = \{(s_2, t_2), (t_1, t_2)\}$. The union of these two cuts is not tight because the edge (t_1, t_2) is redundant. □

5 Multi-Agent Secrecy-Preserving Reasoning

The discussion so far has focused on the restricted case of a knowledge base that is queried by a single querying agent. We now extend our analysis to a knowledge base that can be queried by multiple querying agents. We first note that when

(a) the agents are forbidden from sharing with each other the answers supplied to them by the knowledge base, or

(b) the secrecy sets for all querying agents are identical, or

(c) the querying agents are allowed to freely share the answers supplied to them by the knowledge base,

the setting with multiple querying agents poses no new challenges beyond those encountered in the setting with a single querying agent. Hence, we assume that the knowledge base can have different secrecy sets for different querying agents, and that the querying agents are subject to some restrictions on the sharing of the answers supplied to them by the knowledge base. This is intended to model practical scenarios where there are legal restrictions on sharing of information across different organizations. The main idea behind our approach in this section will be to assume that there is no *external* communication between the querying agents at all, but that a "communication graph" is *internally* stored in the knowledge base and the **K**-reasoner shares answers to queries between the agents "depending on the edges" of the communication graph.

Let $G = \langle V, E\rangle$ be a directed graph, called the **communication graph**, whose nodes represent the querying agents and whose edges represent "a way" in which answers to queries are to be passed (or shared) between the querying agents. Let $\mathbf{K} = \langle K, \{B_v\}_{v\in V}, Q, A\rangle$ be a knowledge base with a secrecy set S_v, for each $v \in V$. Consider a corresponding knowledge base $\mathbf{K}_v = \langle K, B_v, Q, A\rangle$ and a reasoner $R_v : Q \to A$, with a security envelope E_v for S_v, as discussed in Sections 2 and 3. We use the following notation: $Q_Y^v = R_v^{-1}(Y) = K^+\backslash E_v$; this is precisely the set of all Y-queries of the $\mathbf{K}_v$-reasoner R_v.

The goal of the $\mathbf{K}$-reasoner in a multiple querying-agent environment is to prevent u from figuring out formulas in S_u, for all $u \in V$; however, it is quite possible, depending on the protocol being used, that u might figure out formulas in E_v, for $v \neq u$. Whenever this presents a hindrance in an actual application, such a protocol should not be used.

All the queries to $\mathbf{K}$ will take the form (u, x, D) where the *u* indicates that the query is initiated by agent u, $x \in Q$ is the actual query, and $D \subseteq V$ indicates the subset of agents with whom the answer to the query should be shared (parentheses will be omitted for singletons). As hinted above, one can devise several ways in which, given a communication graph *G*, the actual communication protocol can be carried out. Some of these ways are listed and discussed below. The list is not intended to be exhaustive, but rather an initial indication of protocols that may prove useful in some particular applications. In fact, we consider the variety of communication regimes between the querying agents to be an important, application-dependent, research question, open to future exploration.

In the remainder of the section, we assume that for a query $q = (u, x, D)$, each $v \in D \cup \{u\}$ will receive the answer (Y, N or U) as well as the initiator u and the query x. In particular, two distinct agents in D are not made aware of each others' membership in D. We consider two very simple models of communication between the querying agents.

1. *Edge-queries*: Here the set of queries is $Q_e = \{(u, x, v) \mid (u, v) \in E \wedge x \in Q\}$; a query $q = (u, x, v)$ is initiated by u and the answer $R_e(q)$ is submitted to both u and v (and nobody else).

2. *Partial-neighborhood-queries*: Here the query set represents a generalization of the previous two cases $Q_n = \{(u, x, D) \mid u \in V \wedge x \in Q \wedge D \subseteq Adj(u)\}$; the query $q = (u, x, D)$ is initiated by u and the answer $R_n(q)$ is shared with the subset D of neighbors of vertices adjacent to *u*. A *full-neighborhood-query* is one in which $D = Adj(u)$.

Even though the edge-queries represent the simplest kind of communication, through its analysis we will be able to get the basic idea of our approach to the core problem of sharing the answers to queries while protecting the required secret information.

5.1 Edge queries

We shall first consider the edge-queries protocol. Consider a single edge $(u, v) \in E$ and a corresponding query $q = (u, x, v)$. What should $R_e(q)$ be, given that its goal is to disclose neither

1. $x \in S_u$ to u, nor
2. $x \in S_v$ to v?

Define the function $R_e : Q_e \to A$ by setting $R_e(q)$ to be U, if $R_u(x) = U$ or $R_v(x) = U$, and Y, otherwise. I.e., $R_e(q) = Y$ iff $x \in K^+\backslash(E_u \cup E_v) = Q_Y^u \cap Q_Y^v$. As usual, we define $R_e((u, x, v)) = N$ if, and only if, $R_e((u, \neg x, v)) = Y$, in case the underlying language has negation. It is easy to see that R_e satisfies conditions 1 and 2.

Example 3: We want to illustrate a situation in which a querying agent may learn about the membership of a query in another agent's envelope. Suppose $(u, v), (v, w) \in E$, u poses the query $q = (u, x, v)$ and v poses the query $q' = (v, x, w)$. Here are the four possibilities of answers:

1. $R_e(q) = R_e(q') = Y$; in this case u learns that $x \in Q_Y^u \cap Q_Y^v = K^+\backslash(E_u \cup E_v)$, w learns that $x \in Q_Y^v \cap Q_Y^w = K^+\backslash(E_v \cup E_w)$ and v learns that $x \in Q_Y^u \cap Q_Y^v \cap Q_Y^w = K^+\backslash(E_u \cup E_v \cup E_w)$.
2. $R_e(q) = Y$ & $R_e(q') = U$; in this case u learns that $x \in Q_Y^u \cap Q_Y^v = K^+\backslash(E_u \cup E_v)$, w learns that $x \in E_v \cup E_w$ and v learns that $x \in E_w\backslash(E_u \cup E_v)$.
3. $R_e(q) = U$ & $R_e(q') = Y$; in this case u learns that $x \in E_u \cup E_v$, w learns that $x \in Q_Y^v \cap Q_Y^w = K^+\backslash(E_v \cup E_w)$ and v learns that $x \in E_u\backslash(E_v \cup E_w)$.
4. $R_e(q) = R_e(q') = U$; in this case u learns that $x \in E_u \cup E_v$, w learns that $x \in E_v \cup E_w$ and v learns that $x \in E_v \cup (E_u \cap E_w)$.

5.2 Neighborhood queries

Consider now the query $q = (u, x, D)$, initiated by u, whose answer is to be shared with those of its neighbors that happen to belong to the subset $D \subseteq Adj(u)$. In defining $R_n(q)$, the overall goal of being as informative as possible should be balanced against preserving secrecy. Thus R_n must not disclose either

1. $x \in S_u$ to u, or
2. $x \in S_v$ to v, for any $v \in D$.

Define the function $R_n : Q_n \to A$ by setting $R_n(q)$ to be U, if $R_u(x) = U$ or $R_v(x) = U$, for some $v \in D$, and Y, otherwise. In other words, $R_n(q) = Y$ iff $x \in K^+\backslash \left(E_u \cup (\bigcup_{v\in D} E_v)\right) = Q_Y^u \cap \left(\bigcap_{v\in D} Q_Y^v\right)$. It follows that R_n will answer *U* to every query $q = (u, x, D)$ with $x \in E_u \cup (\bigcup_{v\in D} E_v)$. Again, we define $R_n(u, x, D) = N$ if, and only if, $R_n((u, \neg x, D)) = Y$, in case the logical language has negation. Again, it can be easily verified that 1 and 2 are satisfied.

Summary

The widespread adoption of, and reliance on networked information systems call for methods for balancing the need to share information against the need to protect sensitive or secret information. Most of the existing methods for the protection of secret information rely on forbidding access to the sensitive parts of a knowledge base. However, many applications call for a more flexible approach that allows the knowledge base to use secret information to answer queries whenever it is possible to do so without risking the disclosure of secret information. In this paper, we have: formalized this problem of secrecy-preserving reasoning; introduced the notion of a secrecy envelope, i.e., a superset of secret information that should be protected in order to ensure that the secret information is protected, and analyzed some of its key properties; defined the notions of tight and optimal secrecy envelopes that yield maximally informative secrecy-preserving reasoners; presented an algorithm for computing a tight secrecy envelope (depending on the order of the incoming queries). We have also introduced a simple model to facilitate the analysis of secrecy-preserving reasoning in the case of a knowledge base that answers queries from multiple querying agents with different secrecy sets, with the possibility of sharing the answers supplied to them with each other (specified by some "answer sharing" protocols). Work in progress is aimed at the design and implementation of secrecy-preserving reasoners for a broad class of knowledge bases of interest in practical applications, including, in particular, hierarchical, propositional, RDF, and computationally tractable subclasses of description logic knowledge bases.

References

[Bao *et al.*, 2007] Jie Bao, Giora Slutzki, and Vasant Honavar. Privacy-preserving reasoning on the semantic web. In *Web Intelligence*, pages 791–797, 2007.

[Bentz, 2008] Cédric Bentz. On the complexity of the multicut problem in bounded tree-width graphs and digraphs. *Discrete Applied Mathematics*, 156(10):1908–1917, 2008.

[Bertino *et al.*, 2006] Elisa Bertino, L. R. Khan, Ravi S. Sandhu, and Bhavani M. Thuraisingham. Secure knowledge management: confidentiality, trust, and privacy. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 36(3):429–438, 2006.

[Bonatti *et al.*, 2006] Piero A. Bonatti, Claudiu Duma, Norbert Fuchs, Wolfgang Nejdl, Daniel Olmedilla, Joachim Peer, and Nahid Shahmehri. Semantic web policies - a discussion of requirements and research issues. In *ESWC*, pages 712–724, 2006.

[Calinescu *et al.*, 2003] Gruia Calinescu, Cristina G. Fernandes, and Bruce A. Reed. Multicuts in unweighted graphs and digraphs with bounded degree and bounded treewidth. *J. Algorithms*, 48(2):333–359, 2003.

[Chawla *et al.*, 2005] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. In *IEEE Conference on Computational Complexity*, pages 144–153. IEEE Computer Society, 2005.

[Cheriyan *et al.*, 2001] Joseph Cheriyan, Howard J. Karloff, and Yuval Rabani. Approximating directed multicuts. In *FOCS*, pages 320–328, 2001.

[Chuzhoy and Khanna, 2006] Julia Chuzhoy and Sanjeev Khanna. Hardness of cut problems in directed graphs. In Jon M. Kleinberg, editor, *STOC*, pages 527–536. ACM, 2006.

[Clifton *et al.*, 2002] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, , and M. Zhu. Tools for Privacy Preserving Distributed Data Mining. *ACM SIGKDD Explorations*, 4(2), December 2002.

[Cuenca Grau and Horrocks, 2008] Bernardo Cuenca Grau and Ian Horrocks. Privacy-preserving query answering in logic-based information systems. In *Proc. of the 18th Eur. Conf. on Artificial Intelligence (ECAI 2008)*, 2008.

[Farkas *et al.*, 2006] Csilla Farkas, Alexander Brodsky, and Sushil Jajodia. Unauthorized inferences in semistructured databases. *Information Sciences*, 176:3269–3299, 2006.

[Garg *et al.*, 1997] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997.

[Giereth, 2005] Mark Giereth. On partial encryption of rdf-graphs. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2005.

[Jain and Farkas, 2006] Amit Jain and Csilla Farkas. Secure resource description framework: an access control model. In *SACMAT*, pages 121–129, 2006.

[Kagal *et al.*, 2006] Lalana Kagal, Tim Berners-Lee, Dan Connolly, and Daniel J. Weitzner. Using semantic web technologies for policy management on the web. In *AAAI*, 2006.

[Kolovski *et al.*, 2007] Vladimir Kolovski, James A. Hendler, and Bijan Parsia. Analyzing web access control policies. In *WWW*, pages 677–686, 2007.

[Leighton and Rao, 1999] Frank Thomson Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999.

[O'Keefe *et al.*, 2004] Christine M. O'Keefe, Ming Yung, Lifang Gu, and Rohan Baxter. Privacy-preserving data linkage protocols. In *WPES '04: Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, pages 94–102, New York, NY, USA, 2004. ACM.

[Voutsadakis *et al.*, 2008] George Voutsadakis, Giora Slutzki, and Vasant Honavar. Secrecy preserving reasoning over entailment systems: Theory and applications. *Preprint*, 2008.