Comparing Availability in Controlled Query Evaluation Using Unordered Query Evaluation for Known Potential Secrets

George Voutsadakis[1,2]

[1] Department of Computer Science, Iowa State University, Ames, IA, USA
[2] Department of Mathematics and Computer Science, Lake Superior State University, Sault Ste. Marie, MI, USA

Abstract. Controlled Query Evaluation (CQE) is an inference control mechanism used to dynamically preserve confidentiality in secure information systems. In this note we introduce the notion of unordered query evaluation as a vehicle for comparing availability of various CQE methods, a recurring theme in the CQE literature. Moreover, we show that the various procedural approaches introduced thus far in the literature for imposing the declarative requirements of CQE lead to maximally available mechanisms. Finally, we characterize maximally available unordered query evaluation for various enforcement methods for known potential secrets as the unordered query evaluation resulting from CQE mechanisms for some suitably chosen ordering of the queries.

1 Introduction

Preserving confidentiality in information systems that contain both classified and public data is one of the major goals in data security. Two general categories of methods have been considered in the literature: *access control* and *information flow control*, that are characterized by their *static* and *dynamic* nature, respectively. In using access control, one of the major challenges is the *inference problem*, i.e., the potential inference of secret information by the user, based on revealed public information. Some pointers to the literature on the inference problem in access control are [11, 13, 19] (see [12] for a review). This problem is addressed in *Controlled Query Evaluation* (CQE), an inference control mechanism used to dynamically preserve confidentiality [14]. In CQE, the knowledge base administrator specifies the information that is to be kept confidential. When a query is posed against the knowledge base, its true answer is computed and, before an answer is issued to the user, a *censor* is used, aided by a *log* maintained to represent the user's assumed information, to detect potential security risks. If the correct answer jeopardizes confidentiality of sensitive information, then, instead of the correct answer, either a *lie* or a *refusal* is returned.

CQE methods come in various types and forms depending on the value of three parameters used to specify the semantics of the *confidentiality policies*, the *user awareness* and the *enforcement policies* [4, 5]. Confidentiality policies include *secrecies* and *potential secrets*. Roughly speaking, when a method is devised to protect secrecies, then the truth value of a secret may not be revealed or inferred. On the other hand, when a method protects potential secrets, one of the two possible truth values is designated as secret, and the method is charged with not revealing that value or allowing the user to infer that value. Its negation, however, is not protected. User awareness is the parameter that specifies whether or not the user is aware of the information whose truth value the knowledge base is trying to conceal. Finally, the enforcement policies that might be employed to ensure preservation of confidentiality are *lying*, *refusal* or a *combination* of lying and refusal.

CQE was first introduced by Sicherman et al. [14] for the method of refusal for protecting known and unknown secrecies. Bonatti et al. [9] discuss the case of lying for known potential secrets. Lying and refusal for known and unknown secrecies is taken up in [2]. Lying and refusal for known potential secrets is studied in [3]. Finally, Biskup and Bonatti explore a combination of refusal and lying for known secrecies and known potential secrets in [6]. Related later works include the creation of a SAT-based algorithm to pre-process information to create an inference-proof database that can be queried statically [10], as well as using constraint satisfaction for the construction of a secure database that also allows static querying without disclosing sensitive information [7]. It is worth noting that, recently, Biskup and Weibert have extended aspects of CQE to the setting of incomplete databases [8].

In controlled query evaluation, a central theme is the tradeoff between confidentiality of secret information and availability of information (see, e.g., the introduction in [10]). In fact, a recurring issue in all aforementioned works on CQE is a comparison of the various enforcement methods with respect to availability. Many take the forms of "honeymoon lemmas" comparing answers provided to initial segments of incoming query sequences [6]. Others are based on rearranging the incoming query sequence to achieve comparability (see, also, [6]). It seems that order, which is indispensable in an algorithmic treatment of security in information management, is presenting a hinderance when trying to compare at a theoretical level the various enforcement methods with respect to availability. In this paper, we make an attempt at a cleaner approach to this issue, based on an unordered view of confidentiality preserving query evaluation.

Our contribution is two-fold. First, we provide the definition of *unordered query evaluation*. Controlled query evaluation is usually presented in two levels of abstraction. In the higher level, a *declarative framework* is introduced, where a description of the method and its confidentiality requirements are given. In the lower level of abstraction, a detailed *algorithmic procedure* is provided for enforcing the confidentiality goals specified at the declarative level. We view unordered query evaluation as a third (the highest) level of abstraction, which "forgets" the order in which the queries are handled and considers only the association with each query of the corresponding response returned to the user. In this setting the two lower levels are viewed as describing a specific possible "order-dependent" implementation strategy for unordered query evaluation. We use the unordered query evaluation framework to provide a setting in which to estimate the *relative availability* of confidentiality preserving query answering. By associating an *unordered query evaluation companion* to each given CQE method, we carry the applicability of this comparison framework to CQE. We use a comparison result of Biskup and Bonatti [6] to illustrate how our definition may be used to accommodate results of similar kind presented previously in the literature.

The second contribution of the paper is the theoretical *characterization of the maximally available unordered query evaluation methods*. This result also relies on the construction of an unordered query evaluation related to a given CQE. We show that an unordered query evaluation is maximally available iff, for every knowledge base and confidential information, there exists an appropriate ordering of all possible queries, such that the given unordered query evaluation provides identical answers with a CQE processing the queries in the devised order.

The paper is organized as follows. In Section 2 we review the basic definitions and the formalism pertaining to the confidentiality requirements of CQE. In Section 3, we introduce unordered query evaluation and present its confidentiality requirements. Unordered query evaluation is more abstract than CQE, lacking the algorithmic flavor of CQE that results from dealing with the queries in the order in which they are posed to the system. In Section 4, we introduce a relative measure of availability in unordered query evaluation. Moreover, we formalize a way in which an unordered query evaluation may be canonically associated with a CQE and a given ordering of all queries. By using this unordered companion to the CQE and the availability comparison framework for the unordered case, we obtain a relative measure of availability for CQE. In Section 5 we present an availability comparison result of Biskup and Bonatti for various enforcement policies in the case of known potential secrets, placing it in the framework of the present paper as an illustration of the applicability of our general definitions. Finally, in Section 6 we provide a characterization of maximally available unordered query evaluation in terms of the known algorithms for the various enforcement policies in CQE. We concentrate on known potential secrets, but our method is general enough to encompass unknown potential secrets as well as known and unknown secrecies.

2 Controlled Query Evaluation

2.1 Basic Definitions

An *information system* [4, 5] consists of two pieces of data: First, a *schema* DS, which captures the universe of discourse of an intended application and which, for the purposes of this note, will be a finite set of propositional variables. Second, an instance db, which, in general, is a structure interpreting the symbols in DS and, which, for the purposes of this note, will be an assignment of truth values (true ($\mathbf{t}$) or false ($\mathbf{f}$)) to the propositional variables in DS. A *query Φ against* DS is a sentence in classical propositional logic with variables in DS. A *query* *evaluation* $\mathrm{eval}(\Phi)$ determines the truth value of a query Φ against the schema DS for the current instance db as follows:

$$\mathrm{eval}(\Phi) : \mathrm{DS} \rightarrow \{\mathbf{t}, \mathbf{f}\} \text{ with } \mathrm{eval}(\Phi)(\mathrm{db}) = \mathrm{db}\ \texttt{model\_of}\ \Phi,$$

where $\texttt{model\_of}$ is the boolean operator returning $\mathbf{t}$ iff db is a model of Φ in the ordinary sense. As is customary, we also use another version eval^*, that returns either the query sentence or its negation:

$$\mathrm{eval}^*(\Phi) : \mathrm{DS} \rightarrow \{\Phi, \neg\Phi\}, \quad \text{with}$$

$$\mathrm{eval}^*(\Phi)(\mathrm{db}) = \begin{cases} \Phi, & \text{if db } \texttt{model\_of}\ \Phi \\ \neg\Phi, & \text{otherwise} \end{cases}.$$

Let $Q = \langle \Phi_1, \Phi_2, \ldots, \Phi_i, \ldots \rangle$ be a (possibly infinite) query sequence and $\log_0$ be an *initial user log*, which represents the explicit part of the user's *assumed knowledge*. We define a *controlled query evaluation* as a family of partial functions $\mathrm{control\_eval}(Q, \log_0)$, each of which has as parameters the query sequence Q and the initial user log $\log_0$. The inputs are "admissible" pairs $(\mathrm{db}, \mathrm{policy})$, where db is an instance of the information system and policy an instance of a confidentiality policy, which can be a set of *secrecies* or a set of *potential secrects*, as in [4, 5]. Admissibility of $(\mathrm{db}, \mathrm{policy})$ is determined by some formally defined precondition precond associated with the function.

For any specific CQE function, the choices with respect to model of policy (secrecies or potential secrects), user awareness (unknown or known policy) and enforcement method (lying, refusal or combined) are indicated by attaching the superscripts p, a, e, with $p \in \{\texttt{sec}, \texttt{ps}\}, a \in \{\texttt{unknown}, \texttt{known}\}$ and $e \in \{\texttt{L(ying)}, \texttt{R(efusal)}, \texttt{C(ombined)}\}$. In specifying such a function, it is assumed that, given a query, the correct answer to the query according to the current database instance is judged by some *censor*, which decides whether the correct answer may be disclosed or whether a *modificator* must be applied. The censor is assisted by a *user log* log, which represents the explicit part of the user's *assumed knowledge*, much like $\log_0$ represents the explicit part of the user's initial assumed knowledge. The log is updated every time an answer is returned. Therefore, the function is given by

$$\mathrm{control\_eval}^{p,a,e}(Q, \log_0)(\mathrm{db}, \mathrm{policy}) = \langle (\mathrm{ans}_1, \log_1), \ldots, (\mathrm{ans}_i, \log_i), \ldots \rangle,$$

where $\log_i = \begin{cases} \log_{i-1}, & \text{if } \mathrm{ans}_i = \texttt{mum} \\ \log_{i-1} \cup \{\mathrm{ans}_i\}, & \text{otherwise} \end{cases}$, $\texttt{mum}$ signifying refusal to answer.

2.2 Confidentiality Requirements

Depending on whether the model of confidentiality policy is that of secrecies or of potential secrets, we have appropriately adjusted instances of the policy. For the model of secrecies, we have a finite set $\mathrm{secr} = \{\{\Psi_1, \neg\Psi_1\}, \ldots, \{\Psi_k, \neg\Psi_k\}\}$ of complementary pairs of sentences, each called a *secrecy*. On the other hand, for potential secrets, we have a finite set $\mathrm{pot\_sec} = \{\Psi_1, \ldots, \Psi_k\}$ of sentences, called *potential secrets*. The semantics for a secrecy $\{\Psi, \neg\Psi\}$ requires that a user should not be able to distinguish, based on initial knowledge and answers returned by the system, whether Ψ or $\neg\Psi$ is true in the actual instance of the information system. For a potential secret Ψ, a user should not be able to exclude that $\neg\Psi$ is true in the actual instance of the information system. More formally, we have the following definition for preservation of confidentiality for a given controlled query evaluation function (see Definition 1 of [4]):

Definition 1. *Let* $\mathrm{control\_eval}^{p,a,e}(Q, \log_0)$ *be a specific controlled query evaluation with* precond *as its associated precondition and* policy_1 *a policy instance.*

1. $\mathrm{control\_eval}^{p,a,e}(Q, \log_0)$ *is said to* preserve confidentiality with respect to policy_1 *iff, for all finite prefixes* Q' *of* Q*, all instances* db_1 *of the information system, such that* $(\mathrm{db}_1, \mathrm{policy}_1)$ *satisfies* precond *and all* $\Theta \in \mathrm{policy}_1$*, there exists* db_2 *and* policy_2*, such that* $(\mathrm{db}_2, \mathrm{policy}_2)$ *satisfies* precond *and the following conditions hold:*
   (a) [Same Answers]
   $\mathrm{control\_eval}^{p,a,e}(Q', \log_0)(\mathrm{db}_1, \mathrm{policy}_1) = \mathrm{control\_eval}^{p,a,e}(Q', \log_0)(\mathrm{db}_2, \mathrm{policy}_2)$
   (b) [Different Secrects/False Potential Secrets] *If* $p = \mathtt{sec}$*, i.e.,* $\Theta = \{\Psi, \neg\Psi\}$ *a secrecy,* $\{\mathrm{eval}^*(\Psi)(\mathrm{db}_1), \mathrm{eval}^*(\Psi)(\mathrm{db}_2)\} = \{\Psi, \neg\Psi\}$ *and, if* $p = \mathtt{ps}$*, i.e.,* $\Theta = \Psi$ *is a potential secret, then* $\mathrm{eval}^*(\Psi)(\mathrm{db}_2) = \neg\Psi$
   (c) [Awareness] *if* $a = \mathtt{known}$*, then* $\mathrm{policy}_1 = \mathrm{policy}_2$*.*
2. $\mathrm{control\_eval}^{p,a,e}(Q, \log_0)$ *is said to* preserve confidentiality *if it preserves confidentiality with respect to all admissible policy instances.*

3 Unordered Query Evaluation

In this section, we introduce *unordered query evaluation*, which is a modification of controlled query evaluation in which the ordering of the queries does not play a role. It is closer in spirit to alternative treatments of secrecy preserving reasoning that have been introduced in the literature, namely those by Studder [16] (see also [15, 17]) and Bao et al. [1]. The reason for defining unordered query evaluation is that it provides a more elegant way to compare methods of dynamic enforcement of controlled query evaluation with respect to their availability, which is the main topic of this paper. The way this can be achieved will be illustrated in Sections 5 and 6. In this section we give the definition and the relevant confidentiality requirements.

We adopt the same notion of *information system* that was used in Section 2 and the same notion of query and query evaluation. An *unordered query evaluation* is a family of partial functions $\mathrm{unord\_eval}(\log_0)$, each of which has as a parameter the initial user log $\log_0$. The inputs are "admissible" pairs $(\mathrm{db}, \mathrm{policy})$, where db is an instance of the information system and policy an instance of a confidentiality policy, which, as before, can be a set of *secrecies* or a set of *potential secrets*. A precondition precond determines the admissibility of (db, policy). The superscripts p, a, e, with $p \in \{\mathtt{sec}, \mathtt{ps}\}$, $a \in \{\mathtt{unknown}, \mathtt{known}\}$ and $e \in \{\mathtt{L(ying)}, \mathtt{R(efusal)}, \mathtt{C(ombined)}\}$, are used, as before, to indicate model of policy, user awareness and method of enforcement, respectively.

$\mathrm{unord\_eval}^{p,a,e}(\log_0)(\mathrm{db}, \mathrm{policy})$ is a total function from the set of queries $\mathcal{Q}$ to the set of possible answers, i.e., for all $\Phi \in \mathcal{Q}$,

$$\mathrm{unord\_eval}^{p,a,e}(\log_0)(\mathrm{db}, \mathrm{policy})(\Phi) \in \{\Phi, \neg\Phi, \mathtt{mum}\}.$$

The following definition for preservation of confidentiality for a given unordered query evaluation function adapts the corresponding one for controlled query evaluation (Definition 1):

Definition 2. *Let* $\mathrm{unord\_eval}^{p,a,e}(\log_0)$ *be a specific unordered query evaluation with* precond *as its associated precondition and* policy_1 *a policy instance.*

1. $\mathrm{unord\_eval}^{p,a,e}(\log_0)$ *is said to* preserve confidentiality with respect to policy_1 *iff, for every finite* $\mathcal{Q}' \subseteq \mathcal{Q}$*, all instances* db_1 *of the information system, such that* $(\mathrm{db}_1, \mathrm{policy}_1)$ *satisfies* precond *and all* $\Theta \in \mathrm{policy}_1$*, there exists* db_2 *and* policy_2*, such that* $(\mathrm{db}_2, \mathrm{policy}_2)$ *satisfies* precond *and the following conditions hold:*
   (a) [Same Answers] *for all* $\Phi \in \log_0 \cup \mathcal{Q}'$, $\mathrm{unord\_eval}^{p,a,e}(\log_0)(\mathrm{db}_1, \mathrm{policy}_1)(\Phi) = \mathrm{unord\_eval}^{p,a,e}(\log_0)(\mathrm{db}_2, \mathrm{policy}_2)(\Phi)$
   (b) [Different Secrects/False Potential Secrets] *If* $p = \mathtt{sec}$*, i.e.,* $\Theta = \{\Psi, \neg\Psi\}$ *a secrecy,* $\{\mathrm{eval}^*(\Psi)(\mathrm{db}_1), \mathrm{eval}^*(\Psi)(\mathrm{db}_2)\} = \{\Psi, \neg\Psi\}$ *and, if* $p = \mathtt{ps}$*, i.e.,* $\Theta = \Psi$ *is a potential secret, then* $\mathrm{eval}^*(\Psi)(\mathrm{db}_2) = \neg\Psi$
   (c) [Awareness] *if* $a = \mathtt{known}$*, then* $\mathrm{policy}_1 = \mathrm{policy}_2$.
2. $\mathrm{unord\_eval}^{p,a,e}(\log_0)$ *is said to* preserve confidentiality *if it preserves confidentiality with respect to all admissible policy instances.*

4 Availability

4.1 Availability in Unordered Query Evaluation

In many instances in previous work on controlled query evaluation, there has been explicit reference to the important tradeoff between *confidentiality* of secret information and *availability* of information (see, e.g., [10]). In general, in the framework of CQE, one way that has been employed for comparisons of difference enforcement methods with respect to availability has been via the so-called "Honeymoon Lemmas" and, also, via some query reordering-style lemmas [6]. In this subsection, we provide a general definition for comparing the availability of two *unordered* query evaluations. In the following subsection, we will show how a controlled query evaluation gives rise to an *unordered query evaluation companion* (essentially by forgetting the order of the queries) and we will use this associated unordered query evaluation together with the comparison framework of this subsection to provide a setting for comparing controlled query evaluations with respect to availability. Finally, in Section 5 we show how a comparison lemma of Biskup and Bonatti can be seen as a particular comparison result on availability in the sense of the present section.

Given a database instance db and a query Φ, we define a partial ordering $\leq_{\mathrm{db}}$ on the set $\{\Phi, \neg\Phi, \mathtt{mum}\}$ of the three possible answers of a controlled query evaluation on Φ by setting

$$\mathtt{mum} \leq_{\mathrm{db}} \mathrm{eval}^*(\Phi)(\mathrm{db}), \quad \neg\mathrm{eval}^*(\Phi)(\mathrm{db}) \leq_{\mathrm{db}} \mathrm{eval}^*(\Phi)(\mathrm{db}).$$

Let $\mathrm{unord\_eval}_1^{p,a,e_1}(\log_0)$ and $\mathrm{unord\_eval}_2^{p,a,e_2}(\log_0)$ be unordered query evaluations, with associated preconditions $\mathrm{precond}_1$ and $\mathrm{precond}_2$, respectively, and $(\mathrm{db}, \mathrm{policy})$ an admissible pair according to both $\mathrm{precond}_1$ and $\mathrm{precond}_2$. Then

$$\mathrm{unord\_eval}_1^{p,a,e_1}(\log_0)(\mathrm{db}, \mathrm{policy}) \leq \mathrm{unord\_eval}_2^{p,a,e_2}(\log_0)(\mathrm{db}, \mathrm{policy})$$

signifies that, for all $\Phi \in \mathcal{Q}$ (the set of all queries),

$$\mathrm{unord\_eval}_1^{p,a,e_1}(\log_0)(\mathrm{db}, \mathrm{policy})(\Phi) \leq_{\mathrm{db}} \mathrm{unord\_eval}_2^{p,a,e_2}(\log_0)(\mathrm{db}, \mathrm{policy})(\Phi).$$

Definition 3. *Let* $\mathrm{unord\_eval}_1^{p,a,e_1}$ *and* $\mathrm{unord\_eval}_2^{p,a,e_2}$ *be unordered query evaluations with preconditions* $\mathrm{precond}_1$ *and* $\mathrm{precond}_2$*, respectively.*

1. $\mathrm{unord\_eval}_1^{p,a,e_1}$ *is said to be* more available than $\mathrm{unord\_eval}_2^{p,a,e_2}$ with respect to *assumed knowledge* $\log_0$ and *confidentiality policy* policy *if, for every database instance* db*, such that* $(\mathrm{db}, \mathrm{policy})$ *is admissible according to both* $\mathrm{precond}_1$ *and* $\mathrm{precond}_2$*, we have that*

$$\mathrm{unord\_eval}_1^{p,a,e_1}(\log_0)(\mathrm{db}, \mathrm{policy}) \geq \mathrm{unord\_eval}_2^{p,a,e_2}(\log_0)(\mathrm{db}, \mathrm{policy});$$

2. $\mathrm{unord\_eval}_1^{p,a,e_1}$ *is said to be* maximally available with respect to *assumed knowledge* $\log_0$ and *confidentiality policy* policy *if, for every unordered query evaluation* $\mathrm{unord\_eval}_2^{p,a,e_1}$*, all database instances* db*, such that* $(\mathrm{db}, \mathrm{policy})$ *is admissible according to both* $\mathrm{precond}_1$ *and* $\mathrm{precond}_2$*, we have that*

$$\mathrm{unord\_eval}_1^{p,a,e_1}(\log_0)(\mathrm{db}, \mathrm{policy}) \not< \mathrm{unord\_eval}_2^{p,a,e_1}(\log_0)(\mathrm{db}, \mathrm{policy}).$$

4.2 From Controlled to Unordered Query Evaluation

In this subsection we show how to construct, given a controlled query evaluation, an associated unordered query evaluation, called its *unordered query evaluation companion*. We use the construction for the purpose of rigorously comparing controlled query evaluations with respect to availability.

Suppose that $\mathrm{control\_eval}^{p,a,e}$ is a controlled query evaluation, $\log_0$ a user's initial assumed knowledge and Q an infinite sequence, which is surjective on $\mathcal{Q}$, i.e., whose range includes all possible queries against the schema DS. Define the *unordered query evaluation companion* $\mathrm{unord\_eval}^{p,a,e}(\log_0)$ *of* $\mathrm{control\_eval}^{p,a,e}$ *relative to* $\log_0$ *and* Q as follows:

The precondition precond of the unordered query evaluation includes all pairs $(\mathrm{db}, \mathrm{policy})$ that are included in the precondition for $\mathrm{control\_eval}^{p,a,e}(Q, \log_0)$. Moreover, for all pairs $(\mathrm{db}, \mathrm{policy}) \in \mathrm{precond}$ and all $\Phi \in \mathcal{Q}$, we define

$$\mathrm{unord\_eval}^{p,a,e}(\log_0)(\mathrm{db}, \mathrm{policy})(\Phi) = \mathrm{ans}_i,$$

where

$$\text{control\_eval}^{p,a,e}(Q,\log_0)(\text{db},\text{policy}) = \langle(\text{ans}_1,\log_1),\ldots,(\text{ans}_i,\log_i),\ldots\rangle$$

and i is the first occurrence of Φ in the sequence $Q = \langle\Phi_1, \Phi_2, \ldots, \Phi_i, \ldots\rangle$.

Proposition 1. *If a controlled query evaluation* $\text{control\_eval}^{p,a,e}(Q,\log_0)$ *preserves confidentiality with respect to* policy*, then its unordered query evaluation companion* $\text{unord\_eval}^{p,a,e}(\log_0)$ *relative to* $\log_0$ *and* Q *also preserves confidentiality with respect to* policy*.*

Proof. (Sketch) Suppose $\text{control\_eval}^{p,a,e}(Q,\log_0)$, with precondition precond and policy_1 a policy instance, is a CQE, that preserves confidentiality with respect to policy_1. Let $\text{unord\_eval}^{p,a,e}(\log_0)$ be its unordered query evaluation companion relative to $\log_0$ and Q. Consider finite $\mathcal{Q}' \subseteq \mathcal{Q}$ and a database instance db_1, such that $(\text{db}_1, \text{policy}_1) \in \text{precond}$, and $\Theta \in \text{policy}_1$. Let Q' be a finite prefix of Q, whose range includes $\mathcal{Q}'$. Since precond is a common precondition for both $\text{control\_eval}^{p,a,e}(Q,\log_0)$ and $\text{unord\_eval}^{p,a,e}(\log_0)$, and $(\text{db}_1, \text{policy}_1) \in \text{precond}$, there exists, by preservation of confidentiality for the CQE, $(\text{db}_2, \text{policy}_2) \in \text{precond}$, such that all three conditions of Definition 1 are satisfied. Now it is straightforward to check that all three corresponding conditions of Definition 2 hold for $\text{unord\_eval}^{p,a,e}(\log_0)$ and, therefore, $\text{unord\_eval}^{p,a,e}(\log_0)$ also preserves confidentiality.

The definitions for comparing availability for unordered query evaluations may now be applied to the case of controlled query evaluations using their unordered query evaluation companions.

Definition 4. *Let* $\text{control\_eval}_1^{p,a,e_1}$ *and* $\text{control\_eval}_2^{p,a,e_2}$ *be controlled query evaluations with preconditions* precond_1 *and* precond_2*, respectively,* $\log_0$ *a user's initial assumed knowledge and* Q_1, Q_2 *infinite sequences, which are surjective on* $\mathcal{Q}$*.*

1. $\text{control\_eval}_1^{p,a,e_1}(Q_1,\log_0)$ *is said to be* more available than $\text{control\_eval}_2^{p,a,e_2}$ $(Q_2,\log_0)$ with respect to *confidentiality policy* policy *if* $\text{unord\_eval}_1^{p,a,e_1}$ *is more available than* $\text{unord\_eval}_2^{p,a,e_2}$ *with respect to assumed knowledge* $\log_0$ *and confidentiality policy* policy*, where*

$$\text{unord\_eval}_1^{p,a,e_1}(\log_0) \quad \textit{and} \quad \text{unord\_eval}_2^{p,a,e_2}(\log_0)$$

are the unordered query evaluation companions of $\text{control\_eval}_1^{p,a,e_1}(Q_1,\log_0)$ *and* $\text{control\_eval}_2^{p,a,e_2}(Q_2,\log_0)$*, respectively.*
2. $\text{control\_eval}_1^{p,a,e_1}(Q_1,\log_0)$ *is said to be* maximally available with respect to *confidentiality policy* policy *if* $\text{unord\_eval}_1^{p,a,e_1}$ *is maximally available with respect to assumed knowledge* $\log_0$ *and confidentiality policy* policy*.*

It is worth noting that in Definition 4 we chose to define maximal availability for controlled query evaluations to mean that the corresponding unordered query evaluation is the "best" among all other unordered query evaluations having the *same enforcement policy.*

5 A Comparison Result of Biskup and Bonatti

In this section, we revisit a result formulated and proven by Biskup and Bonatti in [6] on comparing the uniform with the hybrid methods in the case of known potential secrets in order to illustrate how this and other results of similar flavor may be accommodated in the general comparison framework that was presented in the previous section.

5.1 Uniform Refusal and Lying for Known Potential Secrets

We start by briefly reviewing the definition of the *refusal censor* and the resulting controlled query evaluation procedure in the case of *known potential secrets*. The requirement is that the censor refuse the answer to a query if the correct answer or its negation together with the current log imply a potential secret:

$$\begin{aligned}\mathrm{censor}^{\mathtt{R}}_{\mathtt{ps}}(\Phi, \mathrm{log}, \mathrm{db}, \mathrm{pot\_sec}) = (\exists \Psi)[&\Psi \in \mathrm{pot\_sec} \text{ and } (\mathrm{log} \cup \{\mathrm{eval}^*(\Phi)(\mathrm{db})\} \models \Psi \\ &\text{or } \mathrm{log} \cup \{\neg \mathrm{eval}^*(\Phi)(\mathrm{db})\} \models \Psi)].\end{aligned}$$

Now, $\mathrm{control\_eval}^{\mathtt{R}}_{\mathtt{ps}}(Q, \mathrm{log}_0)(\mathrm{db}, \mathrm{pot\_sec})$ is defined by

$$\mathrm{ans}_i = \begin{cases} \mathtt{mum}, & \text{if } \mathrm{censor}^{\mathtt{R}}_{\mathtt{ps}}(\Phi_i, \mathrm{log}_{i-1}, \mathrm{db}, \mathrm{pot\_sec}) \\ \mathrm{eval}^*(\Phi_i)(\mathrm{db}), & \text{otherwise} \end{cases}$$

$$\mathrm{log}_i = \begin{cases} \mathrm{log}_{i-1}, & \text{if } \mathrm{ans}_i = \mathtt{mum} \\ \mathrm{log}_{i-1} \cup \{\mathrm{ans}_i\}, & \text{otherwise} \end{cases}$$

In the case of refusal the appropriate precondition for applying the algorithm is $(\mathrm{db}, \mathrm{policy}) \in \mathrm{precond}$ iff db `model_of` log_0.

We next review the definition of the *lying censor* and the resulting controlled query evaluation procedure in the case of *known potential secrets*. The requirement is that the censor refuse the answer to a query if the correct answer together with the current log imply the *disjunction of all potential secrets*:

$$\mathrm{censor}^{\mathtt{L}}_{\mathtt{ps}}(\Phi, \mathrm{log}, \mathrm{db}, \mathrm{pot\_sec}) = \mathrm{log} \cup \{\mathrm{eval}^*(\Phi)(\mathrm{db})\} \models \mathrm{pot\_sec\_disj}$$
where $\mathrm{pot\_sec\_disj} = \bigvee_{\Psi \in \mathrm{pot\_sec}} \Psi$.

Now, $\mathrm{control\_eval}^{\mathtt{L}}_{\mathtt{ps}}(Q, \mathrm{log}_0)(\mathrm{db}, \mathrm{pot\_sec})$ is defined by

$$\mathrm{ans}_i = \begin{cases} \neg \mathrm{eval}^*(\Phi_i)(\mathrm{db}), & \text{if } \mathrm{censor}^{\mathtt{L}}_{\mathtt{ps}}(\Phi_i, \mathrm{log}_{i-1}, \mathrm{db}, \mathrm{pot\_sec}) \\ \mathrm{eval}^*(\Phi_i)(\mathrm{db}), & \text{otherwise} \end{cases}$$

$$\mathrm{log}_i = \mathrm{log}_{i-1} \cup \{\mathrm{ans}_i\}.$$

In the case of refusal the appropriate precondition for applying the algorithm is that for all $\Psi \in \mathrm{pot\_sec}$, $\mathrm{log}_0 \not\models \Psi$.

5.2 Combined Method for Known Potential Secrets

Biskup and Bonatti [6], realizing that both uniform controlled query evaluation methods (Refusal and Lying) have disadvantages, introduced the *combined lying and refusal method* for *known potential secrets*. In this method, when a query Φ is posed, the database could refuse, lie or provide the correct answer. Refusal occurs when the current log and the correct answer imply a potential secret and, in addition, the current log and the false answer also imply a potential secret. Lying occurs when the current log and the correct answer imply a potential secret but the current log and the false answer do not. Finally, the correct answer is provided in case the current log and the correct answer do not imply any potential secrets.

Thus, the controlled query evaluation method by combining refusal and lying for known potential secrets, $\text{control\_eval}^{\texttt{C}}_{\texttt{ps}}(Q, \log_0)(\text{db}, \text{pot\_sec})$, with

$$\text{control\_eval}^{\texttt{C}}_{\texttt{ps}}(Q, \log_0)(\text{db}, \text{pot\_sec}) = \langle(\text{ans}_1, \log_1), \ldots, (\text{ans}_i, \log_i), \ldots\rangle,$$

is defined by

$$\text{ans}_i = \begin{cases} \texttt{mum}, & \text{if } (\exists \Psi_1, \Psi_2 \in \text{pot\_sec})(\log_{i-1} \cup \{\text{eval}^*(\Phi_i)(\text{db})\} \models \Psi_1 \\ & \qquad \text{and } \log_{i-1} \cup \{\neg\text{eval}^*(\Phi_i)(\text{db})\} \models \Psi_2) \\ \neg\text{eval}^*(\Phi_i)(\text{db}), & \text{if } (\exists \Psi_1 \in \text{pot\_sec})(\log_{i-1} \cup \{\text{eval}^*(\Phi_i)(\text{db})\} \models \Psi_1) \\ & \text{and } (\nexists \Psi_2 \in \text{pot\_sec})(\log_{i-1} \cup \{\neg\text{eval}^*(\Phi_i)(\text{db})\} \models \Psi_2) \\ \text{eval}^*(\Phi_i)(\text{db}), & \text{otherwise} \end{cases}$$

$$\log_i = \begin{cases} \log_{i-1}, & \text{if ans}_i = \texttt{mum} \\ \log_{i-1} \cup \{\text{ans}_i\}, & \text{otherwise} \end{cases} .$$

In the case of the combined method the appropriate precondition for applying the algorithm is, for all $\Psi \in \text{pot\_sec}$, $\log_0 \not\models \Psi$. In Theorem 1 of [6], it is shown that the combined method is secure according to the general Definition 1.

5.3 Comparison Result

In Theorem 2 of [6], Biskup and Bonatti prove that the combined method is "more cooperative" than any of the uniform methods for *known potential secrets*. In Section 5 of [6], they also point out that the same holds for the case of *known secrecies*. In this section, after revisiting their result, we see that the informal notion of "more cooperative" is accurately captured by our formal notion of "more available", as detailed in Definition 4.

Theorem 1 (Biskup and Bonatti). *Let* `M` *denote either refusal (*`R`*) or lying (*`L`*) and* $\log_0$*,* $(\text{db}, \text{pot\_sec})$ *appropriate parameters. Then, for all query sequences* Q^M *for uniform* `M`*, there exists a query sequence* Q^C *for the combined method, such that:*

1. $\text{control\_eval}^{\texttt{C}}_{\texttt{ps}}(Q^C, \log_0)(\text{db}, \text{pot\_sec})$ *delivers all the answers that are correct under* $\text{control\_eval}^{\texttt{M}}_{\texttt{ps}}(Q^M, \log_0)(\text{db}, \text{pot\_sec})$ *and possibly more correct answers.*

2. Q^C *is defined by a reordering that shifts correctly answered queries towards the beginning of the query sequence.*

Using the terminology introduced in the present paper, Theorem 1 may be rephrased as follows:

Theorem 2. *Let* $\mathtt{M}$ *denote either refusal (*$\mathtt{R}$*) or lying (*$\mathtt{L}$*) and* $\log_0$*,* $(\mathrm{db}, \mathrm{pot\_sec})$ *appropriate parameters. Then, for all query sequences* Q^M *for uniform* $\mathtt{M}$*, that are surjective on* $\mathcal{Q}$*, there exists a query sequence* Q^C *for the combined method, also surjective on* $\mathcal{Q}$*, such that* $\mathrm{control\_eval}^{\mathtt{C}}_{\mathtt{ps}}(Q^C, \log_0)$ *is more available with respect to* pot_sec *than* $\mathrm{control\_eval}^{\mathtt{M}}_{\mathtt{ps}}(Q^M, \log_0)$

Theorem 2 is a direct consequence of Theorem 1 and Definition 4.

6 Maximal Availability for Known Potential Secrets

Since, in Definition 4 of maximal availability for a controlled query evaluation, its corresponding unordered query evaluation is compared with other unordered query evaluations having the *same enforcement policy*, Theorem 2 does not have any impact on maximality, since it is a comparison between controlled query evaluations with different enforcement policies. In this section, we undertake the task of showing that all three enforcement policies for known potential secrets are maximally available. We also present a result connecting maximally available unordered query evaluation with CQE. This result is related to the reordering of query sequences that has been a recurring theme in the studies of controlled query evaluation for various enforcement policies by Biskup and Bonatti (see, e.g., Lemma 2 of [6]), as well as with the, so-called, *order-induced secrecy preserving reasoners* studied in [18].

Theorem 3. *Let* $\mathtt{M}$ *denote refusal (*$\mathtt{R}$*), lying (*$\mathtt{L}$*) or the combined (*$\mathtt{C}$*) enforcement method,* $\log_0$*,* $(\mathrm{db}, \mathrm{pot\_sec})$ *appropriate parameters and* $Q = \langle \Phi_1, \Phi_2, \ldots, \Phi_i, \ldots \rangle$ *a query sequence that is surjective on* $\mathcal{Q}$*. Then,* $\mathrm{control\_eval}^{\mathtt{M}}_{\mathtt{ps}}(Q, \log_0)$ *is maximally available with respect to* pot_sec*.*

Proof. Let $\mathrm{unord\_eval}^{\mathtt{M}}_{\mathtt{ps}}(\log_0)(\mathrm{db}, \mathrm{pot\_sec})$ be the unordered query evaluation companion of $\mathrm{control\_eval}^{\mathtt{M}}_{\mathtt{ps}}(Q, \log_0)$. Suppose, for the sake of obtaining a contradiction, that $\mathrm{control\_eval}^{\mathtt{M}}_{\mathtt{ps}}(Q, \log_0)$ is not maximally available with respect to pot_sec. Thus, there exists $\mathrm{unord\_eval}'^{\mathtt{M}}_{\mathtt{ps}}(\log_0)(\mathrm{db}, \mathrm{pot\_sec})$ and db', such that $(\mathrm{db}', \mathrm{pot\_sec})$ is admissible for both unordered query evaluations and such that $\mathrm{unord\_eval}^{M}_{\mathtt{ps}}(\log_0)(\mathrm{db}', \mathrm{pot\_sec}) < \mathrm{unord\_eval}'^{M}_{\mathtt{ps}}(\log_0)(\mathrm{db}', \mathrm{pot\_sec})$. This means that there exists $i \geq 1$, such that

$$\mathrm{unord\_eval}^{\mathtt{M}}_{\mathtt{ps}}(\log_0)(\mathrm{db}', \mathrm{pot\_sec})(\Phi_i) <_{\mathrm{db}'} \mathrm{unord\_eval}'^{\mathtt{M}}_{\mathtt{ps}}(\log_0)(\mathrm{db}', \mathrm{pot\_sec})(\Phi_i).$$

Consider the smallest such i. Then, we must have, for all $j < i$,

$$\mathrm{unord\_eval}^{\mathtt{M}}_{\mathtt{ps}}(\log_0)(\mathrm{db}', \mathrm{pot\_sec})(\Phi_j) = \mathrm{unord\_eval}'^{\mathtt{M}}_{\mathtt{ps}}(\log_0)(\mathrm{db}', \mathrm{pot\_sec})(\Phi_j)$$

and $\text{unord\_eval}'^{\mathtt{M}}_{\mathtt{ps}}(\log_0)(\text{db}', \text{pot\_sec})(\Phi_i) <_{\text{db}'} \text{eval}^*(\Phi_i)(\text{db}')$. But, then, using the definition of $\text{unord\_eval}^{\mathtt{M}}_{\mathtt{ps}}(\log_0)(\text{db}', \text{pot\_sec})$ together with the description of $\text{control\_eval}^{\mathtt{M}}_{\mathtt{ps}}(Q, \log_0)$, we conclude that $\text{unord\_eval}'^{\mathtt{M}}_{\mathtt{ps}}(\log_0)$ does not preserve confidentiality, which is a contradiction. Thus, $\text{control\_eval}^{\mathtt{M}}_{\mathtt{ps}}(Q, \log_0)$ is maximally available.

Theorem 4. *Let* $\mathtt{M}$ *denote refusal* ($\mathtt{R}$)*, lying* ($\mathtt{L}$) *or the combined* ($\mathtt{C}$) *enforcement method and* $\log_0$, $(\text{db}, \text{pot\_sec})$ *appropriate parameters for a maximally available unordered query evaluation* $\text{unord\_eval}^{\mathtt{M}}_{\mathtt{ps}}(\log_0)$*. Then, there exists a query sequence* $Q = \langle \Phi_1, \Phi_2, \ldots, \Phi_i, \ldots \rangle$*, that is surjective on* $\mathcal{Q}$*, such that the controlled query evaluation* $\text{control\_eval}^{\mathtt{M}}_{\mathtt{ps}}(Q, \log_0)$*, with precondition* precond*, satisfies*

1. $(\text{db}, \text{pot\_sec}) \in \text{precond}$ *and*
2. $\text{control\_eval}^{\mathtt{M}}_{\mathtt{ps}}(Q, \log_0)(\text{db}, \text{pot\_sec}) = \langle (\text{ans}_1, \log_1), \ldots, (\text{ans}_i, \log_i), \ldots \rangle$*, with* $\text{ans}_i = \text{unord\_eval}^{\mathtt{M}}_{\mathtt{ps}}(\log_0)(\text{db}, \text{pot\_sec})(\Phi_i)$*, for all* $i \geq 1$.

Proof. We present the proof for $\mathtt{M} = \mathtt{L}$, i.e., for the uniform lying enforcement policy. The proofs for the other two cases are similar.

Let $\text{unord\_eval}^{\mathtt{L}}_{\mathtt{ps}}(\log_0)$ be a maximally available unordered query evaluation, $(\text{db}, \text{pot\_sec})$ appropriate parameters, $\text{Acc} = \{\Gamma_1, \Gamma_2, \ldots\}$ be the set of all accurate answers, i.e., such that $\text{unord\_eval}^{\mathtt{L}}_{\mathtt{ps}}(\log_0)(\text{db}, \text{pot\_sec})(\Gamma_i) = \text{eval}^*(\Gamma_i)(\text{db})$, for all i, and $\text{Alt} = \{\Delta_1, \Delta_2, \ldots\}$ the set of all altered answers, i.e., such that $\text{unord\_eval}^{\mathtt{L}}_{\mathtt{ps}}(\log_0)(\text{db}, \text{pot\_sec})(\Delta_i) = \neg\text{eval}^*(\Delta_i)(\text{db})$, for all i. To shorten notation, we write $\Gamma_i^* = \text{eval}^*(\Gamma_i)(\text{db})$ and, similarly for Δ_i^*. We also set $\text{Acc}^* = \{\Gamma^* : \Gamma \in \text{Acc}\}$ and similarly for Alt^*. We will exhibit a sequence $Q = \langle \Phi_1, \Phi_2, \ldots \rangle$, surjective on the set of all possible queries $\mathcal{Q}$, such that the controlled query evaluation $\text{control\_eval}^{\mathtt{L}}_{\mathtt{ps}}(Q, \log_0)$ satisfies the conditions of the statement.

The difficulty in constructing such a sequence lies in the fact that Acc may be countably infinite. If not, i.e., if $\text{Acc} = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_n\}$ is finite, then the sequence $\Gamma_1, \ldots, \Gamma_n, \Delta_1, \Delta_2, \ldots$ accomplishes our goal. In the case of countably infinite Acc, consider the sequence

$$\Gamma_1, \Gamma_2, \Gamma_3, \ldots. \tag{1}$$

Since $\text{unord\_eval}^{\mathtt{L}}_{\mathtt{ps}}(\log_0)$ preserves confidentiality, we must have that $\log_0 \cup \text{Acc}^* \not\models \Psi$, for every potential secret Ψ. On the other hand, the maximal availability of $\text{unord\_eval}^{\mathtt{L}}_{\mathtt{ps}}(\log_0)(\text{db}, \text{pot\_sec})$ implies that, for every $i = 1, 2, \ldots$, there exists a potential secret Ψ, such that $\log_0 \cup \text{Acc}^* \cup \{\Delta_i^*\} \models \Psi$. To place the elements $\Delta_1, \Delta_2, \ldots$ in the List (1), we work by induction on the index i as follows:

Since Δ_1 is such that $\log_0 \cup \text{Acc}^* \cup \{\Delta_1^*\} \models \Psi_1$, for some potential secret Ψ_1, there exist finitely many $\Gamma_{i_1^1}, \Gamma_{i_2^1}, \ldots, \Gamma_{i_{m_1}^1} \in \text{Acc}$, $i_1^1 < i_2^1 < \cdots < i_{m_1}^1$, such that $\log_0 \cup \{\Gamma_{i_1^1}^*, \ldots, \Gamma_{i_{m_1}^1}^*\} \cup \{\Delta_1^*\} \models \Psi_1$. Let $l = \max\{i : \Gamma_i^* \in \log_0\}$ and $n_1 = \max\{l, i_{m_1}^1\}$. Insert Δ_1 immediately after Γ_{n_1} in the List (1).

Suppose, next that $\Delta_1, \Delta_2, \ldots, \Delta_{k-1}$ have all been placed in appropriate positions in the List (1). We work to place Δ_k. Since Δ_k is such that $\log_0 \cup \text{Acc}^* \cup \{\Delta_k^*\} \models \Psi_k$, for some potential secret Ψ_k, there exist $\Gamma_{i_1^k}, \Gamma_{i_2^k}, \ldots, \Gamma_{i_{m_k}^k} \in$

Acc, $i_1^k < i_2^k < \cdots < i_{m_k}^k$, such that $\log_0 \cup \{\Gamma^*_{i_1^k}, \ldots, \Gamma^*_{i_{m_k}^k}\} \cup \{\Delta_k^*\} \models \Psi_k$. Let $n_k = \max\{l, i_{m_1}^1, \ldots, i_{m_k}^k, n_{k-1}+1\}$. Insert Δ_k immediately after Γ_{n_k} in the List (1).

We will show by induction on the index i that, for all $i = 1, 2, \ldots$, if $\Phi_i = \Gamma_j$, then $\text{ans}_i = \Gamma_j^*$ and, if $\Phi_i = \Delta_j$, then $\text{ans}_i = \neg\Delta_j^*$, where $\text{control\_eval}^{\mathtt{L}}_{\mathtt{ps}}(Q, \log_0)(\text{db}, \text{pot\_sec}) = \langle(\text{ans}_1, \log_1), \ldots, (\text{ans}_i, \log_i), \ldots\rangle$.

For the basis of the induction, let $i = 1$ and consider two cases:

- If $\Phi_1 = \Gamma_1$, then, by preservation of confidentiality, for all $\Psi \in \text{pot\_sec}$, $\log_0 \cup \{\Gamma_1^*\} \not\models \Psi$. Thus, by the algorithm for lying, $\text{ans}_1 = \Gamma_1^* = \text{unord\_eval}^{\mathtt{L}}_{\mathtt{ps}}(\log_0)(\text{db}, \text{pot\_sec})(\Gamma_1)$.
- If $\Phi_1 = \Delta_1$, then, by the construction, exists $\Psi_1 \in \text{pot\_sec}$, $\Delta_1^* \models \Psi_1$. Thus, a fortiori, $\log_0 \cup \{\Delta_1^*\} \models \Psi_1$. Therefore, $\text{ans}_1 = \neg\Delta_1^* = \text{unord\_eval}^{\mathtt{L}}_{\mathtt{ps}}(\log_0)(\text{db}, \text{pot\_sec})(\Delta_1)$.

Assume, now, that for all $i = 1, \ldots, k-1$, we have that $\text{ans}_i = \text{unord\_eval}^{\mathtt{L}}_{\mathtt{ps}}(\log_0)(\text{db}, \text{pot\_sec})(\Phi_i)$. If $\Phi_k = \Gamma_j$, for some j, then by preservation of confidentiality and the induction hypothesis, for all $\Psi \in \text{pot\_sec}$, $\log_0 \cup \{\text{ans}_1, \ldots, \text{ans}_{k-1}\} \cup \{\text{unord\_eval}^{\mathtt{L}}_{\mathtt{ps}}(\log_0)(\text{db}, \text{pot\_sec})(\Gamma_j)\} \not\models \Psi$. Thus, for all $\Psi \in \text{pot\_sec}$, $\log_{k-1} \cup \{\Gamma_j^*\} \not\models \Psi$. Therefore, $\text{ans}_k = \Gamma_j^* = \text{unord\_eval}^{\mathtt{L}}_{\mathtt{ps}}(\log_0)(\text{db}, \text{pot\_sec})(\Gamma_j)$. The case where $\Phi_i = \Delta_j$, for some j, may be handled similarly.

7 Summary

In this paper, we defined *unordered query evaluation* as a way to study the *relative availability* of various *controlled query evaluation enforcement methods*. We first compare availability of unordered query evaluation in a straightforward way and, then, by associating an *unordered query evaluation companion* to each given CQE method, we carry the applicability of this comparison framework to CQE. We used a comparison result of Biskup and Bonatti [6] to illustrate how our definition may be used to accommodate availability comparison results considered in the literature.

We also theoretically connected *maximally available unordered query evaluation* methods with CQE. We showed that an unordered query evaluation is maximally available only if, for every input instance, there exists an appropriate ordering of all possible queries, such that the given unordered query evaluation provides identical answers with a CQE that is based on the ordering.

In the results of the previous section, concerning maximal availability, we focused on the policy method of *known potential secrets*. We believe, however, that the same techniques should allow us to prove corresponding results for all four combinations of policy methods and user awareness, i.e., unknown potential secrets and both known and unknown secrecies, and for all enforcement policies for which existing methods of controlled query evaluation are applicable.

References

1. Bao, J., Slutzki, G., and Honavar, V., Privacy-Preserving Reasoning on the Semantic Web, 2007 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2007, pp. 791-797

2. Biskup, J., For Unknown Secrecies Refusal is Better Than Lying, Data and Knowledge Engineering, Vol. 33 (2000), pp. 1-23
3. Biskup, J., and Bonatti, P.A., Lying Versus Refusal for Known Potential Secrets, Data and Knowledge Engineering, Vol. 38 (2001), pp. 199-222
4. Biskup, J., and Bonatti, P.A., Confidentiality Policies and their Enforcement for Controlled Query Evaluation, Proceedings of the 7th European Symposium on Research in Computer Security, ESORICS 2002, Lecture Notes in Computer Science, Vol. 2502, pp. 39-54
5. Biskup, J., and Bonatti, P., Controlled Query Evaluation for Enforcing Confidentiality in Complete Information Systems, International Journal of Information Security, Vol. 3 (2004), pp. 14-27
6. Biskup, J., and Bonatti, P.A., Controlled Query Evaluation for Known Policies by Combining Lying and Refusal, Annals of Mathematics and Artificial Intelligence, Vol. 40 (2004), pp. 37-62
7. Biskup, J., Burgard, D.M., Weibert, T., and Wiese, L., Inference Control in Logic Databases as a Constraint Satisfaction Problem, Proceedings of the 3rd International Conference on Information Systems Security, ICISS 2007, pp. 128-142
8. Biskup, J., and Weibert, T., Keeping Secrects in Incomplete Databases, International Journal of Information Security, Vol. 7 (2008), pp. 199-217
9. Bonatti, P.A., Kraus, S., and Subrahmanian, V.S., Foundations of Secure Deductive Databases, IEEE Transactions of Knowledge and Data Engineering, Vol. 7, No. 3 (1995), pp. 406-422
10. Biskup, J., and Wiese, L., On Finding an Inference-Proof Complete Database for Controlled Query Evaluation, Proceedings of the 20th Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Data and Applications Security XX (2006), pp. 30-43
11. Chang, L.W., and Moskowitz, I.S., A Study of Inference Problems in Distributed Databases, Proceedings of the 16th Annual IFIP WG 11.3 Conference on Data and Applications Security, Data and Applications Security XVI (2002), pp. 191-204
12. Farkas, C., and Jajodia, S., The Inference Problem: A Survey, ACM SIGKDD Explorations Newsletter, Vol. 4, No. 2 (2002), pp. 6-11
13. Hale, J., and Shenoi, S., Analyzing FD Inference in Relational Databases, Data and Knowledge Engineering, Vol. 18 (1996), pp. 167-183
14. Sicherman, G.L., de Jonge, W., and van de Riet, R.P., Answering Queries Without Revealing Secrets, ACM Transactions on Database Systems, Vol. 8, No. 1 (1983), pp. 41-59
15. Stoffel, K., and Studer, T., Provable Data Privacy, Database and Expert Systems Applications, DEXA 2005, pp. 324-332
16. Stouppa, P., and Studer, T., A Formal Model of Data Privacy, 6th International Andrei Ershov Memorial Conference, Perspectives of Systems Informatics, PSI 2006, pp. 400-408
17. Stouppa, P., and Studer, T., Data Privacy for ALC Knowledge Bases, Logical Foundations of Computer Science, LFCS 2009, pp. 409-421
18. Voutsadakis, G., Slutzki, G., and Honavar, V., Secrecy Preserving Reasoning Over Entailment Systems Theory and Applications, Technical Report, Department of Computer Science, Iowa State University
19. Yang, X., and Li, C., Secure XML Publishing Without Information Leakage in the Presence of Data Inference, Proceedings of the 30th Very Large Data Base Conference, VLDB 2004, pp. 96-107