On the Privacy Protection in Information Sharing

Nan Zhang and Wei Zhao
Department of Computer Science
Texas A&M University
College Station, TX 77843, U.S.A
{nzhang, zhao} @ cs.tamu.edu

ABSTRACT

In distributed database systems, information sharing across different parties is a common application. Almost all information sharing applications face a common threat from adversaries intending to obtain private data from the other parties. In this paper, we address privacy protection in distributed information sharing environments. In particular, we model and analyze privacy intrusion attacks. Most existing work adopts a model of honest-but-curious adversaries. We consider much more malicious and aggressive adversaries which may launch multiple correlated attacks, and propose and evaluate countermeasures. Numerical data show that while simple, these countermeasures can effectively defeat malicious attacks in practical application environments. Game theory is utilized to derive optimal strategies for both defending party and adversary.

1. INTRODUCTION

In this paper, we address issues related to sharing information across autonomous entities, each of which holds a private database. The entities are supposed to follow a distributed protocol to answer queries spanning their databases. This kind of information sharing has a wide range of applications including document sharing, medical databases, online recommendation service, etc [2].

There is an increasing concern regarding the protection of private data in information sharing. In particular, there may be adversaries among the entities which intend to obtain the private data of the other entities. Thus, the information sharing protocol should prevent the private data of an entity from being disclosed to the other entities. Thereby, for each entity, the benefit of information sharing can be enjoyed without privacy disclosure. Privacy preserving data mining is one example of privacy protection in information sharing across distributed databases. It performs data mining tasks across multiple databases without compromising the privacy of each individual database [7, 8, 17, 18, 19, 21, 24, 25, 26].

1.1 Previous Work

A number of studies have been carried out to protect privacy in information sharing. Several protocols have been proposed to address privacy preserving in a wide variety of information sharing applications including intersection [1, 2, 10, 15, 20], equijoin [1, 2], association rule mining [17, 24], classification [8, 18, 19, 25], and statistical analysis [7].

The majority of previous protocols assume that the adversaries do not collaborate with each other and are well disciplined to follow the protocol strictly (i.e., honest-but-curious, also known as semi-honest). Under this assumption, the only attack an adversary can perform is to record the communication and intermediate results and infer private information from them. Based on the honest-but-curious assumption, most existing protocols model the information sharing problem as a variation of the secure multi-party computation problem [13], which has been proven solvable by a general combinatorial-circuit-based protocol proposed by Yao [27] and extended by Goldreich, Micali, and Wigderson [14]. However, the general protocol has a high communication overhead which makes it costly for many practical systems with large databases. Researchers have noted the presence of this issue and reported studies on the design of more specific and efficient protocols [1, 2, 3, 4, 7, 8, 10, 17, 18, 19, 20, 21, 24, 25].

1.2 Our Contribution

Major contributions of this paper can be summarized as follows.

- In this paper, we adopt a model which covers a much broader range of adversaries. In particular, we stress that although the honest-but-curious assumption is reasonable in some cases, it is not sufficient in many practical situations. In our model, adversaries may maliciously manipulate their data to be shared and launch multiple

correlated information sharing requests in order to obtain private data belonging to other parties.

- We design and evaluate countermeasures to defeat the privacy intrusion attacks. Our countermeasures are simple but effective. Numerical data show that our countermeasures can sufficiently prolong the time taken for an adversary to achieve its goal of privacy intrusion in many practical distributed information sharing environments.
- We use game theory to derive the unique Nash equilibrium of the system, which is a state in which both the defending party and the adversary achieve their optimal strategies. Neither party can benefit if it unilaterally changes its strategy. Thus, to benefit their own interests, both parties have to adopt the strategies defined by the unique Nash equilibrium.

Our results are significant since this study is the first to efficiently and effectively protect privacy given the presence of multiple correlated adversaries which are much more powerful and aggressive than honest-but-curious adversaries. Our work is also the first to formally model the problem as a game between a defending party and multiple attacking adversaries, and to successfully obtain the unique Nash equilibrium. Our results can be directly applied to information sharing systems which are under the threat of privacy intrusion attacks.

1.3 Paper Outline

The rest of the paper is organized as follows. We formally define the information sharing system and the participating parties in Section 2. We describe the strategies of adversaries in Section 3, and develop the corresponding countermeasures of defending party in Section 4. In Section 5, we give a game theoretic analysis of the adversary strategies and defending countermeasures. We present the numerical results in Section 6, and use the results to estimate the time taken for the adversaries to compromise the private information of the defending party. We conclude with a summary and extensions of our result in Section 7.

2. MODELS

In this section, we introduce models of information sharing systems. First, we define the information sharing system. We will start with a simple model and extend it to a more general one. Then, we define a classification of parties in information sharing systems, depending on their objectives and roles.

2.1 A Simple System Model

An information sharing system consists of two parties, named P_0 and P_1 respectively. Each party P_i has a private dataset V_i which contains numerous data points. We assume that information sharing is realized in a distributed manner, which does not rely on any trusted third party [16]. Hence, P_0 and P_1 are supposed to follow a pre-designed protocol and communicate via network to realize information sharing. As such, we assume that for each party, there is a local processing module that processes its dataset and exchanges information with (the local processing module of) the other party. The protocol is implemented by the processing of and communication between the local processing modules of the two parties. Figure 1 shows an information sharing system under this framework.

We use $f(V_0, V_1)$ to denote the information sharing function realized by the system. That is, as is shown in Figure 1, $f(V_0, V_1)$ is the output of the system based on the input V_0 and V_1. Examples of information sharing functions include intersection ($V_0 \cap V_1$), equijoin ($V_0 \triangleright\!\triangleleft V_1$), scalar product ($V_0 \cdot V_1$), etc.

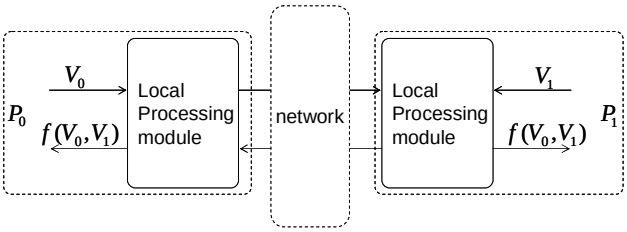

Figure 1. A Simple Information Sharing System

Consider an example system where the information sharing function realized is intersection. Let V_0 and V_1 be $\{0,1,2,3\}$ and $\{2,3,4,5\}$, respectively. We have

$$f(V_0, V_1) = V_0 \cap V_1 = \{2,3\}. \quad (1)$$

We note that $\{0,1\}$ is a subset of V_0 but has no intersection with $V_0 \cap V_1$. Thus, $\{0,1\}$ is private to P_0. Similarly, the private data of P_1 is $\{4,5\}$.

Generally speaking, given a party P_i, if a data point in V_i cannot be inferred from $f(V_0, V_1)$, then the data point is *private* to P_i. Let the set of private data points of P_i be V_i^P. In the intersection problem, we have

$$V_i^P = V_i \setminus f(V_0, V_1). \quad (2)$$

In the rest of this paper, we will focus our investigation on a problem in which the information sharing function performed is intersection. Extensions to other functions will be discussed in Section 7.

In an information sharing system, we define an *information sharing session* as a time interval which starts when a party initializes the information sharing process and ends when the execution of the protocol is completed and both parties obtained the output of the information sharing function. In the rest of the paper, an information sharing session will also be called as a *query*.

2.2 Extended System Model

The above model of information sharing system can be generalized to the case where one party (say, P_0) shares its information with a group of parties, say $\mathbb{P}_1 = \{P_{1j} \mid j = 1, \ldots, k\}$, where P_{1j} has data set V_{1j}. We assume that datasets of parties may vary with time and hence it may be necessary for a member in $\mathbb{P}_1$ to perform multiple queries with P_0. For example, let V_0 be the list of products of an enterprise. If the enterprise updates its product list once every three months, a party P_{1j} may then like to perform a query once every three months to update the shared information.

In an information sharing system, members of $\mathbb{P}_1$ may perform queries asynchronously. Nevertheless, we assume that two or more members in $\mathbb{P}_1$ will not simultaneously perform queries. This assumption simplifies our analysis without loss of generality.

Obviously, our extended model of information sharing systems covers a wide range of distributed information sharing systems.

2.3 Classification of Parties

In this paper, we address issues related to protecting private data of a party. As such, it is necessary to classify parties in an information sharing system into two categories. One category consists of honest parties. An *honest* party would never intentionally intrude the private data of the other party. The other category consists of adversaries. A party is an *adversary* if it intends to obtain the private data of the other party. In this paper, we consider systems where P_0 is always honest while members in $\mathbb{P}_1$ can be either an honest party or an adversary. We call P_0 the *defending* party as it needs to defend itself from potential attacks of adversaries. We refer to the members in $\mathbb{P}_1$ as the *visiting* parties. If a party in $\mathbb{P}_1$ is honest, we say that it is a *legal partner* of P_0.

3. ADVERSARY AND ITS STRATEGIES

Recall that the objective of an adversary is to obtain the private dataset of P_0. In the worst case, an adversary may not have real data to share but may just want to obtain the entire V_0. We study this worst case assumption in this paper.

An adversary can achieve its objective via various attacking techniques. *Single-query* is one kind of attacking techniques. With the single-query technique, the adversary launches one single query to obtain V_0. One example is as follows. Let the entire population of the data points in V_0 (i.e., the set of all possible values that may occur in V_0) be V. Let P_{1j} be an adversary. If P_{1j} sets its input dataset to be $V'_{1j} = V$, then we have

$$f(V_0, V'_{1j}) = f(V_0, V) = V_0 \cap V = V_0. \quad (3)$$

That is, the adversary will successfully obtain the entire set of V_0.

Generally speaking, this kind of single-query attack may not be effective and is easily detectable. In many practical situations, the size of datasets may be several orders less than the size of V. Given this knowledge, P_0 can simply reject a party P_{1j} if $|V'_{1j}| \neq |V|$ for the case in the above example. Thus, adversaries most likely use *multiple-correlated-queries* (MCQ) to achieve their objectives. That is, one or more adversaries may launch multiple correlated queries and aggregate the outputs from these sessions to infer

V_0. In this paper, we deal with adversaries that use MCQ techniques.

The pseudo-code of a typical implementation of MCQ attack technique is given as follows.

$h = 0$;

repeat in each query

$h = h + 1$;

Generate $V'_{1j} \subseteq V$ such that no data point in V'_{1j} has been included in the input of previous queries;

Use V'_{1j} as the input to local processing module;

Receive $W_h = V_0 \cap V'_{1j}$ as the output;

until all values in V has been included in the input of previous queries;

As we can see, the adversaries enumerate the data values in V to collide with the values of the data points in V_0. As we can see, when the adversaries have exhausted all the data values in V, the union of W_i becomes exactly the same as V_0. That is,

$$V_0 = \bigcup_{j=1}^{h} W_j. \quad (4)$$

Thereby, the adversaries achieve their objective of compromising V_0.

4. DEFENDING PARTY AND ITS COUNTERMEASURES

The defending party needs to deploy certain countermeasure in order to prevent its own private data from being obtained by adversaries. We propose a countermeasure consisting of three components: a local processing module which implements a secured exchanged protocol, an adversary detection module, and an input decision module.

The secured exchange protocol, which is implemented in the local processing module, uses encryption technique to guarantee that only the shared information is accessible to either party. The adversary detection module determines if a visiting party is an (potential) adversary and generates an auxiliary signal to the input decision module. Based on the auxiliary signal, the input decision module decides the input of the defending party to its local processing module

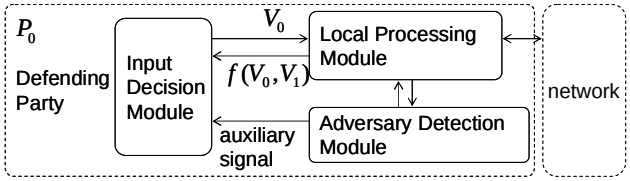

Figure 2. Block Diagram of the Defending Party

Figure 2 shows the block diagram for the defending party with these three components. We will describe algorithms to realize these components. As we will see, these three components integrally reduce the impact of attacks launched by adversaries.

4.1 Secured Exchange Protocol

The secured exchange protocol that we adopt is a variation of an intersection protocol proposed in [2]. Let the visiting party be P_{1j}. It has a dataset of V_{1j} to be shared with P_0, which has a dataset V_0. Let E_0 and E_1 be a pair of commutative encryption functions [6, 13] such that $E_0(E_1(\cdot)) = E_1(E_0(\cdot))$. E_0 is only known to P_0 while E_1 is only known to P_{1j}. By using E_0 to encrypt V_0, the defending party is assured that P_{1j} cannot compute V_0 from $E_0(V_0)$.

The pseudo-code of the secured exchange protocol is given as follows.

1. P_{1j} encrypts its dataset V_{1j} to $E_1(V_{1j})$ by using encryption function E_1; P_{1j} sends $E_1(V_{1j})$ to P_0;

2. P_0 encrypts V_0 and $E_1(V_{1j})$ to $E_0(V_0)$ and $E_0(E_1(V_{1j}))$, respectively by using encryption function E_0; P_0 sends $E_0(V_0)$ to P_{1j}.

3. P_{1j} encrypts $E_0(V_0)$ to $E_1(E_0(V_0))$ by using E_1 and sends $E_1(E_0(V_0))$ to P_0;

4. P_0 computes

$$E_0(E_1(V_{1j})) \cap E_1(E_0(V_0)) = E_0(E_1(V_0 \cap V_{1j})). \quad (5)$$

and sends $E_0(E_1(V_0 \cap V_{1j}))$ to P_{1j};

5. P_{1j} decrypts $E_0(E_1(V_0 \cap V_{1j}))$ to $E_0(V_0 \cap V_{1j})$ and sends $E_0(V_0 \cap V_{1j})$ to P_0;

6. P_0 decrypts $E_0(V_0 \cap V_{1j})$ to $V_0 \cap V_{1j}$ and sends $V_0 \cap V_{1j}$ to P_{1j}.

(To Reviewers: We made a mistake on this protocol in our paper submitted to PODS 2005. This version is correct. We apologize for the confusion.)

It can be shown that this protocol produces countermeasures to prevent an adversary from modifying its local processing module. That is, during the execution of this protocol, only the shared data (i.e., $V_0 \cap V_{1j}$) is accessible to either party even if P_{1j} is an adversary and intends to deviate from the protocol. The proof of this property can be found in Appendix 1. This property guarantees the security for the systems where adversaries do not change their input datasets. Note that adversaries of this kind still have a valid dataset to share. However, at the same time, they are curious about the private data of the defending party.

We stress that this property is necessary but insufficient for the systems we consider. In our systems, adversaries are much more malicious and aggressive. They may not have any valid dataset and are only interested in compromising the private data of the defending party. Thus, we need an additional two components (adversary detection module and input decision module) to defeat the adversaries.

Adversary Detection Module

Recall that the objective of the adversary detection module (ADM) is to determine whether a visiting party is a legal partner or an adversary and send an auxiliary signal to the input decision module.

We propose the following algorithm for this module. Let the input dataset of the visiting party be V_{1j}'.

If $|V_{1j}'| \leq N_S$, ADM identifies the visiting party as an honest party.

If $|V_{1j}'| > N_S$, ADM identifies the visiting party as an honest party with probability $g(|V_{1j}'|)$.

where N_S and $g(\cdot)$ are parameters of the algorithm set by the defending party.

This algorithm is designed based on the following intuition: The size of the population V is usually very large in comparison with the size of the datasets used in information sharing. For example, if two hospitals are to exchange patient records, the size of V, which should contain all the records of potential patients, can be in the order of 10^8 or higher. The real dataset in a particular hospital, however, may have size in the order of 10^3. As such, the adversary may have to forge a large input dataset in information sharing to obtain the private dataset of the defending party in a reasonable amount of time. On the other hand, a legal partner is unlikely to have a large dataset. Thus, the algorithm uses a threshold (N_S) to make the first determination if a visiting party is an adversary.

Input Decision Module

The input decision module uses the auxiliary signal bit from the adversary detection module to determine the input to the local processing module. Let the input to the local processing module be V_0'. There are two possible strategies for the input decision module; one is to *keep honest* and use V_0 as its input (i.e., $V_0' = V_0$). The other is to change the input to $V_0' \neq V_0$. In particular, if the input decision module uses the empty set as the input (i.e., $V_0' = \phi$), we say that the defending party *rejects* the query from the visiting party. In Section 5, we will use game theory to formulate a specific strategy for this module.

Performance Metrics

We now define the performance metrics of the countermeasures of the defending party. The defending party has two goals in the system. One is to obtain the correct result of the information sharing function with legal partners. The other is to prevent privacy leakage. Thus, the performance can be measured by the following two metrics, named error rate and average total-query-interval of adversaries, respectively.

We define the error rate ε as the probability that an information sharing session cannot generate the correct result with a legal partner. To keep the information sharing effective, we assume that the defending party requires a higher bound on the error rate to be ε_0. That is, the defending party requires the information sharing system to have an error rate $\varepsilon \leq \varepsilon_0$.

We define the *average total-query-interval of adversaries*, denoted by T, as the expected value of the total time taken by the adversaries to obtain the complete dataset of the defending party. Let t_i be the time taken to perform the i-th query. Then, T can be calculated as follows.

$$T = E(\sum_{i=1}^{h} t_i), \quad (6)$$

where $E(\cdot)$ refers to the expected value and h is the total number of queries taken by the adversaries to obtain V_0. In the case when t_i is a constant (i.e., $t_i = t$ for all $i = 1, \ldots, h$), T becomes

$$T = E(h \cdot t) = E(h) \cdot t. \quad (7)$$

If we normalize the time such that t equals to one time unit, we have

$$T = E(h). \quad (8)$$

Recall that there are various parameters in the system need to be set. On the side of defending party, N_S, $g(\cdot)$, and V_0' need to be determined. On the side of adversary, the input dataset should be set. Obviously, the selection of these parameters has direct impact on the performance metrics. The defending party would want to choose the parameters so that ε can be minimized and T can be maximized. On the other hand, the adversary certainly wants to minimize T. In the next section, we address how the parameters should be chosen so that the system reaches equilibrium.

5. GAME THEORETIC ANALYSIS

In this section, we formalize parameter-setting strategies for both the defending party and the adversary based on game theory. Specifically, based on the game theoretic formulation, we derive a fixed point where both the defending party and the adversary reach their optimal strategies.

Our derivation is based on the following assumptions. These assumptions are reasonable and have been commonly taken in the literature [5, 13].

ASSUMPTION 1. *Both $|V_0|$ and $|V_{1j}|$ have Poisson distribution with mean N. That is,*

$$\Pr\{|V_0| = i\} = \Pr\{|V_{1j}| = i\} = \frac{N^i e^{-N}}{i!}. \quad (9)$$

ASSUMPTION 2. (**Complete Information**) *Both the defending party and the adversary have full knowledge of all the parameters in the system, which are N_S, $g(\cdot)$, and the distribution of $|V_0|$ and $|V_{1j}|$.*

ASSUMPTION 3. (**No External Knowledge**) *No other information is available to either party.*

For example, the defending party has no external knowledge, other than the auxiliary signal, about whether a visiting party is an adversary or an honest party. The adversary has no external knowledge of the data points in V_0.

5.1 Overview

We can model the information sharing process as a two-party, non-cooperative, complete-information, repeated game [11] between the defending party and the visiting parties. The game is non-cooperative as the defending party does not have a pre-knowledge of whether a visiting party is a legal partner or an adversary.

The defending party has three parameters to determine: N_S, $g(\cdot)$, and its input to the local processing module, denoted by V_0'. The utility function of the defending party is given as follows.

$$u_H(N_S, g(\cdot), V_0') = \begin{cases} 0, & \text{if } \varepsilon \geq \varepsilon_0, \\ T, & \text{otherwise.} \end{cases} \quad (10)$$

Recall that ε_0 is the maximum error rate that can be tolerated by the defending party. Let the input of a visiting party to its local processing module be V_{1j}'. The utility function of an adversary is then given by

$$u_A(|V_{1j}'|) = \frac{1}{T}. \quad (11)$$

Note that the physical meaning of (11) is that the utility of the adversary is proportional to the average amount of private information obtained by the adversary in one query when the adversary obtains V_0.

Since a legal partner always keeps honest and has no intention of privacy intrusion, we do not consider the legal partners in the game.

5.2 Nash Equilibrium

We now derive the fixed point where both the defending party and the adversaries reach their optimal strategies, thereby maximizing their utilities. This fixed point is called Nash equilibrium [11], and is defined as follows.

DEFINITION 1. *Nash equilibrium is a pair of strategies, one for the defending party and the other for the adversary, such that no party can gain more utility by changing its strategy when the other party keeps its strategy unchanged.*

That is, Nash equilibrium represents a situation where no party has an incentive to unilaterally change its strategy. If a unique Nash equilibrium exists in a game with rational [1] players, then the players will always choose the strategies defined in the equilibrium [11].

For our information sharing system, the unique Nash equilibrium is given as follows.

THEOREM 1. *The following strategies form the unique Nash equilibrium of our information sharing system.*

- *The defending party chooses a strategy S_H which includes the maximum N_S that satisfies*

$$\sum_{i=N_S}^{|V|}(1-\frac{N_S}{i})\cdot\frac{N^i e^{-N}}{i!}\le \varepsilon_0, \tag{12}$$

the distribution $g(\cdot)$ that satisfies

$$\forall i\in[N_S+1,|V|],\quad g(i)=\frac{N_S}{i}, \tag{13}$$

and the input dataset V_0' that satisfies

$$V_0'=\begin{cases}V_0, & \text{if the auxiliary input is } 0,\\ \phi, & \text{if the auxiliary input is } 1.\end{cases} \tag{14}$$

- *The adversary chooses a mixed (randomized) strategy S_A which uses an input dataset of size i with probability*

$$\Pr\{|V_{1j}'|=i\}=\begin{cases}0, & \text{if } i<N_S,\\ \dfrac{N^i}{i\cdot i!\cdot\sum_{j=N_S}^{|V|}\frac{N^j}{j\cdot j!}}, & \text{if } i\ge N_S.\end{cases} \tag{15}$$

PROOF. We will prove the theorem in three steps. In the first step, we show that an adversary cannot obtain more utility by unilaterally changing its strategy. In the second step, we show that the defending party cannot obtain more utility by unilaterally changing its strategy. In the third step, we show that the equilibrium is unique in the game.

PROPOSITION 1. *An adversary cannot obtain more utility by unilaterally changing its strategy.*

We first consider the number of queries needed by an adversary to obtain V_0 when the strategies $\langle S_H, S_A\rangle$ are used. In one query, the expected number of data points in V_0 that can be obtained by the adversary is

$$N_S\cdot\frac{N}{|V|}=\frac{N\cdot N_S}{|V|}. \tag{16}$$

Given a certain $|V_{1j}'|$, the adversary will always choose its input dataset V_{1j}' such that no data point in V_{1j}' has been included in the input of previous queries. Thus, the expected number of T is

$$T\approx\frac{|V|}{N_S}. \tag{17}$$

That is, if the strategies $\langle S_H, S_A\rangle$ are used, the expected utility of an adversary is

$$u_A(|V_{1j}'|)=\frac{N_S}{T}. \tag{18}$$

We now consider an adversary that changes its strategy to S_A'. Let the input dataset generated by S_A' be $\tilde{V}_{1j}'$. To be different from V_{1j}', $\tilde{V}_{1j}'$ must satisfy at least one of the following two conditions.

- There exists $i<N_S$ such that $\Pr\{|\tilde{V}_{1j}'|=i\}\neq 0$.
- There exists $i\ge N_S$ such that

$$\Pr\{|\tilde{V}_{1j}'|=i\}\neq\Pr\{|V_{1j}'|=i\}. \tag{19}$$

[1] Here, we say that a player is rational if the player wants to maximize its expected utility and is capable of choosing the strategy that maximizes its expected utility.

If the first condition is satisfied, the expected number of data points in V_0 that can be obtained by the adversary in one query is always less than $N \cdot N_S / |V|$.

If the second condition is satisfied, the expected number of compromised data points remains unchanged because for every $\tilde{V}'_{1j}$ with size $|\tilde{V}'_{1j}| \geq N_S$, the expected size of $|V_0 \cap \tilde{V}'_{1j}|$ is always equal to $N \cdot N_S / |V|$.

Thus, an adversary cannot increase its utility by changing its strategy unilaterally.

PROPOSITION 2. *The defending party cannot obtain more utility by unilaterally changing its strategy.*

Suppose that the defending party changes it strategy to $S_H' : \langle N'_S, g'(\cdot), \tilde{V}'_0 \rangle$. Let $\tilde{g}(i)$ be the probability that a legal partner with an input dataset of size i obtains the correct result of information sharing. The error rate when the defending party uses S_H' satisfies

$$\varepsilon' \geq \sum_{i=N_S}^{|V|} (1 - \tilde{g}(i)) \cdot \frac{N^i \cdot e^{-N}}{i!}. \qquad (20)$$

The expected number of data points in V_0 that can be obtained by the adversary in one query becomes at least

$$\frac{N}{|V|} \sum_{i=N_S}^{|V|} \left(i \cdot \tilde{g}(i) \cdot N^i \Big/ i \cdot i! \cdot \sum_{j=N_S}^{|V|} \frac{N^j}{j \cdot j!} \right). \qquad (21)$$

If the defending party can obtain more utility from S_H', we have

$$\sum_{i=N_S}^{|V|} (1 - \frac{N_S}{i}) \cdot \frac{N^i}{i!} \geq \sum_{i=N_S}^{|V|} (1 - \tilde{g}(i)) \cdot \frac{N^i}{i!}, \qquad (22)$$

and

$$\sum_{i=N_S}^{|V|} N_S \cdot \frac{N^i}{i \cdot i!} > \sum_{i=N_S}^{|V|} i \cdot \tilde{g}(i) \cdot \frac{N^i}{i \cdot i!}. \qquad (23)$$

From (22), we have

$$\sum_{i=N_S}^{|V|} \frac{N_S}{i} \cdot \frac{N^i}{i!} \leq \sum_{i=N_S}^{|V|} \tilde{g}(i) \cdot \frac{N^i}{i!}, \qquad (24)$$

which contradicts (23). Thus, the defending party cannot increase its utility by changing its strategy unilaterally.

PROPOSITION 3. *The equilibrium $\langle S_H, S_A \rangle$ is unique.*

Suppose that there is Nash equilibrium $\langle S'_H, S'_A \rangle$ such that $\langle S'_H, S'_A \rangle \neq \langle S_H, S_A \rangle$. Let the error rate and the average total-query-interval of adversaries when the parties choose $\langle S'_H, S'_A \rangle$ be ε' and T', respectively. Apparently, there is

$$\varepsilon' \leq \varepsilon_0 \qquad (25)$$

because otherwise we have $u_H = 0$. That is, the defending party can simply keep honest to increase its utility. In the following, we consider three cases: $T' > T$, $T' < T$, and $T' = T$, respectively.

Case 1. $T' > T$. Consider an adversary that changes the size of its input dataset to

$$|V_{1j}^0| = \arg \max_{i=N_S}^{|V|} \{ i \cdot \tilde{g}(i) \}, \qquad (26)$$

where $\tilde{g}(i)$ is the probability that a legal partner with input dataset of size i obtains the correct result of information sharing. Let T^0 be the average total-query-interval of adversaries after the adversary changes its strategy. Due to the proof of Proposition 2, if $T^0 > T$, we have $\varepsilon' > \varepsilon_0$, which contradicts (25). If $T^0 \leq T$, the adversary obtains more utility by changing its strategy. Thus, $\langle S'_H, S'_A \rangle$ is not Nash equilibrium.

Case 2. $T' < T$. As we have shown in the proof of Proposition 2, the defending party can always increase its utility by changing its strategy to S_H.

Case 3. $T' = T$. In this case, the set of strategies $\langle S_H, S'_A \rangle$ must also have an average total-query-interval of adversaries equal to T. Apparently, the possibility of this case has been excluded in our proof to Proposition 2. □

6. PERFORMANCE EVALUATION

We now evaluate the system capability for defeating privacy intrusion attacks. From our discussion, it is now clear that it is virtually impossible to completely

eliminate or stop the privacy intrusion attacks defined in this paper. The best measure that a defending party can take is to prolong the time taken for an adversary to achieve its objective of privacy intrusion. As we will show, our countermeasure is effective in the sense that the time taken by an adversary to obtain the private data has become too long to be practical.

Our evaluation is based on the assumption that both defending party and adversary use strategies defined in the Nash equilibrium.

6.1 Numerical Results

In our experiment, we set the size of universal set V to be 10^8 and investigate how the size of V_0 impacts on T when ε_0 is set to 0.1, 0.01, and 0.001, respectively. The numerical results are shown in Figure 3, from which we can make the following observations.

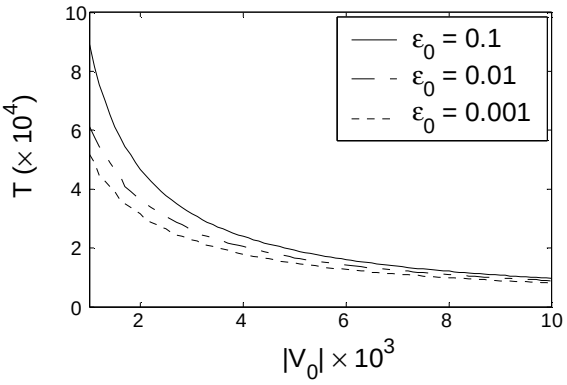

Figure 3. Relationship between T and V_0

- For a given value of $|V_0|$, the lower ε_0 is, the more time (which is proportional to T) the adversaries need to compromise V_0. This coincides with intuition. Recall that ε_0 measures the error in information sharing with legal partners. The defending party pays the price of this error in exchange with prolonging the time taken for the adversaries to compromise V_0.
- T decreases as $|V_0|$ increases as expected. Note that we set $|V|$ to be 10^8 which, we believe, is a reasonable (lower bound of) size of the population of database records. In the figure, $|V_0|$ changes from $1,000$ to $10,000$. Once again, these are the typical range for the database sizes. When $|V_0|$ and $|V|$ are set as in practical systems, we can see that the values of T will be in the range of $15,000$ to $85,000$. If each information sharing session lasts approximately one minute, the time needed by the adversaries to obtain V_0 will be more than 10 days. In many practical systems, this would be too long for the data in V_0 to be meaningful.

7. FINAL REMARKS

We have addressed issues related to privacy protection in information sharing, which has become an important and common application in distributed database systems. Different from most of the existing work which usually adopts a model of honest-but-curious adversaries, we consider much more malicious and aggressive adversaries which may launch attacks with multiple correlated queries in order to obtain private data belonging to other parties. We design countermeasures against such privacy intrusion attacks. Numerical data show that our countermeasures can sufficiently prolong the time taken for an adversary to complete its privacy intrusion mission, such that it becomes impractical for the adversary to achieve its objective. We have formally modeled the problem as a game between a defending party and multiple attacking adversaries and have successfully derived the unique Nash equilibrium of the system. These results provide guidelines for design and configuration of the distributed systems that provide information sharing across multiple parties and are under the threat of multiple correlated query attacks.

The work reported in this paper is preliminary and allows for many possible extensions. One possible extension is to detection an intrusion by analyzing query traffic. For example, if in a short period of time, an abnormal number of queries are from the same IP address or subnet, there is a high probability that these parties are actually forged by an adversary to perform privacy intrusion attacks. This kind of traffic analysis is similar to intrusion detection in denial of service (DOS) attack [9, 12, 22, 23]. Another extension is to extend the information sharing function from intersection to other operations. Our results can be readily applied to information sharing systems with functions equijoin ($V_0 \bowtie V_1$) and scalar product ($V_0 \cdot V_1$). We are currently investigating the privacy

preserving protocols for sum, union, and other information sharing functions.

8. REFERENCES

[1] R. Agrawal, D. Asonov, and R. Srikant, Enabling sovereign information sharing using Web Services, in *Proceedings of the twenty-third ACM SIGMOD international conference on Management of data*, 2004, pp. 873-877.

[2] R. Agrawal, A. Evfimievski, and R. Srikant, Information sharing across private databases, in *Proceedings of the twenty-second ACM SIGMOD international conference on Management of data*, 2003, pp. 86-97.

[3] B. Chor and N. Gilboa, Computationally private information retrieval (extended abstract), in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, 1997, pp. 304-313.

[4] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, Private information retrieval, in *Proceedings of the thirty-sixth Annual Symposium on Foundations of Computer Science (FOCS'95)*, 1995, pp. 41-50.

[5] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, Adversarial classification, in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 99-108, 2004.

[6] W. Diffie and M. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, vol. 22(6), pp. 644-654, 1976.

[7] W. Du, Y. S. Han, and S. Chen, Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification, in *Proceedings of the fourth SIAM International Conference on Data Mining*, 2004, pp. 222-233.

[8] W. Du and Z. Zhan, Building decision tree classifier on private data, in *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining*, 2002, pp. 1-8.

[9] P. Ferguson and D. Senie, Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, RFC 2267, 1998.

[10] M. J. Freedman, K. Nissim, and B. Pinkas, Efficient Private Matching and Set Intersection, in *Advances in Cryptology - Proceedings of Eurocrypt '2004, International Conference on the Theory and Applications of Cryptographic Techniques*. Interlaken, Switzerland, 2004.

[11] D. Fudenberg and J. Tirole, Game Theory, 1991.

[12] T. M. Gil, MULTOPS: a data-structure for denial-of-service attack detection, in *Division of Mathematics and Computer Science*. Netherlands: Vrije Universiteit, 2000, pp. 50.

[13] O. Goldreich, *The Foundations of Cryptography*, vol. 2: Cambridge University Press, 2004.

[14] O. Goldreich, S. Micali, and A. Wigderson, How to play ANY mental game, in *Proceedings of the nineteenth annual ACM conference on Theory of computing*, 1987, pp. 218-229.

[15] B. A. Huberman, M. Franklin, and T. Hogg, Enhancing privacy and trust in electronic communities, in *Proceedings of the first ACM conference on Electronic commerce*, 1999, pp. 78-86.

[16] N. Jefferies, C. Mitchell, and M. Walker, A proposed architecture for trusted third party services, in *Cryptography: Policy and Algorithms Conference, Springer LNCS v1029*, pp. 98-104, 1995.

[17] M. Kantarcioglu and C. Clifton, Privacy-preserving distributed mining of association rules on horizontally partitioned data, in *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, 2004, pp. 1026-1037.

[18] M. Kantarcioglu and J. Vaidya, Privacy Preserving Naive Bayes Classifier for Horizontally Partitioned Data, in *Workshop on Privacy Preserving Data Mining held in association with The third IEEE International Conference on Data Mining*, 2003.

[19] Y. Lindell and B. Pinkas, Privacy Preserving Data Mining, in *Proceedings of the twentieth Annual International Cryptology Conference on Advances in Cryptology*, 2000, pp. 36-54.

[20] M. Naor and B. Pinkas, Oblivious transfer and polynomial evaluation, in *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, 1999, pp. 245-254.

[21] B. Pinkas, Cryptographic techniques for privacy-preserving data mining, *ACM*

SIGKDD Explorations Newsletter, vol. 4(2), pp. 12 - 19, 2002.

[22] T. Ptacek and T. Newsham, Insertion, Evasion, And Denial Of Service: Eluding Network Intrusion Detection, Secure Networks Inc., 1998.

[23] S. Savage, D. Wetherall, A. R. Karlin, and T. Anderson, Practical network support for IP traceback, in *Proceedings of the fifteenth ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*: ACM Press, 2000, pp. 295--306.

[24] J. Vaidya and C. Clifton, Privacy preserving association rule mining in vertically partitioned data, in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 639-644.

[25] J. Vaidya and C. Clifton, Privacy Preserving Naive Bayes Classifier for Vertically Partitioned Data, in *Proceedings of the fourth SIAM Conference on Data Mining*, 2004, pp. 330-334.

[26] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, State-of-the-art in Privacy Preserving Data Mining, *SIGMOD Record*, vol. 33(1), pp. 50-57, 2004.

[27] A. C. Yao, How to generate and exchange secrets, in *Proceedings of twenty-seventh Annual Symposium on Foundations of Computer Science*, 1986, pp. 162-167.

9. APPENDIX

9.1 Proof of the Security of Secured Exchange Protocol

The pseudo-code of the secured exchange protocol is stated as follows.

1. P_{1j} encrypts its dataset V_{1j} to $E_1(V_{1j})$ by using encryption function E_1; P_{1j} sends $E_1(V_{1j})$ to P_0;

2. P_0 encrypts V_0 and $E_1(V_{1j})$ to $E_0(V_0)$ and $E_0(E_1(V_{1j}))$, respectively by using encryption function E_0; P_0 sends $E_0(V_0)$ to P_{1j}.

3. P_{1j} encrypts $E_0(V_0)$ to $E_1(E_0(V_0))$ by using E_1 and sends $E_1(E_0(V_0))$ to P_0;

4. P_0 computes

$$E_0(E_1(V_{1j})) \cap E_1(E_0(V_0)) = E_0(E_1(V_0 \cap V_{1j})). \quad (27)$$

and sends $E_0(E_1(V_0 \cap V_{1j}))$ to P_{1j};

5. P_{1j} decrypts $E_0(E_1(V_0 \cap V_{1j}))$ to $E_0(V_0 \cap V_{1j})$ and sends $E_0(V_0 \cap V_{1j})$ to P_0;

6. P_0 decrypts $E_0(V_0 \cap V_{1j})$ to $V_0 \cap V_{1j}$ and sends $V_0 \cap V_{1j}$ to P_{1j}.

(To Reviewers: We made a mistake on this protocol in our paper submitted to PODS 2005. This version is correct. We apologize for the confusion.)

We will prove that given the input of P_{1j} as V_{1j}, even if P_{1j} deviates from the protocol, P_{1j} cannot learn more information about V_0 than the information in $V_0 \cap V_{1j}$.

PROOF. First, we show that if P_{1j} is semi-honest, the protocol is secure. Then, we consider the case when P_{1j} deviates from the protocol.

Due to the property of commutative encryption functions E_0 and E_1, P_{1j} cannot infer any information about V_0 from $E_0(V_0)$. Since $E_0(E_1(V_0))$ is derived from $E_0(V_0)$, we have

$$V_0 \rightarrow E_0(V_0) \rightarrow E_0(E_1(V_0)). \quad (28)$$

That is, P_{1j} cannot infer any information about V_0 from $E_0(E_1(V_0))$ either. Thus, if P_{1j} is semi-honest, P_{1j} cannot learn more information about V_0 than the information in $V_0 \cap V_{1j}$.

Now we consider the case when P_{1j} deviates from the protocol. P_{1j} is involved in three steps in the protocol: Step 1, Step 3, and Step 5. Without loss of generality, we assume that P_{1j} properly follow the protocol in Step 1. The reason is that if P_{1j} sends $E_1(\tilde{V}_{1j})$

instead of $E_1(V_{1j})$ to P_0, we can always consider the case as P_{1j} chooses another input $\tilde{V}_{1j}$ instead of V_{1j}.

In Step 3, P_{1j} is supposed to send $E_1(E_0(V_0))$ to P_0. Due to the property of E_0 and E_1, P_{1j} cannot increase the expected number of data points in $E_1(E_0(V_0)) \cap E_0(E_1(V_{1j}))$ by changing $E_1(E_0(V_0))$ [13]. Since $E_0(E_1(V_0 \cap V_{1j}))$ is calculated by P_0 in Step 4, even if P_{1j} deviates from the protocol, P_{1j} cannot change the expected number of data points in $E_0(E_1(V_0 \cap V_{1j}))$.

In Step 5, P_{1j} is supposed to decrypt $E_0(E_1(V_0 \cap V_{1j}))$ to $E_0(V_0 \cap V_{1j})$. Note that the number of data points in $E_0(V_0 \cap V_{1j})$ has already been known by P_0 because $|E_0(V_0 \cap V_{1j})| = |E_0(E_1(V_0 \cap V_{1j}))|$. Since P_{1j} cannot derive $V_0 \cap V_{1j}$ from $E_0(V_0 \cap V_{1j})$, P_{1j} cannot learn more information about V_0 by changing $E_0(V_0 \cap V_{1j})$.

Thus, even if P_{1j} deviates from the protocol, P_{1j} cannot learn more information about V_0 than the information in $V_0 \cap V_{1j}$. □