A Method of Deciding the Security in Publishing Views

Zhang Kun
Hebei University of Science and Technology
Shijiazhuang, Hebei, China
euphkun@163.com

Zhou Wan-zhen
Hebei University of Science and Technology
Shijiazhuang, Hebei, China
lixiaodong_78@163.com

Abstract: As the increasing of data exchange technology, the security in publishing views has been paid more attention to. Inspired by the knowledge of probabilistic database, this paper erects a probability model of security deciding in publishing views, and it gives the formalized definition of publishing views, private information, prior knowledge and so on. Then a new deciding theorem based on entropy is provided, and introduce techniques for measuring the magnitude of information disclosures in publishing view.

Key words: Publishing Views; Probabilistic Database; Information Dependency; Entropy

I. INTRODUCTION

With the development of database technology, data exchange frequency and quantity increase continually, the problem of information disclosure is outstanding day by day in the view publishing process, so guaranteeing security of published view becomes a new subject of database security. As enterprises collect and maintain increasing amounts of personal data, individuals are exposed to greater risks of privacy breaches and identity theft. Many recent reports of personal data theft and misappropriation highlight these risks. As a result, many countries have enacted data protection laws requiring enterprises to account for the disclosure of personal data they manage [1]. Hence, modern information systems must be able to track who has disclosed sensitive data and the circumstances of disclosure.

Traditional security mechanisms protect data at the physical level. For example, firewalls and other perimeter mechanisms prevent the release of raw data, as do conventional access controls for file systems and databases. In data exchange, however, such mechanisms are limited since they can only protect the data up to the first authorized recipient. When data is exchanged with multiple partners, information may be unintentionally disclosed, even when all physical protection mechanisms work as intended. As an extreme example, Sweeney proved this [2] when she retrieved the privileged medical data of William Weld, former governor of the state of Massachusetts, by linking information from two publicly available databases, each of which was considered secure in isolation.

The problem of auditing a log of past queries and updates by means of an audit query that represents the leaked data has been studied in [3]. However, given some sensitive data, it is often difficult to formulate a concise audit query with near-perfect recall and precision. Moreover, the tuples in the sensitive table may have undergone a certain amount of arbitrary perturbation. Finally, the number of suspicious queries produced can be very large, necessitating an ordering based on relevance for an auditor’s investigation. In addition Rakesh Agrawal presented an auditing methodology that ranks potential disclosure sources according to their proximity to the leaked records. Given a sensitive table that contains the disclosed data, methodology prioritizes by relevance the past queries to the database that could have potentially been used to produce the sensitive table [4].

Database watermarking [5] has also been proposed to track the disclosure of information. Database fingerprinting can additionally identify the source of a leak by injecting different marks in different released copies of the data. Oracle [7] offers a “finegrained auditing” function where the administrator can specify that read queries should be logged if they access specified tables. However There does not appear to be any automated facility to determine which queries should be audited.

This paper erects a probability model of security
deciding at the logical level. Then provide a new deciding theorem, it can measuring information disclosures.

II. THE MODEL OF SECURITY-DECIDING

We assume a standard relational schema consisting of several relation names R1, R2, ..., each with a set of attribute names. Our research is based on the standard relational schema.

In Probabilistic database, a database instance is regarded as a real world. In the schema, a database instance is denoted by I, and the probabilistic of the database instance is P(I). Let define all database instance as INST={I1, I2, I3, ..., In}. Any tuple ti in the INST can be seen an incident in probabilistics. In the security-deciding model, if these tuples are independent on others, that is P(t1 ∧ t2)=P(t1)× P(t2).

Let’s define V={v1, v2, v3, ..., vn} as publishing view. In the publishing view, vi (i=(1, 2, ..., n)) is a tuple of V, A1, A2, ..., Am is a set of attribute. Let VD={VD1, VD2, ..., VDm} be the finite domain, which includes all values that can occur in any attributes in any of the relations on the publishing view. S={S1, S2, S3, ...} is the private information which is a 2-dimensional table. S represent the tuples in I which make the result of query Q be ‘true’. We call Q private query.

The prior knowledge K is the information that you can’t get it by publishing view V. However, the K is relevant to view V. In the security-deciding model, prior knowledge test which we call K test will be considered. Firstly, The test will check a single tuple. There are two relationship in different tuples, positive and negative relationship[3].

Use Tup(D)=Tup(VD) ∪ Tup(SD) to represent all tuples which pass K test, and we can use all tuples composition in Tup(D) to generate a database instance set INST. If the INST passes K test, we can get the sub(INST) which is the sub set of INST. In the situations that attackers are different, that is the values of K are different, there will be different database instance sets (sub(INST)).

Definition 1. In a limited domain D, if P(t) represents the probability of any tuple in Tup(D) occurs in database instance; sub(INST) is the database instance set which don’t pass the K test, it’s database instance probability is presented by P(I'); Therefore, we have a probability:

\[
P(I) = \prod_{t \in \text{sub(INST)}} P(t) \prod_{t \notin \text{sub(INST)}} (1 - P(t)) + \frac{\sum_{t \in \text{sub(INST)}} P(I')}{k_u}
\]

Definition 2. For every tuples si∈S and vi∈V, define a d-dimensional comparison vector γ = γ(si, vi) such that γj = 1 if the tuples match on the jth attribute and 0 otherwise. If the jth attribute is missing in one of the tuples, let γj = *.

\[
\forall j=1...d: \gamma^j = \begin{cases} 1 & S_j^i = V_j^i \\ 0 & S_j^i \neq V_j^i \\ * & \text{missing } S_j^i \text{ or } V_j^i \end{cases}
\]

Overall, we have |S|×|V| vectors γ(si, vi), one for each pair of tuples. Let Γ denote the |S|×|V|×d matrix of all comparison vectors. We shall define a probabilistic model that describes the distribution of these vectors. The model is centered on the notion of true matching between two tuples. We assume that there is an unknown function Match:

\[S \times V \rightarrow \{M, U\}\]

Here “M” means “tuples match” and “U” means “tuples do not match”.

III. THE THEOREM OF SECURITY-DECIDING

The theory of security deciding in publishing views is enlightened by “query-view security”. The attacker knows prior knowledge K, the domain D, and the probabilistic of database instance P(I) and publishing view V. Attacker’s aim is to infer the private information S by these factors. To realize the security deciding in publishing views, we introduce the concept of entropy. In the information system, information entropy is always used to judge the indefinity among variables.

If private query is denoted by Q, private information is S publishing view is V we can get private information entropy:

\[H(Q) = - \sum_{Q(i)=true} p(I) \log p(I) = - \sum_{\text{Inst} S_i \times S_i = \phi} P(S_i) \log P(S_i)\]
Entropy of publishing view:

\[ H(V) = - \sum_{I \in V} p(I) \log p(I) = - \sum_{I \in V} P(V_I) \log P(V_I) \]

The union entropy of private information and publishing view:

\[ H(Q, V) = - \sum_{S \in S} \sum_{V \in V} p(S, V) \log p(S, V) \]

A. The Decision of Absolute Secure View

The absolute secure view is that the attacker can’t infer any information in private information S, even if he knows publishing view V and prior knowledge K.

Theorem 1. If private query is denoted by Q, publishing view is V. The prior knowledge is K (K is indicated by a 2-dimensional table U). The union entropy of Q, V and U equals the sum of the union entropy Q, U and the union entropy V, U. That is:

\[ H(Q, U, V) = H(Q) + H(V) \]

If attacker gets prior knowledge K, V is an absolute secure view for private query Q.

When a view is published first time, attacker does not have any prior knowledge K, which is 2-dimensional table U is empty. Then we can simplify Theorem 1, and use Theorem 2 to decide secure of publishing view V.

Theorem 2. If private query is denoted by Q, publishing view is V. The union entropy of two variables equals the sum of entropy Q and V. That is:

\[ H(Q, V) = H(Q) + H(V) \]

B. MEASURING DISCLOSURES

In chapter 3, we knew “M” means “tuples match to S and V” and “U” means “tuples do not match to S and V”.

The record linkage process attempts to classify each tuple pair \( s_i, v_i \) as either M or U, by observing comparison vectors \( \gamma(s_i, v_i) \). This clarification is possible because the distribution of \( \gamma(s_i, v_i) \) for M-labeled tuple pairs is very different from its distribution for U-labeled pairs. Let us define two sets of conditional probabilities:

- \( m(\gamma) = P[\gamma(s_i, v_i) | s_i, v_i \in M] \)
- \( u(\gamma) = P[\gamma(s_i, v_i) | s_i, v_i \in U] \)

Assume that the M-label and U-label are themselves independently assigned to each pair, with probability \( p \in [0, 1] \) to assign an M-label and probability \( 1-p \) to assign a U-label. Then, the probability that some unlabeled pair \( s, v \) has a comparison vector \( \gamma \) equals:

\[ P[\gamma(s, v) = \gamma] = p P[\gamma | M] + (1-p)P[\gamma | U] = pm(\gamma) + (1-p)u(\gamma) \]

Relevance computation of S and V. If private query is denoted by Q, publishing view is V. The prior knowledge is K (K is indicated by a 2-dimensional table U). The value of Relevance computation of S and V is \( H_{sec1} \)

\[ H_{sec1} = H(Q, U, V) + H(U) - H(Q, U) - H(V, U) \]

Matching computation of S and V. We shall assume that, all pairs and their comparison vectors \( \gamma \in \Gamma \) with index \( k = 1 \ldots K_b \) are left unlabeled, whereas all \( \gamma_k \) with index \( k = K_b+1 \ldots |S| |V| \) are labeled with U. Now one can use maximum likelihood estimation to search for \( m(\gamma) \) and \( u(\gamma) \) that maximize the probability. This estimation is carried out through the EM algorithm [8, 9]. Before we turn to EM, let us denote by \( z_k \in \{0, 1\} \) a random variable such that \( z_k = 1 \iff \text{Match } s_i, v_i \).

The value of matching computation of S and V is \( H_{sec2} \)

\[ H_{sec2} = \log \frac{m(\gamma)}{u(\gamma)} \]

Our definition of leakage is the following:

\[ \text{Leak} = H_{sec1} + H_{sec2} \]

Here, We call \( H_{sec1} \) security coefficient 1 and \( H_{sec2} \) security coefficient 2. The variable x and y are determined by expert.

IV. EXPERIMENTAL RESULTS

Now we have done some experiments to prove the theory of security deciding in publishing views, and certify
the validity of the above theory. We implemented the experiment as C++ applications and performed experiments on a Windows XP Professional Version 2002 SP 2 workstation with 2.4GHz Intel Xeon dual processors, 2 GB of memory.

The data set that we used is the database of a faculty’s information in a college. We set P(t)=0.5, the probabilities of all database instances are equal.

Now we analyze the experiment. The inputs of the experiment are prior knowledge K, publishing view V, and private query Q. In order to measure the disclosure, we can get “leak” by formula (3), Then “leak” is the value of output.

A faculty’s information in a college is described with the language Datalog: College (name, age, profession), publishing view V(profession):College (name, age, profession); S = (“Wang”, “20”, “tutor”).

N_name, N_age, N_profession indicate the size of attribute of name, age, profession.

\[ n = N_{\text{name}} + N_{\text{age}} + N_{\text{profession}} \]

![Figure 5-1 Variational curve of leak value](image)

Analysis of the experiment: with the increase of variable’s amount, the relevance of publishing view and private information will decrease. So the disclosure of information will also decrease. From table 5-1, we can see that with the increase of the size of publishing attribute, the leak will decline quickly. The reason is that attacker always infers the private information through publishing view, so when the size of N_profession increases, the probability that attacker can get a single tuple will decrease. But with the increase of N_profession, H(V) will decrease. H_{sec1} and H_{sec2} will be close to zero. That is the value of leak will be close to zero, the insecurity of the publishing view will decrease. In the real world, when S is private information, we will still publish the data of profession attribute.

V. CONCLUSION

We have presented a novel definition of security for analyzing. The information disclosure of publishing views and shown several important results. We argue that it is indispensable for developing practical tools for monitoring information disclosure. We have already shown the theory of security deciding in publishing views (Theorem1 and Theorem2). We provide a method of measure leakage. We believe these results may be a basis for logically security of publishing views in the future.

REFERENCES