15

Protection of Database Security Via Collaborative Inference Detection *

Yu Chen and Wesley W. Chu

Computer Science Department,
University of California, USA
{chenyu, wwc}@cs.ucla.edu

Abstract. Malicious users can exploit the correlation among data to infer sensitive information from a series of seemingly innocuous data accesses. Thus, we develop an inference violation detection system to protect sensitive data content. Based on data dependency, database schema and semantic knowledge, we constructed a semantic inference model (SIM) that represents the possible inference channels from any attribute to the pre-assigned sensitive attributes. The SIM is then instantiated to a semantic inference graph (SIG) for query-time inference violation detection. For a single user case, when a user poses a query, the detection system will examine his/her past query log and calculate the probability of inferring sensitive information. The query request will be denied if the inference probability exceeds the pre-specified threshold. For multi-user cases, the users may share their query answers to increase the inference probability. Therefore, we develop a model to evaluate collaborative inference based on the query sequences of collaborators and their task-sensitive collaboration levels. Experimental studies reveal that information authoritativeness and communication fidelity are two key factors that affect the level of achievable collaboration. An example is given to illustrate the use of the proposed technique to prevent multiple collaborative users from deriving sensitive information via inference.

15.1 Introduction

Access control mechanisms are commonly used to protect users from the divulgence of sensitive information in data sources. However, such techniques are insufficient because malicious users may access a series of innocuous information and then employ inference techniques to derive sensitive data using that information.

To address this inference problem, we develop an inference detection system that resides at the central directory site. Because inference channels can be used to provide a scalable and systematic sound inference, we need to construct a semantic inference model (SIM) that represents all the possible inference channels from any attribute in the system to the set of pre-assigned sensitive attributes. The SIM can be constructed by linking all the related attributes which can be derived via attribute dependency from data dependency, database schema and semantic related knowledge. Based on the semantic inference model, the violation detection system keeps track of a user's query history.

* This research is supported by NSF grant number IIS-03113283.

H. Chen and C.C. Yang (Eds.): Intelligence and Security Informatics, SCI 135, pp. 275–303, 2008.
springerlink.com

When a new query is posed, all the channels where sensitive information can be inferred will be identified. If the probability to infer sensitive information exceeds a pre-specified threshold, the current query request will then be denied. Therefore, our system can prevent malicious users from obtaining sensitive information.

This inference detection approach is based on the assumption that users are isolated and do not share information with one another. This assumption, however, may not be the case in a real-life situation. Most users usually work as a team, and each member can access the information independently. Afterwards, the members may merge their knowledge together and jointly infer the sensitive information. Generalizing from a single-user to a multi-user collaborative system greatly increases the complexity of the inference detection system.

For example, one of the sensitive attributes in the system can be inferred from four different inference channels. There are two collaborators and each poses queries on two separate channels. Based on individual inference violation detection, neither of the users violates the inference threshold from their query answers. However, if the two users share information, then the aggregated knowledge from the four inference channels can cause an inference violation (see Sect. 15.7.2).

This motivates us to extend our research from a single user to the multiple user case, where users may collaborate with each other to jointly infer sensitive data. We have conducted a set of experiments, using our inference violation detector as a test bed to understand the characteristics in collaboration as well as the effect on collaborative inference. From the experiments, we learn that for a given specific task, the amount of information that flows from one user to another depends on the closeness of their relationships and the knowledge related to the task. Thus, collaborative inference for a specific task can be derived by tracking the query history of all the users together with their collaboration levels.

This chapter is organized as follows. Sect. 15.2 presents related work. Sect. 15.3 introduces a general framework for the inference detection system, which includes the knowledge acquisition module, semantic inference model and violation detection module. Sect. 15.4 discusses how to acquire and represent knowledge that could generate inference channels. Sect. 15.5 integrates all possible inference channels into a Semantic Inference Model which can be instantiated and then mapped into a Bayesian network to reduce the computation complexity for data inference. As shown in Sect. 15.6, we are able to detect inference violation at query time for both individual user and multiple collaborative users. Sect. 15.7 presents an example to illustrate the use of the proposed technique for collaboration inference detection. Sect. 15.8 presents collaboration level experiments and their estimations. Sect. 15.9 discusses the robustness of inference detection and threshold determination via sensitivity analysis. Sect. 15.10 presents the conclusion.

15.2 Related Work

Database inferences have been extensively studied. Many approaches to address the inference problem were presented in [20]. Particularly, Delugach and Hinke used database schema and human-supplied domain information to detect inference problems during database design time [18, 28, 29]. Garvey, et al. developed a tool for database

designers to detect and remove specific types of inference in a multilevel database system [22]. Both approaches use schema-level knowledge and do not infer knowledge at the data level. These techniques are also used during database design time and not at run time. However, Yip, et al. pointed out the inadequacy of schema-level inference detection, and he identifies six types of inference rules from the data level that serve as deterministic inference channels [47]. In order to provide a multilevel secure database management system, an inference controller prototype was developed to handle inferences during query processing. Rule-based inference strategies were applied in this prototype to protect the security [43]. Further, since data update can affect data inference, [21] proposed a mechanism that propagates update to the user history files to ensure no query is rejected based on the outdated information. To reduce the time in examining the entire history log in computing inference, [44] proposed to use a prior knowledge of data dependency to reduce the search space of a relation and thus reduce the processing time for inference. Open inference techniques were proposed to derive approximate query answering when network partitions occurred in distributed databases. Feasible open inference channels can be derived based on query and database schema [10].

The previous work on data inference mainly focused on deterministic inference channels such as functional dependencies. The knowledge is represented as rules and the rule body exactly determines the rule head. Although such rules are able to derive sound and complete inference, much valuable non-deterministic correlation in data is ignored. For example, salary ranges may not deterministically depend on the ranks. Further, many semantic relationships, as well as data mining rules, can not be specified deterministically. To remedy this shortcoming, we propose a probabilistic inference approach to treat the query-time inference detection problem. The contribution of our research consists of: 1) Derive probabilistic data dependency, relational database schema and domain-specific semantic knowledge and represent them as probabilistic inference channels in a Semantic Inference Model. 2) Map the instantiated Semantic Inference Model into a Bayesian network for efficient and scalable inference computation. 3) Propose an inference detection framework for multiple collaborative users.

15.3 The Inference Framework

The proposed inference detection system consists of three modules, as shown in Fig. 15.1: knowledge acquisition, semantic inference model (SIM), and security violation detection including user collaboration relation analysis.

The *Knowledge Acquisition* module extracts data dependency knowledge, data schema knowledge and domain semantic knowledge. Based on the database schema and data sources, we can extract data dependency between attributes within the same entity and among entities. Domain semantic knowledge can be derived by semantic links with specific constraints and rules. A semantic inference model can be constructed based on the acquired knowledge.

The *Semantic Inference Model (SIM)* is a data model that combines data schema, dependency and semantic knowledge. The model links related attributes and entities as well as semantic knowledge needed for data inference. Therefore SIM represents

all the possible relationships among the attributes of the data sources. A *Semantic Inference Graph* (*SIG*) can be constructed by instantiating the entities and attributes in the SIM. For a given query, the SIG provides inference channels for inferring sensitive information.

Based on the inference channels derived from the SIG, violation detection combines the new query request with the request log, and it checks to see if the current request exceeds the pre-specified threshold of information leakage. If there is collaboration according to collaboration analysis, the *Violation Detection* module will decide whether to answer a current query based on the acquired knowledge among the malicious group members and their collaboration level to the current user.

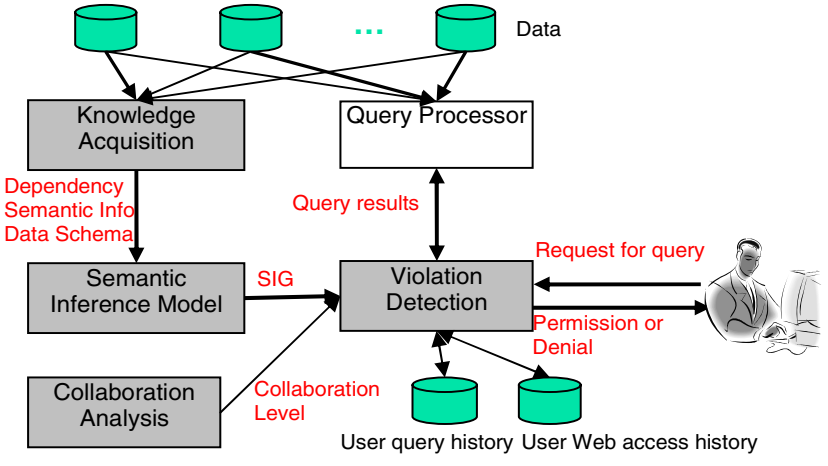


Fig. 15.1. The framework for an Inference Detection System

15.4 Knowledge Acquisition for Data Inference

Since users may pose queries and acquire knowledge from different sources, we need to construct a semantic inference model for the detection system to track user inference intention. The semantic inference model requires the system to acquire knowledge from data dependency, database schema and domain-specific semantic knowledge. This section will discuss how to acquire that knowledge.

15.4.1 Data Dependency

Data dependency represents causal relationships and non-deterministic correlations between attribute values. Because of the non-deterministic nature, the dependency between two attributes A and B is represented by conditional probabilities $p_{ilj}=Pr(B=b_i|A=a_j)$. Thus, the non-deterministic data dependency is a more general representation than the relational functional dependency or other types of deterministic relationships. There are two typies of non-deterministic data dependencies as defined in the Probabilistic Relational Model [19, 24]: *dependency-within-entity* and *dependency-between-related-entities*, as defined in the following.

Dependency-within-entity: Let A and B be two attributes in an entity E; if B depends on A, then for each instance of E, its value of attribute B depends on its value of attribute A with a probability value. To learn the parameter of dependency-within-entities from relational data, from a relational table that stores entity E, we can derive the conditional probabilities $p_{ilj}=Pr(B=b_i|A=a_j)$ via a sequential scan of the table with a counting of the occurrences of A, B, and co-occurrences of A and B.

Dependency-between-related-entities: Let A be an attribute in entity E_1 and C be an attribute in E_2, and E_1 and E_2 are related by R, which is a relation that can be derived from database schema. If C depends on A, then only for related instances of E_1 and E_2, the value of attribute C in E_2 instances depends on the value of attribute A in related instances of E_1. Such dependency-between-related-entities only exists for related instances of entities E_1 and E_2. The parameters of dependency-between-related-entities can be derived by first joining the two entity tables based on the relation R and then scanning and counting the frequency of occurrences of the attribute pair in the joined table. If two entities have an m-to-n relationship, then the associative entity table can be used to join the related entity tables to derive dependency-between-related-entities [12].

15.4.2 Database Schema

In relational databases, database designers use data definition language to define data schema. The owners of the entities specify the primary key and foreign key pairs. Such pairing represents a relationship between two entities. If entity E_1 has primary key *pk*, entity E_2 has foreign key *fk*, and $e_1.pk=e_2.fk$, then dependency-between-related-entities from attribute A (in e_1) to attribute C (in e_2) can be derived.

15.4.3 Domain-Specific Semantic Knowledge

Other than data dependencies inside relational data sources, outside information such as domain knowledge can also be used for inferences. Specifically, domain-specific semantic relationships among attributes and/or entities can supplement the knowledge of malicious users and help their inference. For example, the semantic knowledge "can land" between Runway and Aircraft implies that the length of Runway should be greater than the minimum Aircraft landing distance, and the width of Runway should be greater than the minimum width required by Aircraft. If we know the runway requirement of aircraft *C-5*, and *C-5* "can land" in the instance of runway *r*, then the values of attributes *length* and *width* of *r* can be inferred from the semantic knowledge. Therefore, we want to capture the domain-specific semantic knowledge as extra inference channels in the Semantic Inference Model.

Semantic knowledge among attributes is not defined in the database and may vary with context. However, from a large set of semantic queries posed by the users, we can extract the semantic constraints [50]. For example, in the WHERE clause of the following query, clauses #3 and #4 are the semantic conditions that specify the semantic relation "can land" between entity Runways and entity Aircrafts. Based on this

query, we can extract semantic knowledge "can land" and integrate it into the Semantic Inference Model shown in Fig. 15.3.[1]

- **Query: Find airports that *can land* a C-5 cargo plane.**

```
SELECT AP.APORT_NM
FROM AIRCRAFTS AC, AIRPORTS AP, RUNWAYS R
WHERE AC.AC_TYPE_NM = 'C-5' and                                    #1
AP.APORT_NM = R.APORT_NM and                                       #2
AC.WT_MIN_AVG_LAND_DIST_FT <= R.RUNWAY_LENGHT_FT and               #3
AC.WT_MIN_RUNWAY_WIDTH_FT <= R.RUNWAY_WIDTH_FT;                    #4
```

15.5 Semantic Inference Model

The Semantic Inference Model (SIM) represents dependent and semantic relationships among attributes of all the entities in the information system. As shown in Fig. 15.2, the related attributes (nodes) are connected by three types of relation links: dependency link, schema link and semantic link.

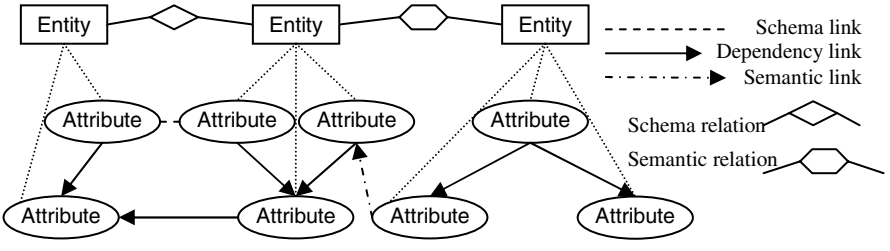


Fig. 15.2. A Semantic Inference Model. Entities are interconnected by schema relations (diamond) and semantic relations (hexagon). The related attributes (nodes) are connected by their data dependency, schema and semantic links.

Dependency link connects dependent attributes within the same entity or related entities. Consider two dependent attributes A and B. Let A be the parent node and B be the child node. The degree of dependency from B to A can be represented by the conditional probabilities pi|j =Pr(B=bi|A=aj). The conditional probabilities of the child node given all of its parents are summarized into a conditional probability table (CPT) that is attached to the child node. For instance, Fig. 15.3(b) shows the CPT of the node "TAKEOFF_LANDING_CAPACITY" of the SIM in Fig. 15.3(a). The conditional probabilities in the CPT can be derived from the database content [19, 24]. For example, the conditional probability Pr(B=bi|A=aj) can be derived by counting the co-occurrence frequency of the event B=bi and A=a_j and dividing it by the occurrence frequency of the event $A=a_j$.

[1] Clearly, the set of the semantic queries may be incomplete, which can result in the semantic knowledge being incomplete as well. However, additional semantic knowledge can be appended to the Semantic Inference Model as the system gains more semantic queries. The system can then reset to include the new knowledge. Otherwise, this will result in inference with knowledge update and is beyond the scope of this chapter.

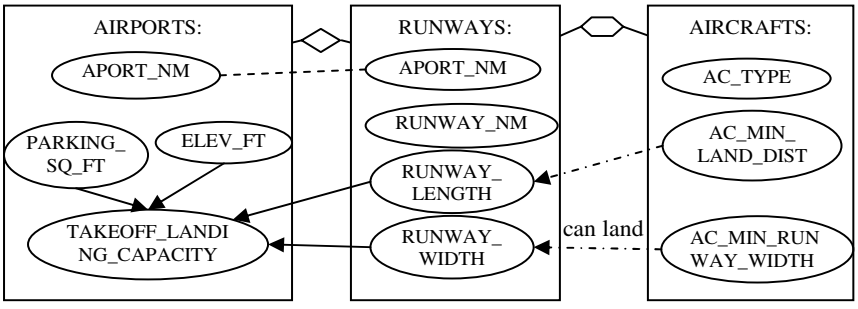


Fig. 15.3(a). A Semantic Inference Model example for Airports, Runways and Aircraft

	Conditional Probability of TAKEOFF_LANDING_CAPACITY																							
Cond	parking_sq_ft	small												large										
	elev_ft	low						high						low						high				
	runway_length	short		medium		long		short		medium		long		short		medium		long		short		medium		long	
	runway_width	narrq	wide	narrq	wide	narrq	wide	narrq	wide	narrq	wide	narrq	wide	narrq	wide	narrq	wide	narrq	wide	narrq	wide	narrq	wide	narrq	wide
Takeoff_Landing_Cap	small	0.9	0.8	0.8	0.7	0.7	0.6	0.95	0.85	0.85	0.75	0.75	0.65	0.85	0.75	0.75	0.65	0.4	0.3	0.8	0.7	0.55	0.5	0.4	0.25
	large	0.1	0.2	0.2	0.3	0.3	0.4	0.05	0.15	0.15	0.25	0.25	0.35	0.15	0.25	0.25	0.35	0.6	0.7	0.2	0.3	0.45	0.5	0.6	0.75

Fig. 15.3(b). Conditional probability table (CPT) for the attribute "TAKEOFF_ LANDING_ CAPACITY" summarizes its dependency on the four parent nodes. For example, *Pr(Takeoff_ landing_capacity=small | Parking_sq_ft=small, Elev_ft =low, Runway_length=short, Runway_width=narrow)=0.9*. The conditional probabilities in the CPT can be derived from the database content.

node and set the value of the source node to "unknown." In this case, the source and target node are independent, i.e., $Pr(T=t_i|P_1=v_1, \ldots P_n=v_n, P_S=unknown) = Pr(T=t_i|P_1=v_1, \ldots P_n=v_n)$. When the semantic relationship is known, the conditional probability of the target node is updated according to the semantic relationship and the value of the source node. If the value of the source node and the semantic relation are known, then $Pr(T=t_i| P_1= v_1, \ldots P_n= v_n, P_S=s_j)$ can be derived from the specific semantic relationship. For example, in Fig. 15.4(b), the semantic relationship determines that $Pr(T=t_1| P_1, \ldots P_n, P_S=s_1)=0.6$ and $Pr(T=t_1| P_1, \ldots P_n, P_S=s_2)=0.8$.

Schema link connects an attribute of the primary key to the corresponding attribute of the foreign key in the related entities. For example, in Fig. 15.3(a), APORT_NM is the primary key in AIRPORTS and foreign key of RUNWAYS. Therefore, we connect these two attributes via schema link.

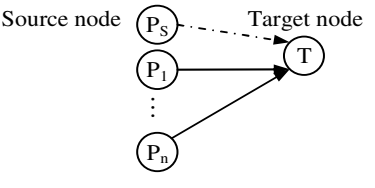


Fig. 15.4(a). Target node T with semantic link from source node P_S and dependency links from parents $P_1, \ldots, P_n$

	Conditional Probability of T											
Cond	Ps	unknown				s1				s2			
	P1	v11		v12		v11		v12		v11		v12	
	Pn	vn1	vn2	vn1	vn2	vn1	vn2	vn1	vn2	vn1	vn2	vn1	vn2
T	t1	0.5	0.3	0.4	0.2	0.6	0.6	0.6	0.6	0.8	0.8	0.8	0.8
	t2	0.5	0.7	0.6	0.8	0.4	0.4	0.4	0.4	0.2	0.2	0.2	0.2

Fig. 15.4(b). The CPT of target node T summarizes the conditional probabilities of T given values of P_S and P_1, …, P_n. For example, $Pr(T=t_1 \mid P_S=unknown, P_1=v_{11}, P_n= v_{n1})=0.5$.

Semantic link connects attributes with a specific semantic relation. To evaluate the inference introduced by semantic links, we need to compute the CPT for nodes connected by semantic links. Let T be the target node of the semantic link, P_S be the source node, and P_1, …, P_n be the other parents of T, as shown in Fig. 15.4(a). The semantic inference from a source node to a target node can be evaluated as follows.

If the semantic relation between the source and the target node is unknown or if the value of the source node is unknown, then the source and target node are independent. Thus, the semantic link between them does not help inference. To represent the case of the unknown semantic relationship, we need to introduce the attribute value "unknown" to the source.

For example, the semantic relation "can land" between Runway and Aircraft (Fig. 15.5(a)) implies that the length of Runway is greater than the minimum required Aircraft landing distance. So the source node is aircraft_min_land_dist, and the target node is runway_length. Both attributes can take three values: "short," "medium" and "long." First, we add value "unknown" to source node aircraft_min_land_dist and set it as a default value. Then we update the conditional probabilities of the target node to reflect the semantic relationship. Here, we assume that runway_length has an equal probability of being short, medium or long. When the source node is set to "unknown," the runway_length is independent of aircraft_min_land_dist; when the source node has a known value, the semantic relation "can land" requires runway_length is greater than or equal to aircraft_min_land_dist. Thus, the corresponding CPT for the node runway_length is shown in Fig. 15.5(b).

15.5.1 Computation Complexity of Constructing Semantic Inference Model

A SIM consists of linking related attributes (structure) and their corresponding conditional probabilities (parameters). Given a relational database, the learning of a SIM can be decomposed into two tasks: parameter learning and structure learning. In the first task, we assume that the structure of the SIM is known, i.e., the links between attributes are fixed, and our goal is to derive the conditional probability tables for each attribute. Since the parameters of semantic link are determined by its semantic constraint, let us now consider the computation complexity on learning parameters of data dependencies. Consider that given structure S has m attributes, each attribute A_i in table T_j has a set of parents $P(A_i)$. If all parents of Ai are in the same table with Ai, then the CPT of Ai can be derived by a single scan of T_j. If attribute A_i has a parent from related entity table T_k, then scanning on the joined table of T_j and T_k is needed to derive the CPT of A_i. In the worst case, the parameters can be learned in

$O(m\prod_i n_i)$ time, where m is the total number of attributes in the model and n_i is the size of the i^{th} table. When the number of dependency-between related-entities is limited, the parameter learning can be reduced to approximately $O(\sum_i m_i n_i)$ where m_i (< m) is the number of attributes in the i^{th} table.

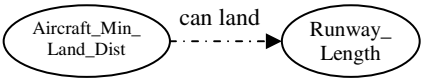


Fig. 15.5(a). The semantic link "can land" between "Aircraft_Min_Land_Dist" and "Runway_Length"

	Conditional Probability of Runway_length				
Cond	aircraft_min	unknown	short	medium	long
Runway_ Length	short	0.33	0.33	0	0
	medium	0.33	0.33	0.5	0
	long	0.33	0.33	0.5	1

Fig. 15.5(b). Conditional Probability Table for Runway_length

If the structure of the SIM is not given by domain experts, we can generate a set of candidate structures with their corresponding parameters, and select the one that best matches the data sources. Algorithms for searching good dependency structures can be found in [19, 23].

15.5.2 Semantic Inference Graph

To perform inference at the instance level, we instantiate the SIM with specific entity instances and generate a semantic inference graph (SIG), as shown in Fig. 15.6. Each node in the SIG represents an attribute for a specific instance. To highlight the attributes of an entity instance, we group all the attributes of the instance into a rectangular box. Related attributes are then connected via instance-level dependency links, instance-level schema links and instance-level semantic links. The attribute nodes in SIG have the same CPT as in SIM because they are just instantiated versions of the attributes in entities. As a result, the SIG represents all the instance-level inference channels in the SIM.

Instance-level dependency link: When a SIM is instantiated, the dependency-within-entity is transformed into dependency-within-instance in the SIG. Similarly, the dependency-between-related-entities in the SIM is transformed into a dependency between two attributes in the related instances. This type of dependency is preserved only if two instances are related by the instantiated schema link. That is, if attribute B in instance e_2 depends on attribute A in instance e_1, and instances e_1 and e_2 are related by R denoted as R(e_1, e_2), then there is a dependency-between-related-instances from B to A.

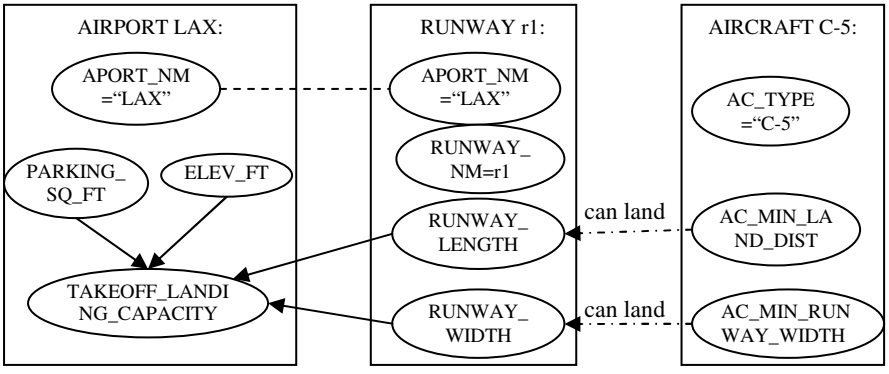


Fig. 15.6. The Semantic Inference Graph for airport instance (LAX), with runway r1 and aircraft C-5

Instance-level schema link: The schema links between entities in the SIM represent "key, foreign-key" pairs. At instance level, if the value of the primary key of an instance e_1 is equal to the value of the corresponding foreign key in the other instance e_2 which can be represented as $R(e_1, e_2)$, then connecting these two attributes will represent the schema link at the instance level. Otherwise, these two attributes are not connected.

Instance-level semantic link: At the instance level, assigning the value of the source node to "unknown" disconnects the semantic link between the attributes of two instances. On the other hand, if two instances have a specific semantic relation, then the inference probability of the target node will be computed based on its CPT and the value of the source node.

15.5.3 Evaluating Inference in Semantic Inference Graph (SIG)

For a given SIG, there are attribute dependencies within an entity, between related entities, and semantic relationships among attributes. As a result, there are many feasible inference channels that can be formed via linking the set of dependent attributes. Therefore, we propose to map the SIG to a Bayesian network to reduce the computational complexity in evaluating user inference probability for the sensitive attributes.

For any given node in a Bayesian network, if the value of its parent node(s) is known, then the node is independent of all its non-descending nodes in the network [26, 27, 30, 39, 40]. This independence condition greatly reduces the complexity in computing the joint probability of nodes in the network. More specifically, let x_i be the value of the node X_i, pa_i be the values of the parent nodes of X_i, then $P(x_i|pa_i)$ denotes the conditional probability of x_i given pa_i where i=1,2,…,n. Thus, the joint probability of the variables x_i is reduced to the product of $P(x_i|pa_i)$:

$$P(x_1, \ldots, x_n) = \prod_i P(x_i \mid pa_i) \tag{15.1}$$

The probability for users to infer the sensitive node $S=s$ given the evidences $D_i=d_i$, $i=1, 2,\ldots, n$ is:

$$P(s \mid d_1, d_2 \ldots, d_n) = \frac{P(s, d_1, d_2 \ldots, d_n)}{P(d_1, d_2 \ldots, d_n)} \tag{15.2}$$

which can be further computed using Eq. 15.1. Thus, the probability of inferring a sensitive node can be computed from the conditional probabilities in the Bayesian network. Many algorithms have been developed to efficiently perform such calculations [16, 31, 35, 51, 52].

The Probabilistic Relational Model (PRM) is an extension of the Bayesian network that integrates schema knowledge from relational data sources [19, 23, 24]. Specifically, PRM utilizes a relational structure to develop *dependency-between-related-entities*. Therefore, in PRM an attribute can have two distinct types of parent-child dependencies: *dependency-within-entity* and *dependency-between-related-entities*, which match the two types of dependency links in the SIM. Since the semantic links in the SIM are similar to dependency links, we can convert each SIM to a PRM-based model. The corresponding Bayesian network can be generated after instantiating the model to instance level. Thus, for a given network, the probability of inferring a specific sensitive attribute can be evaluated via efficient Bayesian inference algorithms. In our test bed, we use SamIam [41], a comprehensive Bayesian network tool developed by the Automated Reasoning Group at UCLA, to compute the inference. The computation complexity for exact inference is mostly $O(n \cdot \exp(w))$, where n is number of nodes and w is the tree-width of the network [8, 2, 13, 17, 31, 52] and is scalable.

15.6 Inference Violation Detection for Individual User

Semantic inference graphs provide an integrated view of the relationships among data attributes, which can be used to detect inference violation for sensitive nodes. In such a graph, the values of the attributes are set according to the answers of the previous posted queries. Based on the list of queries and the user who posted those queries, the value of the inference will be modified accordingly. If the current query answer can infer the sensitive information greater than the pre-specified threshold, then the request for accessing the query answer will be denied [9].

Consider the example in Fig. 15.3. Let the TAKEOFF_LANDING_ CAPACITY of any airport be the sensitive attribute, and it should not be inferred with probability greater than 70%. If the user has known that: 1) Aircraft C-5 can land in airport LAX runway r1; 2) C-5 has aircraft_min_land_dist = long and aircraft_min_runway_width = wide. Then this user is able to infer the sensitive attribute "LAX's TAKEOFF_ LANDING_ CAPACITY = large" via Eqs. 15.2 and 15.1 with probability 58.30%, as shown in Fig. 15.7(a).

Now if the same user poses another query about the "Parking_sq_ft of LAX" and if this query is answered (as shown in Fig. 15.7(b), LAX_Parking_Sq_Ft=large), then the probability of inferring LAX_TAKEOFF_LANDING_CAPACITY = large by this user will increase to 71.50%, which is higher than the pre-specified threshold. Thus, this query request should be denied.

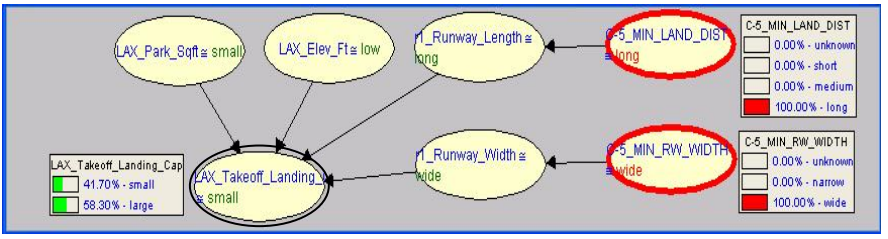


Fig. 15.7(a). Example of inference violation detection for single user. This is a portion of the Bayesian network for the transportation mission planning. The probability distribution of each node is shown in a rectangular box. The values of the bold nodes are given by previous query answers; the probability values of sensitive nodes are inferred.

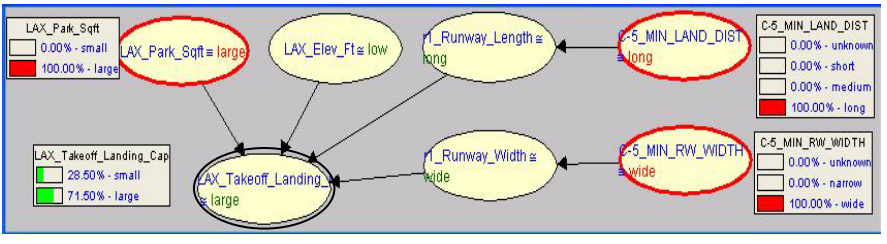


Fig. 15.7(b). Given the additional knowledge "LAX_Parking_Sq_Ft=large", the probability for inferring the sensitive information "LAX_TAKEOFF_LANDING_ CAPACITY =large" is increased to 71.50%

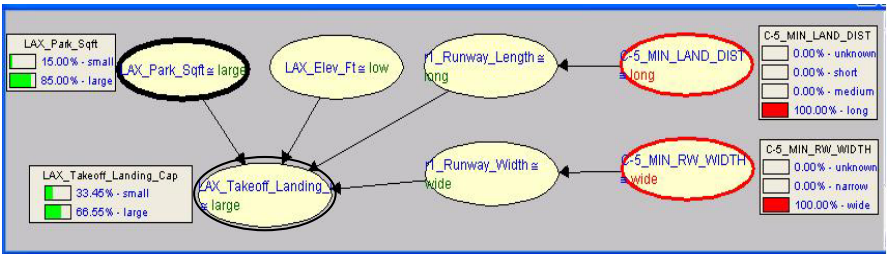
15.7 Inference Violation Detection for Collaborative Users

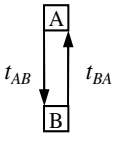
15.7.1 Collaborative Inference Violation Detection

To extend our inference violation detection module for collaborative users, we first need to define the collaboration level among users that is a metric for measuring the percentage of useful information flow from the source to the recipient. The collaboration level depends on two aspects: authoritativeness of the information source and the fidelity of the communication channel between the source and recipient. Authoritativeness can be determined by the reputability and authority of the information provider; fidelity depends on such factors as the willingness of the provider to release information, and/or the recipient's understandability of the received information. We use collaboration levels to combine the source authoritativeness and channel fidelity. The higher the collaboration level between the pair of collaborators, the higher their collaboration effectiveness will be. More discussion of how to derive the collaboration level will be presented in Sect. 8.

Consider users A and B in Fig. 15.8. User B has a collaborative level of 85% for the information from A. Let Q_A and Q_B be the query answer set of user A and user B. User B can combine Q_A with his own previous query answer set Q_B and yield a higher inference probability for the sensitive node. For the example in Fig. 15.7(a), user B

has past query answers Q_B = {C-5_min_land_dist = long, C-5_min_rw_width = wide} and then combines this with his acquired knowledge from user A: Q_A = {LAX_Park_Sqft = large}. Such collaboration increases the inference probability for the sensitive node from 58.30% to 66.55%, as shown in Fig. 15.8. Note that because the collaborative level of B for information from A is 85%, it yields a lower inference probability than the case where user B queries directly about LAX_Parking_Sq_Ft, as in Fig. 15.7(b).





t_{AB}	Conditional Probability		
Cond	A	yes	no
B	yes	0.85	0.15
	no	0.15	0.85

Fig. 15.8. Example of inference violation detection for multiple users. User B knows "C5_min_land_dist=long" and "C5_min_rw_width=wide" from his past query answers. User B also has the knowledge from A "LAX_Park_Sqft =large" with collaborative level 85%. Thus, the probability for user B to infer the sensitive information (shown in double ellipses) "LAX_Takeoff_Landing_Capacity=large" increases to 66.55%.

In general, according to the users' query history, there are two different types of collaborative user pairs, as shown in Fig. 15.9:

Collaboration with non-overlap inference channels: In this case, the two users pose queries on different non-overlap inference channels. The inference probability will be computed based on their combined knowledge discounted by their collaborative level.

Collaboration with overlap inference channels: In this case, the query sets posed by the two users overlap on inference channels. Such overlap may cause the users to have inconsistent belief in the same attribute on the inference channel. Thus, we need to integrate the overlapping knowledge according to the collaborative level to compute the appropriate inference probability.

Case (a) is the simple case of non-overlap inference channels. The influence from user A to user B is given by the collaboration level. Therefore, for user B, the query answers acquired by A (Q_A) can be combined with the query answers that are acquired by B (Q_B), but discounted by B's collaborative level to A. In addition, because Q_A and Q_B are from independent non-overlap inference channels, their inferences to sensitive node S are independent and can be directly combined. Thus the inference probability for the sensitive node can be computed based on the user's knowledge

from his past queries combined with his collaborator's query answers discounted by their respective collaborative level.

For Case (b), the queries posed by user A and user B overlap on their inference channels. Since Q_A and Q_B may cause inconsistent belief on some attribute nodes, these two query answer sets cannot be simply combined. For example, in Fig. 15.10(a), for attribute node X, Q_A indicates A has known $X=x$ and B can believe it with collaboration level t_{AB} ($t_{AB} \leq 1$).

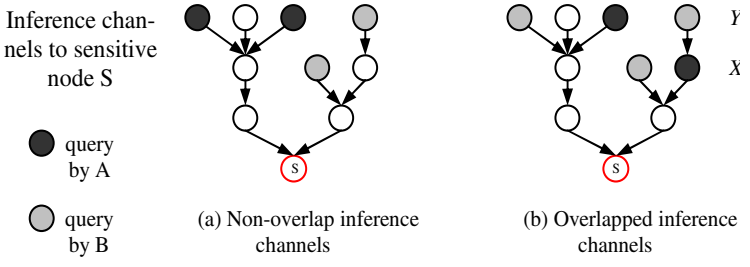
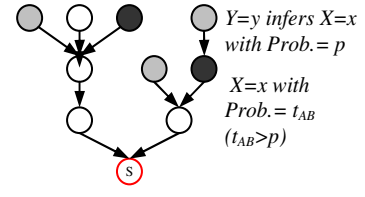
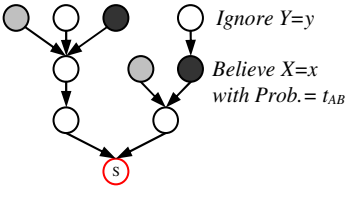


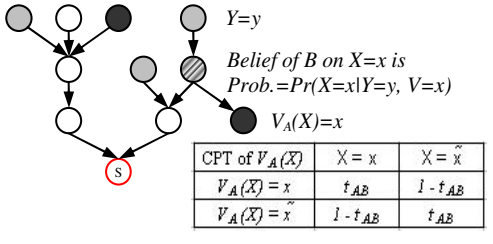
Fig. 15.9. Types of collaborative user pairs posing query sequence on the inference channels



(a) $Q_A$and Q_B cause inconsistent belief on X. From Q_B ($Y=y$), B infers $X=x$ with Prob.= p. From Q_A ($X=x$), B infers $X=x$ with Prob.= t_{AB}.



(b) Believe $X=x$ *with Prob.*= t_{AB} yields inaccurate inference probability to S



CPT of $V_A(X)$	$X = x$	$X = \tilde{x}$
$V_A(X) = x$	t_{AB}	$1 - t_{AB}$
$V_A(X) = \tilde{x}$	$1 - t_{AB}$	t_{AB}

(c) Introduce soft evidence for Q_A ($X=x$) by adding a virtual node $V_A(X)$ as the child of X. User B now believes $X=x$ with *Prob.*=$Pr(X=x|Y=y, V_A(X)=x)$

Fig. 15.10. A virtual node can be used in user B's inference network to resolve inconsistent belief when user B and A overlap on their inference channels

On the other hand, Q_B includes $Y=y$ which can infer $X=x$ with probability p. If $p \neq t_{AB}$, then Q_A and Q_B can cause B to have inconsistent belief on attribute X. Without loss of generality, we assume $p<t_{AB}$ for this example.

One approach to reconciling such inconsistent belief is to assume B will always choose to maximize his inference probability. Therefore, as shown in Fig. 15.10(b), B only follows A's advice ($X=x$ with $prob.=t_{AB}$) and ignore his own acquired knowledge ($Y=y$ infers $X=x$ with $prob.=p$). However, such a "max-inference" approach is not always correct, since people's belief is often strengthened by the confirmation and reduced by the conflicting knowledge. To represent the integration of inconclusive belief, we introduce the concept of *soft evidence* in probability calculus [14]. Soft evidence is inconclusive or unreliable information, as in the previous example, A tells B that $X=x$ and B only believes it with t_{AB} ($t_{AB} < 1$). For user B, $X=x$ is inconclusive knowledge, and therefore it needs to be set as soft evidence. To specify the soft evidence, we use the *Virtual Evidence* method developed in [14]. As shown in the Fig. 15.10(c), this method first adds a virtual attribute node $V_A(X)$ to be the child of the attribute node X to represent the event of receiving the soft evidence of X, that is, A tells B about $X=x$. Then the conditional probability of the virtual node is determined by the reliability of the soft evidence. In our example, both $\Pr(V_A(X) = x \mid X = x)$ and $\Pr(V_A(X) = \bar{x} \mid X = \bar{x})$ are determined by user B's *collaboration level* of information from A t_{AB}. Thus, the soft evidence can be integrated into the user's own knowledge. In the example, if originally B is ignorant about X, once A tells B about $X=x$, B will believe $X=x$ with probability t_{AB}. If originally B can infer X with knowledge $Y=y$, then his current belief in $X=x$ can be computed as $\Pr(X = x \mid Y = y, V_A(X) = x)$. Thus, we are able to integrate queries on overlapped inference channels from multiple collaborators based on their corresponding collaboration levels.

Therefore, for any type of two collaborative users, we can integrate one's knowledge to the other and detect their inference towards sensitive data. When any user poses a query, the system not only checks to see if the query requester can infer sensitive data above the threshold with a query answer, it also checks the other team members to guarantee that the query answer will not indirectly let them infer the sensitive attribute. We can iteratively generalize the above approach to an N-collaborator case. In general, when there are N collaborative users in the team, the violation detection system tracks the query posed by every team member. A query should be denied if the query answer will increase the certainty of any team member to infer the sensitive data above the pre-specified threshold.

```
1.  Assume: current query request Q, malicious team M, sensitive data S, threshold of S is T;
2.  List(M) = sort team members M in descending order of inference probability to S;
3.  While(List(M) is not empty) {
4.     m = first member in List(M) with highest inference probability;
5.     max_col = the maximum collaborative level from any member in List(M)
                 to the query requester;
6.     real_col = m's collaborative level to query requester;
7.        If (m integrate answer to Q with max_col can get inference probability < T)
8.        Then {answer query Q; goto end;}
```

```
9.        Else
10.         If (m integrate answer to Q with real_col can get inference probability >= T)
11.         Then {deny query Q; goto end;}
12.         Else { List(M) = List(M) – m;}
13.   }
```

An inference violation detection algorithm for N collaborative users

We can use the above greedy algorithm to efficiently decide to either answer or deny a query request from any team member. We first sort all N members by their inference probability to the sensitive attribute and start with the member having the highest inference probability. We also compute every member's collaborative level to the query requester and determine the max collaborative level. Suppose that the member with the highest inference probability integrates the current query answer adjusted by the maximum collaborative level and still cannot infer sensitive data above the threshold. Then we can stop checking the rest of the team members and answer the query. This is because no other member in the team will be able to make a higher inference. If the member with the highest inference probability integrates the query answer adjusted by his collaborative level to the requester and can infer the sensitive data above or equal to the threshold, then we can stop checking and deny this query. Otherwise, we continue on to another member with the next highest inference probability until a decision can be made.

15.7.2 An Example of Inference Violation Detection for Collaborative Users

A set of data sources for transportation and related facilities is available for mission planning. Due to the classified nature of the data sources, users can only access limited amounts of information. Malicious users want to identify whether a specific facility is capable of launching certain classified missions. However, the users are unable to access all the information that is required to derive the conclusion. Therefore, they apply inference techniques to infer the otherwise inaccessible information. In the following example, we shall demonstrate how our detection system prevents these users from accessing the relevant information.

As shown in Fig. 15.11, the transportation and facility data sources consist of four types of information: 1) takeoff and landing activities and capacity of the airport, such as parking_area, runway_length, runway_width, aircraft landing requirements etc.; 2) equipment handling capacity, such as weapons, human experts, loading facility; 3) airport cargo and warehouse capacity and activities, such as daily cargo handling capacity, warehouse space; and 4) fueling storage and consumption. Based on these entities and attributes, we can derive the dependency links between attributes, the schema links that join different aspects of information together for each airport. Furthermore, based on the following set of semantic queries:

- Query1: which airports *can land* a C-5 cargo plane?
- Query2: which airports have the loading facility that *can load* weapon type HC-1?
  Query3: which aircraft *can carry* weapon type HC-1?

We can extract the semantic knowledge for "can land," "can load" and "can carry" for semantically linking the related attributes, as shown in Fig. 15.11.

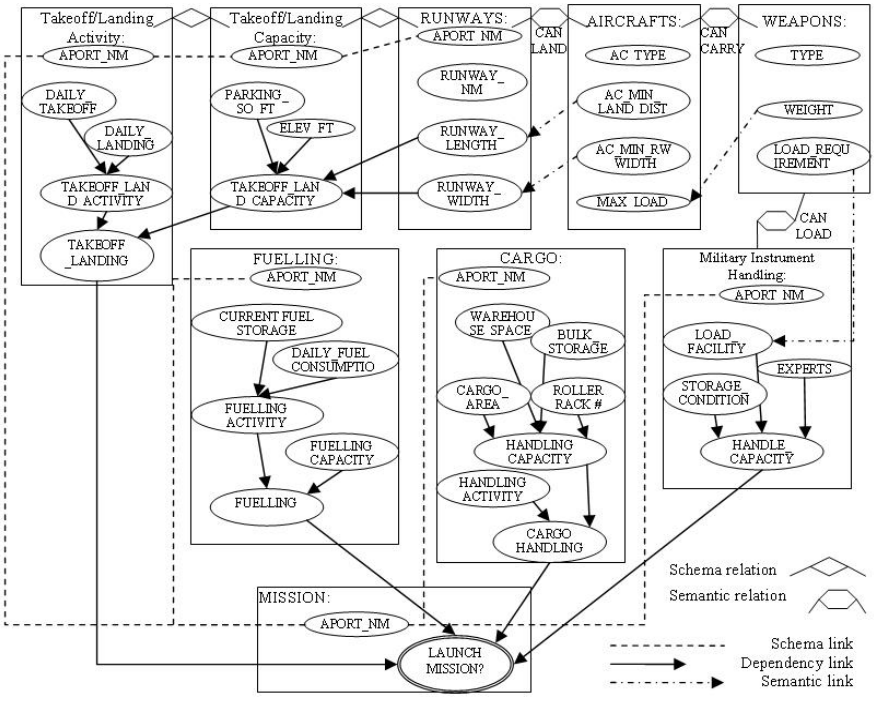


Fig. 15.11. The SIM for a transportation mission planning example

Based on these dependency links, schema links and semantic links, a reduced semantic inference model was constructed (Fig. 15.11) to represent all the possible inference channels between data attributes for all the entities. There are four data sources which yield four main inference channels to the mission entity: takeoff_ landing to launch_mission; fueling to launch_mission; cargo_handling to launch_mission and handle_capacity to launch_mission. Each of the main inference channels consists of many local inference channels. To carry out the inference computation, we need to generate a semantic inference graph (SIG) by substituting the specific instance to the semantic inference model. The corresponding Bayesian network representation mapped from the SIG for airport "LAX" is shown in Fig. 15.12.

Let "Launch Mission?" be the sensitive attribute. The violation detection module examines each user's past query log, as well as the current query request. The probability to infer "Launch Mission?" in the Bayesian network will be evaluated before answering each query. If answering the current query increases the certainty of inferring the sensitive attribute above the pre-specified threshold, then the query will be denied. Let the pre-specified threshold for launch mission be 60%, and the users have prior knowledge of: 1) Aircraft C-5 can land in airport LAX; 2)Airport LAX can load weapon HC-1. When user A poses the sequence of queries shown in Table 15.1, each query answer will update his certainty of inferring the "Launch Mission? = yes" (as shown in the table). The same is true for user B when he poses the queries in Table 15.2.

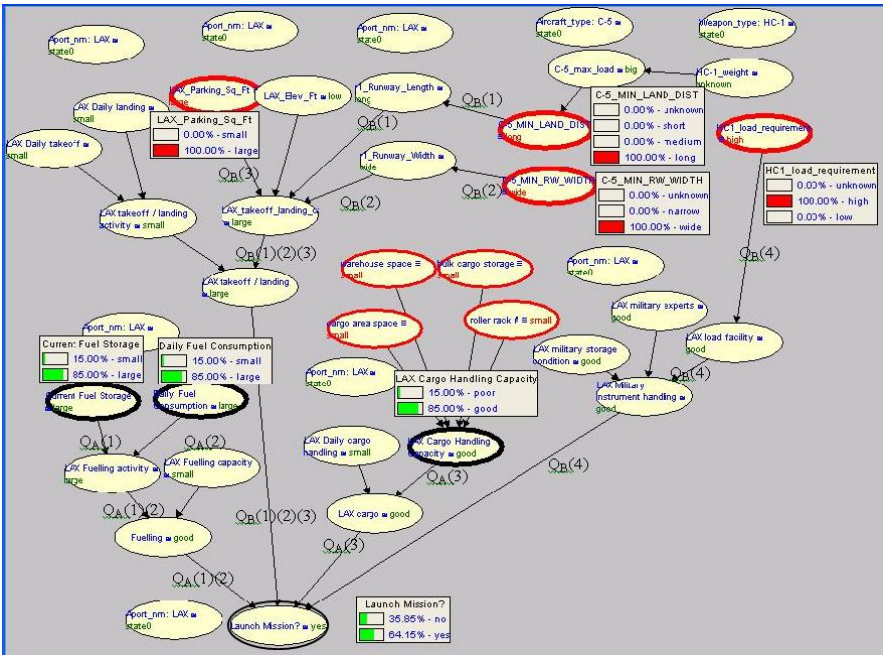


Fig. 15.12. The Bayesian network for the mission planning example. The bold nodes represent user queried attributes. Knowledge from the query answers can be accumulated along the inference channels towards the sensitive attribute. The inference channels used by each query are labeled by its query identifier. The collaborative level from user A of 85% are shown in the probability distribution boxes of $Q_A(1)$, $Q_A(2)$ and $Q_A(3)$. When all the seven queries are answered, user B can infer the sensitive attribute (shown in double ellipses) with a certainty of 64.15%.

Tables 15.1 and 15.2 are assuming that user A and user B do not collaborate. Neither A or B are getting enough information to infer the sensitive attribute above the threshold, thus all the queries are answered. However, based on the questionnaires collected from these two users, we notice that they are collaborators with an 85% collaborative level from B to A for this specific "airport mission" task. Therefore, the

Table 15.1. The inference probability of "Launch Mission? = yes" after answering user A's queries. The probabilities are computed from the Bayesian network in Fig. 15.12.

Query Set of A $Q_A(i)$	$Answer_i$	Pr(Launch_mission? = yes | answer$_1$,…, answer$_i$)
What is current_fuel_storage of airport LAX?	large	52.01%
What is current_fuel_consumption of LAX?	large	56.50%
What is cargo_handling_capacity of LAX?	good	59.80%

Table 15.2. The inference probability of "Launch Mission? = yes" after answering user B's queries. The probabilities are computed from the Bayesian network in Fig. 15.12.

Query Set of B $Q_B(i)$	$Answer_i$	Pr(Launch_mission?=yes | answer$_1$,…, answer$_i$)
What is the min_land_dist of aircraft C-5?	long	50.31%
What is the min_rw_width of aircraft C-5?	wide	50.85%
What is the parking_area_sq_ft of airport LAX?	large	52.15%
What is the load_requirement of weapon type HC-1?	high	57.15%

knowledge from their query answers can be combined for collaborative inference. If we examine their query set Q_A and Q_B on the SIM, we notice that they do not have overlapping inference channels. This is because Q_A focused on the fueling and cargo storage of the airport while Q_B focused on the takeoff and landing activities and military instrument handling. Thus, users A and B belong to the "non-overlap inference channels" case as shown in Fig. 15.9. We can directly integrate their knowledge from query set answers based on their collaboration relation. Thus user B can integrate Q_A into Q_B and adjust the inference probability using their respective collaborative level, as shown in Table 15.3.

Table 15.3. User B integrates user A's query set Q_A into his own query set Q_B. The Bayesian network is used to compute the inference probability in accordance with the posed query sequence and adjusted by the collaborative levels of the corresponding answers.

Integrated Query Set of B (i)	$Answer_i$	Collaborative Level t_i (%)	Pr(Launch_mission? =yes |t_1*answer$_1$,…, t_i*answer$_i$)
Q_B(1)What is min_land_dist of aircraft C-5?	long	100%	50.31%
Q_B(2)What is min_rw_width of aircraft C-5?	wide	100%	50.85%
Q_A(1)What is current_fuel_storage of LAX?	large	85%	52.39%
Q_A(2)What is current_fuel_consumption of LAX?	large	85%	55.54%
Q_B(3)What is parking_area_sq_ft of LAX?	large	100%	56.84%
Q_A(3)What is cargo_handling_capacity of LAX?	good	85%	59.15%
Q_B(4)What is load_requirement of weapon HC-1?	high	100%	64.15%

From Table 15.3, we note that the last query posed by user B will infer sensitive information with probability higher than the pre-specified threshold of 60%. Therefore, $Q_B(4)$ should be denied by the violation detection module. In contrast, in the non-collaborative case as shown in Tables 15.1 and 15.2, all the above queries can be answered.

15.8 Collaboration Level

As defined in Sect. 15.7, a collaboration level (CL) is a metrics that can be used to estimate the collaborative inference by a group of malicious users. CL consists of two factors: information authoritativeness and communication channel fidelity. In this section we shall first conduct a set of two experiments to validate the premise of the proposed metrics and then propose a technique to estimate the parameters.

15.8.1 Experimental study of Collaboration Level

Since both information authoritativeness and fidelity are user-sensitive, we conducted an experiment using the students in one of the authors' classes as test subjects. The experiment was used as homework for the class to ensure their participation. Further, to ensure that the experiment was carried out honestly, the experiment outcome would not affect their grades. However, the winner would receive extra credit. A web interface was developed for our inference test bed so that students could pose queries directly to the test bed and receive the answers. The goal of the experiment was to study how information authoritativeness and communication fidelity affect the CL.

Before posing queries for inference, each student needed to register in the system and fill in the necessary background information, including their age, gender, major, year in school, courses taken, GPA, skills, interests, teamwork ability, social activities, friends in the class, etc. The information gave us clues about the information authoritativeness and communication fidelity of the test subjects. Based on the collected background information, we divided the class into five teams of four students to perform collaborative inference. The first team consisted of Ph.D. students with good knowledge in the database area, which should have provided good authoritativeness. The second team members were good friends, which provide good communication fidelity. The other three teams are randomly formed. In the first test, the teams were given the SIG structure based on the database, but not the SIG parameters (CPTs) nor the threshold of the security attribute. Then we allowed each team to pose a fixed number (e.g. four in this experiment) of queries to infer the security attribute. The test bed computed their inference probability after each member posed the query. The system denied the query request if the posed queries exceeded the threshold. The four members in the team could collaborate in the best way possible to increase their inference probability of the security attribute and avoid denial. In order to monitor the team communication, each team also reported its communication methods in selecting the queries, such as email, meeting, voting after debate on query selection.

Fig. 15.13 displays the maximum inference probability for the five teams. In experiment 1a, we observed that team2 reached the highest inference probability. This is because they held meetings to discuss strategies of posing queries and voted if there

was disagreement; therefore, their queries leveraged on each other to get better inference. Team1 asked a set of effective queries that spanned the inference channels based on their knowledge of the SIG structure; therefore they also performed well. This result reveals that both communication fidelity and information source authoritativeness play very important roles in determining collaboration effectiveness.

Fig. 15.13. The inference results of five collaborative teams. In experiment 1a, the teams were given the SIG structure but without the parameters (CPTs) and the threshold (75%) of the security node. In experiment 1b, the teams were given both the SIG structure and the CPTs and the inference threshold (65%) of the sensitive node.

To study how information source authoritativeness affects the CL, we repeated the same experiment in 1b. This time, we let all the teams know the SIG structure, CPTs and the threshold value of the security attribute. With the same fixed number of queries, we noticed that with the additional knowledge of the CPTs and threshold of the security attribute, all the teams improved their maximum inference probabilities. In fact, they were able to ask better queries to improve their inference probability as close to the threshold as possible. This experimental result reveals that the information source authoritativeness (in this case, the quality of queries) does affect CL in the positive way.

In the first set of experiments, we noted that both information source authoritativeness and communication fidelity played a key role in CL and therefore improved inference probability. This motivated us to study these two factors more closely in the second experiment. More specifically, we wished to investigate the collaboration effectiveness under controlled communication fidelity environment. This experiment was carried out in a manner similar to that of experiment 1, except it was conducted in another graduate class in the following quarter. Because of the small class size, we divided the students into two teams, with three members in first two teams. In order to control the communication fidelity, we assigned the communication method for each team. The first team was allowed to have “full collaboration.” Members were required

to meet and discuss query strategies and exchange their query answers in making their selection of queries. The second team was allowed "limited collaboration" in which they could only email each other with their query answers but could not discuss strategy. In terms of authoritativeness, Team2 had more task-specific knowledge (in this case, Bayesian inference) than Team1.

In this experiment, the two teams were given the SIG, its parameters (CPTs) but not the threshold of the sensitive attribute. As shown in Fig. 15.14, although Team2 was restricted in communication, they could still pose effective queries based on their task-specific knowledge to achieve a higher inference probability than Team1. On the other hand, although the first team could freely communicate and discuss, their lack of task-specific knowledge caused their failure in posing the most aggressive queries and, in turn, hurt their inference results. We notice that the second team's knowledge of the task overcame the limitation of their collaboration, and they outperformed the first team.

The above set of experimental results validates our premise that information authoritativeness and communication fidelity are two key parameters that affect collaboration performance.

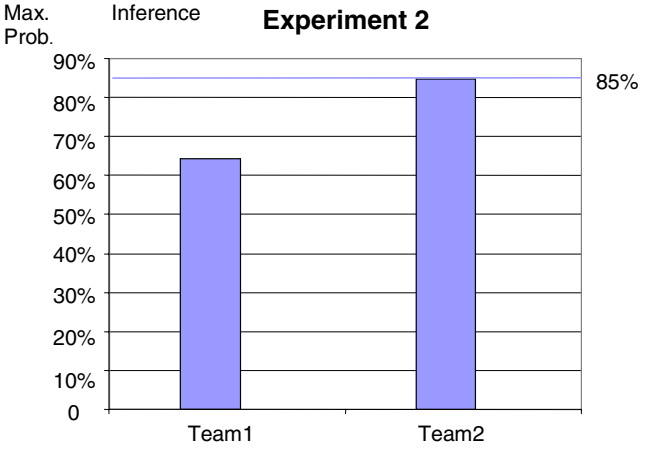


Fig. 15.14. Maximum inference probability for Experiment 2. The teams were given both the SIG structure and the CPTs, but not the inference threshold of the security node which was set at 85%.

15.8.2 Estimation of Collaboration Level

Since the collaboration level consists of two main components: information authoritativeness and communication fidelity, it can be expressed as $CL = g(A, F, e_A, e_F)$, where A is the authoritativeness of the information source, F is the communication fidelity, e_A is the error in estimation of the authoritativeness, and e_F is the error in estimating the fidelity. There are many works on trust negotiation in peer-to-peer networks that are related with e_A and e_F estimation. The interested reader should refer

to papers [1, 7, 32, 48, 49, 45, 42, 46, 15 and 37]. We shall now outline some approaches to estimate A and F under the case that $e_A = e_F = 1$, which corresponds to estimating parameter values in a trusted and honest community.

Estimation of Authoritativeness A: Information authoritativeness represents the task-specific knowledge which can be based on the provider's profile, such as reputation, education, profession, and experience that is related to the task. This authoritativeness can also be enhanced by information derived from the user social network structure . For example, if many individuals (especially highly authoritative ones) indicate user u_i as their friend, then u_i has a significant impact on others and therefore has a higher authority. The link-based similarity analysis (such as page rank) can be used to derive the authority of people [38]. Information authoritativeness can be derived from questionnaire answers and with additional correlated information from web documents or available social network information. In general, Information authoritativeness may be based on a set of multiple attributes that are related to the specific task. The estimation can become more complex and would be beyond the scope of this chapter.

Estimation of Fidelity F: For a given task, the communication fidelity of two collaborators can be based on their closeness on a set of task-sensitive attributes. Based on the registration questionnaire, we can derive their closeness by the similarity computed from the selected attribute set. Additional information from other available sources, such as their web sites and their social networks , can also be used to enhance the estimation.

After estimating A and F, we need to combine them to derive the collaboration inference. One way is to assume they have a linear relationship and thus can be combined linearly. We can then learn the coefficients (such as the weights of A and F) via a set of training data with similar tasks and users.

15.9 Robustness in Inference Detection

Usually security experts or database administrators have some idea of the required level of protection for each security attribute , but they can hardly give a quantitative measurement to describe such protection requirements. Further, in a large database system, the dependency relationship between the security attribute and other attributes is complicated. The inference towards security attribute computed from a Bayesian network depends on both the network topology (qualitative attribute dependencies) and the parameter of the network (conditional probabilities). If a small variation of a given parameter can trigger the inference probability to exceed the threshold, then the inference detection may not satisfy the robustness requirements. This motivates us to find a methodology to systematically quantify the robustness of the threshold for inference violation detection .

Sensitivity measures the impact of small changes in a network parameter on a target probability value or distribution [34]. In other words, a small change in the more sensitive attribute will cause a large impact on the inference probability . Therefore, the sensitivity values of attributes in the network provide an insight to the robustness of inference with respect to the changes in attribute parameter value. In this section, we propose to use the sensitivity analysis results to adjust the security threshold.

15.9.1 Sensitivity Definition

"Sensitivity values are partial derivatives of output probabilities with respect to parameters being varied in the sensitivity analysis . They measure the impact of small changes in a network parameter on a target probability value or distribution" [34]. More formally, for a function *f(x)*, the quantity:

$$\lim_{(x-x_0)\to 0} \frac{(f(x)-f(x_0))/f(x_0)}{(x-x_0)/x_0} \tag{15.3}$$

is typically known as the *sensitivity* of *f* to *x* at x_0, which is the ratio of relative change in output probability over the relative change in the parameter, where x_0 is the initial value of *X*. If we consider the function to be the probability of security node *Y* given the change of attribute node *X*, then the sensitivity for attribute *X* at probability x_0 in a given network *N* with the initial probability of the security node y_{init} can be represented as:

$$\underset{x_0,N,y_{init}}{Sen}(X,Y) = \lim_{(x-x_0)\to 0}\left|\frac{(y-y_0)/y_0}{(x-x_0)/x_0}\right| = \lim_{\Delta x\to 0}\left|\frac{\Delta y/y_0}{\Delta x/x_0}\right| \tag{15.4}$$

The initial probability of the security node is the probability of Y at the state when the set of evidence was given in the network. y_{init} represents the initial probability of *Y*, which is different from y_0 that represents the probability of *Y* when $X = x_0$.

According to this definition, in a Bayesian network , if minor changes to an attribute node's probability can result in a significant change in the output probability of the security node, then this attribute node is considered highly sensitive.

15.9.2 Adjust Security Threshold by Attribute Sensitivity Analysis

To compute the sensitivity of attributes in an inference network, we first identify all inference channels toward each security node so that the sensitivity values for the attributes along the inference channels can be computed. The inference channels include channels coming into the security node and those going out of the security node. For those out-going inference channels , we can treat them as if the channels are coming into the security node by reversing the edges along such channels and revising the corresponding conditional probabilities. This is because, in terms of inference, the security node can be thought of as the "sink" of all information. Regardless of whether the attribute is the ancestor or descendent of the security node, the inference is always from the attribute towards the security node. Thus, we can compute the attribute sensitivities on both in-coming and out-going inference channels .

In a large-scale network, because of the large number of attributes, it is time-consuming to compute the sensitivity value for each attribute on the inference channels . However, for two attribute nodes on the same inference channel , the node that is closer to the security node is more sensitive than the node that is farther from the security node at the same probability value. This difference of sensitivity value between closer and farther nodes is intuitive as closer nodes generally contain more

sensitive information and are more influential on the security node than that of farther nodes. More specifically, the farther node influences the security node through the inference channel which includes the closer node. Therefore, the amount of change at the farther node has the equivalent effect of inferring the security node as a smaller (or equal) amount of change at the closer node. For example, in the inference channel in Fig. 15.15(a), the closest attribute to security node "*Launch Mission?*" is "*Fueling*." Based on Eq. 15.4, the sensitivity of "*Fueling*" is greater than the sensitivity of its parents "*LAX_Fueling_Activity*" for all x_0, as shown in Fig. 15.15(b). Similarly, the sensitivity of "*LAX_Fueling_Activity*" is greater than the sensitivity of "*Daily_Fuel_Consumption*."

By this property, we know that for each inference channel , the attribute node closer to the security node is more sensitive than the farther attribute nodes. So to measure the maximum sensitivity of each inference channel , we only need to consider the sensitivity value of the attribute node on the channel that is closest to the security node to represent the sensitivity of the entire inference channel . Thus, in the entire network, we only need to check the sensitivity of the attributes on an inference channel that is one hop away from the security node.

Each value of the security node is protected by a threshold. For example, we need threshold for "Launch_Mission=Yes" and another threshold for "Launch_ Mission=No" so that the malicious user cannot infer the exact value of this attribute above the thresholds. When the data administrator proposes a threshold value based on the required protection level, he/she can check the sensitivity values of the closest attributes on each inference channel . If one of these inference channels is too sensitive which means that a small change in the attribute value can resulted in exceeding the threshold, then the threshold needs to be tightened to make it less sensitive.

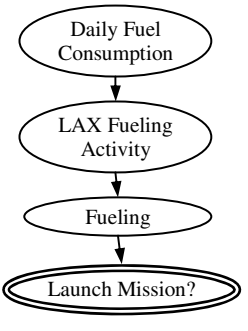


Fig. 15.15(a). A portion of the inference channel in the Bayesian network from the example

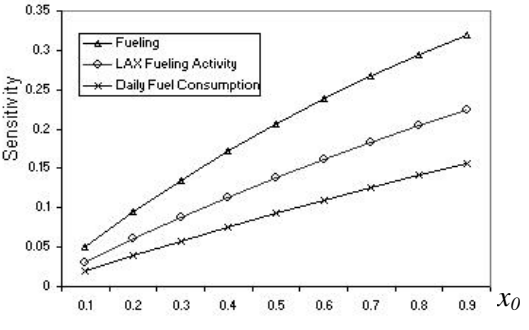


Fig. 15.15(b). The sensitivity of corresponding attribute nodes in (a) to the security node at selected initial values x_0.

15.10 Conclusion

In this chapter we present a technique that prevents users from inferring sensitive information from a series of seemingly innocuous queries. Compared to the deterministic inference approach in previous works, we include non-deterministic relations into

inference channels for query-time inference detection . Specifically, we extract possible inference channels from probabilistic data dependency, the database schema and the semantic knowledge and construct a semantic inference model (SIM). The SIM links represent the possible inference channels from any attribute to the set of pre-assigned sensitive attributes. The parameters of attributes in SIM can be computed in polynomial time in terms of the rows and columns of the relational table. The SIM is then instantiated by specific instances and reduced to a semantic inference graph (SIG) for inference violation detection at query time. To reduce computation complexity for inference, the SIG can be mapped into a Bayesian network so that available Bayesian network tools can be used for evaluating the inference probability along the inference channels . Therefore, our proposed approach can be scalable to large systems.

When a user poses a query, the detection system will examine his/her past query log and calculate the probability of inferring sensitive information from answering the posed query. The query request will be denied if it can infer sensitive information with the probability exceeding the pre-specified threshold. We find that the Bayesian network is able to preserve the structure of the inference channels , which is very useful in providing accurate as well as scalable inference violation detection .

In the multiple-user inference environment, the users can share their query answers to collaboratively infer sensitive information . Collaborative inference is related to the collaboration level as well as the inference channels of the user-posed queries. For inference violation detection , we developed a collaborative inference model that combines the collaborators' query log sequences into inference channels to derive the collaborative inference of sensitive information .

Sensitivity analysis of attributes in the Bayesian network can be used to study the sensitivity of the inference channels . Our study reveals that the nodes closer to the security node have stronger inference effect on the security node. Thus sensitivity analysis of these close nodes can assist domain experts to specify the threshold of the security node to ensure its robustness.

User profiles and questionnaire data provide a good starting point for learning collaboration levels among collaborative users . However, gathering such information is complicated by the fact that the information may be incomplete and incorrect. In addition, the accuracy of such information is task-specific and user-community sensitive. We have constructed a test bed on the inference violation detection system to study the collaboration level for multiple collaborative users . Our preliminary study reveals that information source accuracy and communication fidelity play key roles in the collaboration level . Further research in this area is needed.

References

1. Aberer, K., Despotovic, Z.: Managing Trust in a Peer-2-Peer Information System. In: Proceedings of the tenth international conference on Information and knowledge management, Atlanta, Georgia, USA, October 05–10 (2001)
2. Chavira, M., Allen, D., Darwiche, A.: Exploiting Evidence in Probabilistic Inference. In: Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI), pp. 112–119 (2005)

3. Chan, H., Darwiche, A.: A Distance Measure for Bounding Probabilistic Belief Change. In: Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI), pp. 539–545. AAAI Press, Menlo Park (2002)
4. Chan, H., Darwiche, A.: When Do Numbers Really Matter? Journal of Artificial Intelligence Research 17, 265–287 (2002)
5. Chan, H., Darwiche, A.: Reasoning about bayesian network classifiers. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence, pp. 107–115 (2003)
6. Chan, H., Darwiche, A.: Sensitivity analysis in Bayesian networks: From single to multiple parameters. In: Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI), Arlington, Virginia, pp. 67–75. AUAI Press (2004)
7. Cornelli, F., Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., Samarati, P.: Choosing reputable servents in a P2P network. In: Proceedings of the 11th international conference on World Wide Web, Honolulu, Hawaii, USA, May 07–11 (2002)
8. Chavira, M., Darwiche, A.: Compiling bayesian networks with local structure. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1306–1312 (2005)
9. Chen, Y., Chu, W.W.: Database Security Protection via Inference Detection. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) ISI 2006. LNCS, vol. 3975. Springer, Heidelberg (2006)
10. Chu, W.W., Chen, Q., Hwang, A.Y.: Query Answering via Cooperative Data Inference. Journal of Intelligent Information Systems (JIIS) 3(1), 57–87 (1994)
11. Chu, W.W., Yang, H., Chiang, K., Minock, M., Chow, G., Larson, C.: CoBase: A Scalable and Extensible Cooperative Information System. Journal of Intelligence Information Systems (JIIS) 6 (1996)
12. Date, C.J.: An Introduction to Database Systems, 6th edn. Addison-Wesley, Reading (1995)
13. Darwiche, A.: Recursive conditioning. Arificial Intelligence 126(1-2), 5–41 (2001)
14. Darwiche, A.: Class notes for CS262A: Reasoning with Partial Beliefs, UCLA (2003)
15. Duma, C., Shahmehri, N., Caronni, G.: Dynamic trust metrics for peer-to-peer systems. In: Proceedings of the Sixteenth International Workshop on Database and Expert Systems Applications, pp. 776–781 (2005)
16. Dechter, R.: Bucket elimination: A unifying framework for probabilistic inference. In: Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 211–219 (1996)
17. Dechter, R.: Bucket elimination: A unifying framework for reasoning. Artificial Intelligence 113, 41–85 (1999)
18. Delugach, H.S., Hinke, T.H.: Wizard: A Database Inference Analysis and Detection System. IEEE Trans. Knowledge and Data Engeneering 8(1), 56–66 (1996)
19. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning Probabilistic Relational Models. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, August 1999, pp. 1300–1307 (1999)
20. Farkas, C., Jajodia, S.: The Inference Problem: A Survey. SIGKDD Explorations 4(2), 6–11 (2002)
21. Farkas, C., Toland, T.S., Eastman, C.M.: The Inference Problem and Updates in Relational Databases. In: Proceedings of the 15th IFIP WG11.3 Working Conference on Database and Application Security, pp. 181–194 (2001)
22. Garvey, T.D., Lunt, T.F., Quain, X., Stickel, M.: Toward a Tool to Detect and Eliminate Inference Problems in the Design of Multilevel Databases. In: Proceedings of the 6th Annual IFIP WG 11.3 Working Conference on Data and Applications Security (1992)
23. Getoor, L., Taskar, B., Koller, D.: Selectivity Estimation using Probabilistic Relational Models. In: Proceedings of the ACM SIGMOD (Special Interest Group on Management of Data) Conference (2001)

24. Getoor, L., Friedman, N., Koller, D., Pfeffer, A.: Learning Probabilistic Relational Models. In: Dzeroski, S., Lavrac, N. (eds.) Relational Data Mining. Springer, Heidelberg (2001)
25. He, J., Chu, W.W., Liu, Z.: Inferring Privacy Information From Social Networks. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) ISI 2006. LNCS, vol. 3975. Springer, Heidelberg (2006)
26. Heckerman, D., Mamdani, A., Wellman, M.P.: Real-world applications of Bayesian networks. Communications of the ACM 38(3), 24–68 (1995)
27. Heckerman, D.: A Tutorial on Learning with Bayesian Networks. Techinical Report, Microsoft Research (1996)
28. Hinke, T.H., Delugach, H.S.: Aerie: An Inference Modeling and Detection Approach for Databases. In: Proceedings of the 6th Annual IFIP WG 11.3 Working Conference on Data and Applications Security (1992)
29. Hinke, T.H., Delugach, H.S., Wolf, R.: A Framework for Inference-Directed Data Mining. In: Proceedings of the 10th Annual IFIP WG 11.3 Working Conference on Data and Applications Security (1996)
30. Jensen, F.V.: An Introduction to Bayesian Networks. Springer, New York (1996)
31. Jensen, F.V., Lauritzen, S.L., Olesen, K.G.: Bayesian updating in recursive graphical models by local computation. Computational Statistics Quarterly 4, 269–282 (1990)
32. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigentrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th international conference on World Wide Web, Budapest, Hungary, May 20–24 (2003)
33. Kautz, H., Selman, B., Shah, M.: The Hidden Web. AI magazine (1997)
34. Laskey, K.B.: Sensitivity Analysis for Probability Assessments in Bayesian Networks. IEEE Transactions on Systems, Man and Cybernetics 25, 909–909 (1995)
35. Lauritzen, S.L., Spiegelhalter, D.J.: Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems (with Discussion). Journal of the Royal Statistical Society, Series B 50(2), 157–224 (1988)
36. Lee, W., Stolfo, S.J., Chan, P.K., Eskin, E., Fan, W., Miller, M., Hershkop, S., Zhang, J.: Real Time Data Mining-based Intrusion Detection. In: Proceedings of DISCEX II (June 2001)
37. Marti, S., Garcia-Molina, H.: Taxonomy of trust: Categorizing P2P reputation systems. Computer Networks 50(4), 472–484 (2006)
38. Page, L., Brin, S.: The anatomy of a large-scale hypertextual web search engine. In: Proceedings of the Seventh International World-Wide Web Conference, Brisbane, Australia (April 1998)
39. Pearl, J.: Probabilistic Reasoning in Intelligence Systems. Morgan Kaufmann, San Mateo (1988)
40. Pearl, J.: Bayesian Networks, Causal Inference and Knowledge Discovery. UCLA Cognitive Systems Laboratory, Technical Report (R-281), March. Second Moment (March 1, 2001)
41. SamIam, Automated Reasoning Group, UCLA, http://reasoning.cs.ucla.edu/samiam/
42. Shafiq, B., Bertino, E., Ghafoor, A.: Access control management in a distributed environment supporting dynamic collaboration. In: Workshop On Digital Identity Management, Proceedings of the 2005 workshop on Digital identity management (2005)
43. Thuraisingham, B.M., Ford, W., Collins, M., Keeffe, J.O.: Design and Implementa-tion of a Database Inference Controller. Data Knowl. Eng. 11(3), 271 (1993)
44. Toland, T.S., Farkas, C., Eastman, C.M.: Dynamic Disclosure Monitor (D<Superscript>2</Superscript>Mon): An Improved Query Processing Solution. In: The Secure Data Management Workshop (2005)
45. Winsborough, W., Li, N.: Safety in automated trust negotiation. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 147–160 (2004)

46. Xiong, L., Liu, L.: Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE Transactions on Knowledge and Data Engineering 16(7), 843–857 (2004)
47. Yip, R.W., Levitt, K.N.: Data Level Inference Detection in Database Systems. In: PCSFW: Proceedings of the 11th Computer Security Foundations Workshop (1998)
48. Yu, T., Winslett, M.: A Unified Scheme for Resource Protection in Automated Trust Negotiation. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy, May 11–14, 2003, p. 110 (2003)
49. Yu, T., Winslett, M.: Policy migration for sensitive credentials in trust negotiation. In: Proceedings of the 2003 ACM workshop on Privacy in the electronic society, Washington, DC, October 30 (2003)
50. Zhang, G., Chu, W.W., Meng, F., Kong, G.: Query Formulation from High-Level Concepts for Relational Databases. User Interfaces to Data Intensive Systems (UIDIS) 1999, 64–75 (1994)
51. Zhang, N.L., Poole, D.: Exploiting Causal Independence in Bayesian Network Inference. Journal of Artificial Intelligence Research 5, 301–328 (1996)
52. Zhang, N.L., Poole, D.: A simple approach to bayesian network computations. In: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI), pp. 171–178 (1994)

Questions for Discussions

1. Discuss what are the benefits of using probabilistic approach as compare with the deterministic approach for handling database security?
2. Discuss the types of knowledge needed to construct the semantic inference model.
3. Discuss how to acquire the conditional probability table (CPT) for each attribute from the data sources and give an example.
4. Collaboration level can be used as a metric to measure the percentage of useful information transfer from the source to the recipient in a social network. Provide a method (with an example) for determining the collaboration level.
5. Considering the introducing of virtual node in resolving inconsistent belief as shown in Fig. 15.11(c), in addition to collaborator A, suppose user B has another collaborator C who also informs B $X=x$. What will be user B's belief on X based on both A and C's input?
6. A robust threshold is defined as any small change of attribute values will not cause large impact on the security node. Discuss how can sensitivity analysis be used to improve the robust threshold of the security node.