Distributed Privacy Preserving Information Sharing

Nan Zhang Wei Zhao

Department of Computer Science, Texas A&M University
College Station, TX 77843
USA
{nzhang, zhao}@cs.tamu.edu

Abstract

In this paper, we address issues related to sharing information in a distributed system consisting of autonomous entities, each of which holds a private database. Semi-honest behavior has been widely adopted as the model for adversarial threats. However, it substantially underestimates the capability of adversaries in reality. In this paper, we consider a threat space containing more powerful adversaries that includes not only semi-honest but also those malicious adversaries. In particular, we classify malicious adversaries into two widely existing subclasses, called weakly malicious and strongly malicious adversaries, respectively. We define a measure of privacy leakage for information sharing systems and propose protocols that can effectively and efficiently protect privacy against different kinds of malicious adversaries.

1 Introduction

In this paper, we address issues related to sharing information in a distributed system consisting of autonomous entities, each of which holds a private database. The entities are willing to share information across their databases. Nevertheless, no entity is willing to disclose its private data to other entities due to privacy concern. Typical applications of privacy preserving information sharing problem include document sharing, shared medical databases, etc [2].

Various solutions have been proposed to preserve privacy in distributed information sharing systems. In [13], an architecture was proposed to share information using the trusted third party services. However, since the third party has to be trusted by all entities, it may be difficult, if not impossible, for the existence of such an entity in real systems. Therefore, we focus on the information sharing problem in a system without a trusted third party.

In this case, the problem is usually formulated as a variation of the secure multiparty computation (SMC) problem, which has been extensively studied in the literature [10]. Although a general solution to SMC problems has been proven to exist [11, 26], it has a high computational overhead and thus cannot be efficiently used in practice. By making a tradeoff between generality and efficiency, various solutions have been proposed to solve a wide variety of information sharing problems including intersection [1, 2, 9, 12, 18], equijoin [1, 2], association rule mining [14,22], classification [6,15,16,23], top-k queries [24], and statistical analysis [5].

As with many SMC protocols, most solutions share a common assumption that all entities are honest or semi-honest [10]. That is, all entities are well disciplined to follow the protocol properly, with the only exception that an adversary may keep a record of all intermediate computation. This assumption substantially underestimates the capability of adversaries on compromising the privacy of other entities, and thus does not always suffice for real systems.

Some work has been shown to remove the semi-honest assumption (e.g., [9]). However, a weaker assumption is then taken: the size of the database of each entity is known by all entities as pre-knowledge [1]. This assumption tacitly eliminates from consideration a class of adversaries which remove real data from or insert forged data into their databases. Note that by changing (the size of) its database, an adversary can infer considerable private information from the (legitimate) result of information sharing. Thus, this assumption also ignores some privacy intrusion attacks that may be launched by adversaries.

In this paper, we remove the constraints on the behavior of entities. In our system setting, an adversary may deviate from the protocol and/or manipulate its database for the purpose of privacy intrusion. We also allow an honest entity to do the same for certain defensive countermea-

Proceedings of the 31st VLDB Conference,
Trondheim, Norway, 2005

[1] Besides, the work in [9] is based on the client-server model and only requires one entity (client) to know the information sharing result.

sures. Since behavior can no longer be used to differentiate adversaries, we classify honest entities and adversaries by their willingness to share information and/or compromise the privacy of other entities. As in common cases, we assume that all entities are rational in that they make decisions (e.g., attacking methods, defensive countermeasures, etc) based on their intent to optimize their individual benefits.

In this paper, we propose a formal model of adversaries. In particular, we identify two classes of adversaries which widely exist in real systems.

1. The first class of adversaries are those that intend to compromise the privacy of the other entities but will only do so if 1) they will not be convicted as adversaries by the other entities, and 2) the information sharing will still succeed. We refer to this class of adversaries as *weakly malicious adversaries*.

   Semi-honest adversaries are in this class. Nevertheless, an adversary in this class may not necessarily be semi-honest. We note that while semi-honest adversaries are required to follow the protocol properly, weakly malicious adversaries may deviate from the protocol and/or manipulate its database as long as by doing so, the above two conditions are still met.

2. The second class of adversaries are those that will do whatever they can to compromise the privacy of the other entities. In particular, they may even not share any data but manipulate an input database and use it to compromise private information. We refer to this class of adversaries as *strongly malicious adversaries*.

In this paper, we will address both classes of adversaries. Compared with classifying adversaries based on their behavior, our intent-based classification is more tractable for system designers. In real-world applications of information sharing (e.g., sharing information across cooperating companies or government agencies whose objectives are easy to assess), it is relatively easy to identify whether or not an entity has the need to share information. However, it is rather hard to determine if an entity has the capability to change its input database or deviate from the protocol (i.e., if it is appropriate to model the entity as semi-honest).

In this study, we focus on the intersection problem, in which two entities collaborate to share the intersection of their databases. Intersection is one of the most important problems in information sharing. Intersection protocols have been widely used as a primitive in many information sharing applications including classification, association rule mining, etc. Nevertheless, we would like to remark that our goal in this paper is not to design solutions for specific information sharing problems. Rather, we are using the intersection problem as an example to demonstrate our methodology to deal with adversaries without behavior restriction.

For weakly malicious adversaries, we derive a lower bound on the communication complexity of protocols that are capable of preserving privacy without compromising the accuracy of information sharing result. We design a protocol which indeed outperforms this lower bound with the tradeoff of little privacy disclosure. We evaluate the amount of privacy disclosure when our protocol is used and show that the privacy of the defending entity is effectively preserved.

For strongly malicious adversaries, as we will show in this paper, a tradeoff has to be made between privacy protection and accuracy of information sharing result. As such, we propose a game theoretic formulation of the system based on the attacking methods and defensive countermeasures. Based on this formulation, we derive a Nash equilibrium of the game, which is a state in which both the adversary and the defending entity achieve their optimal strategies (i.e., attacking methods or defensive countermeasures). Neither entity can benefit by unitarily changing its strategy. Thus, to benefit their own interests, both entities have to adopt the strategies defined by the Nash equilibrium. We evaluate the performance of defensive countermeasure in this state and show that with an acceptable loss of accuracy, the privacy of the defending entity can be effectively preserved in many systems.

Our results are significant as this is the first effort to remove the restriction on adversary behavior and design simple solutions for information sharing problems by either 1) constraining the adversary goal to be weakly malicious, or 2) allowing a tradeoff between accuracy and privacy.

The rest of the paper is organized as follows: We introduce the system model in Section 2. In Section 3, we introduce our model of adversaries. We present two protocols: Protocol A designed for systems with weakly malicious adversaries, and Protocol B for systems with strongly malicious adversaries, respectively in Section 4. Theoretical analysis on the performance of Protocol A is presented in Section 5. For systems using Protocol B, we propose a game theoretic formulation of the system and derive a Nash equilibrium of the game in Section 6 and Section 7. A numerical performance evaluation of our protocols is provided in Section 8, followed by final remarks in Section 9.

2 System Models

2.1 Parties

Let there be two entities P_0 and P_1 in the system that we refer to as parties. In this paper, unless otherwise indicated, we assume that P_1 intends to compromise the privacy of P_0 while P_0 does not have such intension. Thus, we call P_0 a defending party and P_1 an adversary. Neither party knows if the other party is an adversary.

Each party P_i has a private dataset V_i which contains numerous data. Since the parties are supposed to share the intersection of their datasets, we assume that no data value appears more than once in the same dataset. As is commonly assumed in the literature, each data point in V_i is chosen independently and randomly from a (much larger) set $V = \{v_1, \ldots, v_m\}$. We use p_{ij} to denote the probability that a data point $v_j \in V$ appears in V_i. For the simplicity of discussion, we assume that for all $i \in \{0, 1\}$ and $j \in [1, m]$, there is $p_{ij} = p$. Both parties know V and p. Nevertheless, neither party knows the size or content of the dataset of the other party.

2.2 Problem Statement

In an ideal situation, both parties should obtain $V_0 \cap V_1$ and nothing else at the end of the information sharing process. In reality, this requirement is often relaxed. A common compromise is to allow each party to learn the size of the dataset of the other party after information sharing [2]. As such, we say a system is *secure* if after information sharing, both parties obtain $V_0 \cap V_1$, the size of the dataset of the other party, and nothing else. We define the *privacy* of party P_0 as

$$V_0^{\mathrm{P}} = V_0 \backslash (V_0 \cap V_1) = V_0 \backslash V_1. \qquad (1)$$

Note that when P_1 changes its input dataset to V_1', the privacy of P_0 does *not* change because we define V_0^{P} based on the *real* datasets instead of the *input* datasets. For example, when P_1 is a malicious adversary with no data to share (i.e., $V_1 = \phi$), the privacy of P_0 should always be V_0 no matter what dataset P_1 manipulates to be its input dataset to the information sharing.

The objective of information sharing is to let both parties know $V_0 \cap V_1$ and make V_0^{P} free from unauthorized intrusion by P_1.

2.3 System Infrastructure

There is an information sharing protocol jointly agreed by all parties. We assume that for each party, there is a local processing module that processes the dataset of the party and exchanges information with (the local processing module of) the other party. The information sharing 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. As in common cases, we assume that the defending party will quit the protocol immediately if it can prove that the other party is an adversary.

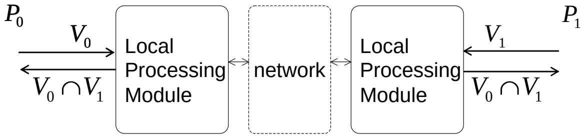

Figure 1: System Infrastructure

Nevertheless, as we mentioned in Section 1, we do not impose any obligatory behavior restriction on either party. We say that a party changes its input dataset if the party manipulates a dataset as the input to its local processing module. We say a party revises its local processing module if the party deviates from the protocol by other means.

2.4 Strategies

In the system, each party needs to choose

- the input from the party to its local processing module,
- the (possibly revised) local processing module of the party.

In addition, an adversary also needs to deliver a dataset $\tilde{V}_0$ that contains all data points that the adversary believes to be in V_0^{P}.

As such, the attacking method of an adversary is to choose a combination of the methods of manipulating its input dataset, modifying local processing module, and generating $\tilde{V}_0$. Since a defending party does not intend to compromise privacy, the defensive countermeasure of a defending party is limited to the former two methods. The attacking methods and defensive countermeasures will be further addressed later in this paper.

2.5 Performance Measurements

Given the attacking method and the defensive countermeasure, we need to measure the accuracy of information sharing result and the amount of privacy disclosure in information sharing.

2.5.1 Accuracy Measurement

Let the attacking method of the adversary and the defensive countermeasure of the defending party be s_{A} and s_{D}, respectively. We propose an *accuracy measure* $l_{\mathrm{a}}(s_{\mathrm{A}}, s_{\mathrm{D}})$ as follows to indicate the success of information sharing.

$$l_{\mathrm{a}}(s_{\mathrm{A}}, s_{\mathrm{D}}) = \begin{cases} 1, & \text{if both parties obtain } V_0 \cap V_1, \\ 0, & \text{otherwise.} \end{cases} \qquad (2)$$

2.5.2 Privacy Measurement

Recall that $\tilde{V}_0$ is the set of data points that the adversary believes to be in V_0^{P} and uses to perform unauthorized intrusion against the defending party. As such, a straightforward measure of privacy disclosure is the number of private data points in $\tilde{V}_0$. Let $\mathrm{Exp}[\cdot]$ be the expected value of a random variable. Since $\tilde{V}_0$ may be randomly generated by the adversary, we formalize this measure as

$$\alpha(s_{\mathrm{A}}, s_{\mathrm{D}}) = \underset{\tilde{V}_0}{\mathrm{Exp}} \left[\frac{|\tilde{V}_0 \cap V_0^{\mathrm{P}}|}{|V_0^{\mathrm{P}}|} \right], \qquad (3)$$

which is the expected percentage of private data points included in $\tilde{V}_0$. This measure is also referred to as *recall* in information retrieval [3]. Readers may raise a question of why we do not measure the maximum number of private data points in $\tilde{V}_0$. We believe that it is not effective to measure such a worst case situation. The reason is as follows: consider an attacking method which randomly generates $\tilde{V}_0$ from V. For any given system, it is always possible for the adversary to generate $\tilde{V}_0 = V_0^{\mathrm{P}}$. As such, the worst case privacy disclosure is always 100% of the private data.

Since the defending party may also change its input dataset, we note that there may also exist data points in $\tilde{V}_0$ which are not in V_0 (i.e., false positives). As such, there is another measure of privacy disclosure

$$\beta(s_{\mathrm{A}}, s_{\mathrm{D}}) = \operatorname*{Exp}_{\tilde{V}_0}\left[\frac{|\tilde{V}_0 \cap V_0^{\mathrm{p}}|}{|\tilde{V}_0|}\right]. \tag{4}$$

This measure is also referred to as *precision* in information retrieval [3]. For the same reason as $\alpha(s_{\mathrm{A}}, s_{\mathrm{D}})$, we measure the expected value instead of the worst case situation. Note that $\beta(s_{\mathrm{A}}, s_{\mathrm{D}})$ is also very important for measuring privacy disclosure because if only $\alpha(s_{\mathrm{A}}, s_{\mathrm{D}})$ is used to measure the privacy disclosure, the maximum privacy disclosure (i.e., $\alpha(s_{\mathrm{A}}, s_{\mathrm{D}}) = 1$) is achieved when the adversary generates $\tilde{V}_0 = V$.

As we can see, the amount of private information obtained by the adversary cannot be determined by either $\alpha(\cdot)$ or $\beta(\cdot)$ unitarily, but can be determined by the combination of them. This results in a problem comparing the amount of privacy disclosure in two cases if one has a larger $\alpha(\cdot)$ while the other one has a larger $\beta(\cdot)$. Such comparison depends on the system setting, as is shown by the following example.

Suppose that the defending party always uses a countermeasure s_{D}. Let s_{A} be an attacking method with $\alpha(s_{\mathrm{A}}, s_{\mathrm{D}}) = 100\%$ and $\beta(s_{\mathrm{A}}, s_{\mathrm{D}}) = 30\%$. Let s'_{A} be an attacking method with $\alpha(s'_{\mathrm{A}}, s_{\mathrm{D}}) = 5\%$ and $\beta(s'_{\mathrm{A}}, s_{\mathrm{D}}) = 100\%$. We will show the comparison between the amount of privacy disclosure when s_{A} and s'_{A} are used in two system settings. First, consider a system where the two parties are two online retailers. The data points in V_i are the telephone numbers of the customers of P_i. The adversary uses the compromised telephone numbers to make unauthorized advertisement to the customers. In this system setting, the adversary prefers s_{A} because a wrong phone call (using $v \in \tilde{V}_0 \backslash V_0^{\mathrm{P}}$) costs the adversary little. As such, s_{A} should have a higher privacy disclosure measure.

We now consider another system where the two parties are two consulting firms. Each data point in V_i is an unpublished profit expectation of a company. The adversary uses the compromised financial data to make investment on a high-risk stock market against the benefit of the defending party. The profit from a successful investment (using $v \in V_0^{\mathrm{P}}$) is huge. Nonetheless, a failed investment (using $v \in \tilde{V}_0 \backslash V_0^{\mathrm{P}}$) costs the adversary five times larger than the profit from a successful investment. In this system setting, the adversary prefers s'_{A} because if s_{A} is used, the expected return from an investment is less than 0 (i.e., the adversary would rather generate $\tilde{V}_0 = \phi$). Thus, s'_{A} should have a higher privacy disclosure measure in this system setting.

As we can see from the above example, we need to introduce the system setting to the measure of privacy disclosure. Let $\delta(v)$ be the profit obtained by the adversary from an unauthorized intrusion based on $v \in V$. Since the adversary intends to compromise the privacy of P_0, we have $\delta(v) > 0$ for all $v \in V_0^{\mathrm{P}}$. Note that there must be $\delta(v) < 0$ for any $v \notin V_0^{\mathrm{P}}$ because otherwise the adversary will always include such v in $\tilde{V}_0$. We define system setting parameter μ as

$$\mu = \left|\frac{\mathrm{Exp}[\delta(v)|v \notin V_0]}{\mathrm{Exp}[\delta(v)|v \in V_0]}\right|. \tag{5}$$

Based on the system setting parameter, we can derive a lower bound on $\beta(s_{\mathrm{A}}, s_{\mathrm{D}})$ to make $\tilde{V}_0$ meaningful for the adversary.

Theorem 2.1. *The profit obtained by the adversary from the unauthorized intrusion is no less than* 0 *if and only if*

$$\beta(s_{\mathrm{A}}, s_{\mathrm{D}}) \geq \frac{\mu}{\mu + 1}. \tag{6}$$

This theorem can be easily proved using our definition of μ. As we can see, when $\beta(s_{\mathrm{A}}, s_{\mathrm{D}}) < \mu/(\mu + 1)$, the return from unauthorized intrusion using $\tilde{V}_0$ is less than 0. Since the adversary is rational, the adversary prefers $\tilde{V}_0 = \phi$ to the $\tilde{V}_0$ generated by s_{A}. When $\tilde{V}_0 = \phi$, the amount of privacy disclosure is 0. As such, we define the *privacy disclosure measure* as follows.

$$l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}}) = \begin{cases} \alpha(s_{\mathrm{A}}, s_{\mathrm{D}}), & \text{if } \beta(s_{\mathrm{A}}, s_{\mathrm{D}}) \geq \mu/(\mu + 1), \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

As we can see, the smaller $l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}})$ is, the less private data is obtained by the adversary and used to perform unauthorized intrusions against the defending party.

We would like to make a few remarks on the relationship between our privacy measure and the security models (e.g., statistically indistinguishable) commonly used in cryptography. The major difference is that while the commonly used security models measure whether the private information is *absolutely* secure against privacy intrusion, we intend to use a continuous value to measure the privacy protection level when absolute security cannot be achieved. As we will show in the paper, when the adversary behavior is not restricted, absolute security can only be achieved with expensive computational cost (for weakly malicious adversaries) or cannot be achieved at all (for strongly malicious adversaries). As such, to design practical solutions against such adversaries, we need to measure the amount of privacy disclosure by a continuous value.

3 Adversary Space

Recall that an adversary wants to compromise the private information of the other party and may or may not want to accomplish the information sharing (i.e., letting both parties know the intersection). Specifically, we assume that the objective of the adversary is to maximize the following objective function.

$$u_{\mathrm{A}}(s_A, s_D) = (1 - \sigma)l_{\mathrm{a}}(s_{\mathrm{A}}, s_{\mathrm{D}}) + \sigma l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}}) \tag{8}$$

where $l_{\mathrm{a}}(\cdot)$ and $l_{\mathrm{p}}(\cdot)$ are defined in (2) and (7), respectively. Note that this model covers a wide range of adversaries. In the case where $\sigma = 1$, the adversary has no interest in accomplishing the information sharing. When $\sigma = 0$, the adversary has no intention to compromise private information and hence becomes a defending party. Generally speaking, the higher σ is, the more desire the adversary has to intrude privacy even at the expense of a failed information sharing. The lower σ is, the more desire the adversary has to share information rather than to compromise the privacy of the other party. In particular, we define two classes of adversaries based on the value of σ as follows.

Definition 1. *An adversary is weakly malicious if and only if the adversary is not semi-honest and has $0 < \sigma < 1/2$. An adversary is strongly malicious if and only if the adversary is not semi-honest and has $1/2 \leq \sigma \leq 1$.*

We now provide an intuitive explanation for our definition of weakly malicious adversaries. Consider the case when the information sharing fails. There is $l_{\mathrm{a}}(s_{\mathrm{A}}, s_{\mathrm{D}}) = 0$. For a weakly malicious adversary, we have

$$u_{\mathrm{A}}(s_A, s_D) = 0 + \sigma l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}}) \leq \sigma < 1 - \sigma. \quad (9)$$

Note that $1 - \sigma$ is a lower bound on $u_{\mathrm{A}}(s_A, s_D)$ when both parties keep honest (i.e., neither revise their local processing modules or change their input datasets). Recall that we assume all parties to be rational in that they make decisions to maximize their objective functions. Thus, when the defending party is honest, a weakly malicious adversary will not intrude privacy if a successful intrusion of privacy will always result in at least one of the following two outcomes: 1) the adversary will be convicted as an adversary by the other party, or 2) at least one party cannot obtain $V_0 \cap V_1$.

With the introduction of weakly and strongly malicious adversaries, we can represent the population of adversaries in a two-dimensional space as is shown in Figure 2. Note that when $\sigma = 0$, the adversary is reduced to a defending party.

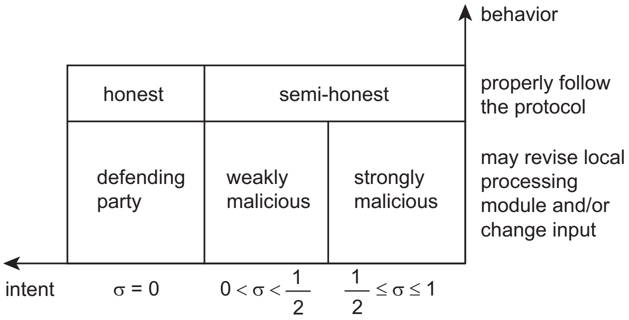

Figure 2: Adversary Space

Given the adversary space, we consider three kinds of systems in this paper.

- Systems with semi-honest adversaries. Parties in these systems are honest or semi-honest.
- Systems with weakly malicious adversaries. Adversaries in these systems are either semi-honest or weakly malicious.
- Systems with strongly malicious adversaries. Adversaries in these systems can be semi-honest, weakly malicious, or strongly malicious.

4 Information Sharing Protocols

We now propose protocols for systems with weakly malicious adversaries and strongly malicious adversaries. We would like to remark that our goal in this paper is not to promote specific protocols, but to demonstrate that when the adversary behavior is not restricted, simple solutions for information sharing problems still exist if we 1) constrain the adversary goal to be weakly malicious, or 2) make a tradeoff between accuracy and privacy.

4.1 Design Goals of Protocols

Before presenting our protocols, we first show that absolute accuracy and security (which we will explain below) can be achieved if the adversaries are weakly malicious or semi-honest, but cannot be achieved if strongly malicious adversaries exist in the system.

When only weakly malicious adversaries exist in the system, there exist protocols which are strictly secure without loss of accuracy of information sharing result. Consider a protocol in which for each pair of data points in the two input datasets (i.e., $\forall [v_0, v_1]$ such that $v_0 \in V_0$ and $v_1 \in V_1$), the two parties call a protocol for Yao's millionaire problem [25] as a subroutine to determine if the data points are equal. If the protocol for Yao's millionaire problem is secure against malicious adversaries, the intersection protocol is secure against weakly malicious adversaries. Basically, the reason is that if only the adversary successfully compromises a private data point of the defending party, the information sharing result obtained by the defending party will always be wrong. As such, a weakly malicious adversary will choose a strategy to keep honest. As we can see, the protocol satisfies the following two conditions:

- (absolute accuracy) The optimal defensive countermeasure for the defending party is to keep honest. Thus, when both parties are defending parties, the information sharing always succeeds.
- (absolute security) After information sharing, the weakly malicious adversary P_1 obtains $V_0 \cap V_1$, $|V_0|$, and nothing else.

Given the presence of such protocol, the objective of a protocol designed for systems with weakly malicious adversaries is to protect privacy without loss of accuracy of information sharing result.

For strongly malicious adversaries, such a protocol does not exist. Consider a strongly malicious adversary with $\sigma = 1$. A possible (though not necessarily optimal) attacking method for the adversary is not to revise its local processing module, but always to insert one data point $v \notin V_1$ into its input dataset. Since the defending party does not know the exact size of V_1, either the malicious adversary compromises v when $v \in V_0$, or another honest party cannot obtain the correct information sharing result when it happens to have a dataset equal to $V_1 \cup \{v\}$. As such, tradeoff has to be made between privacy protection and accuracy of information sharing result. Thus, the goal for designing a protocol for systems with strongly malicious adversaries is to achieve an optimal tradeoff between privacy protection and accuracy of information sharing result.

4.2 Protocols

We now present our protocols designed for systems with weakly malicious adversaries and strongly malicious adversaries, respectively. In both protocols, we use commutative encryption functions [7,21] $E_0(\cdot)$ and $E_1(\cdot)$ on $v \in V$ that satisfy the following properties.

1. E_i is computable in polynomial time. Given E_i, there exists a corresponding decryption function $D_i(\cdot) = E_i^{-1}(\cdot)$ which is also computable in polynomial time.
2. E_0 and E_1 have the same value range. Suppose that c is chosen uniformly at random from the value range of $E_i(\cdot)$. For any $v, v' \in V$ which satisfies $v \neq v'$, no polynomial time algorithm $\mathcal{A}$ with time complexity $O(k)$ can generate output in $\{0,1\}$ such that

   $$\left|\Pr\{\mathcal{A}(v, E_i(v), v', E_i(v')) = \mathcal{A}(v, E_i(v), v', c)\} - \frac{1}{2}\right| > \frac{1}{poly(k)}, \tag{10}$$

   where $poly(\cdot)$ is a polynomial function. Using the terms in cryptography, we say that c and $E_i(v')$ is computationally indistinguishable given v, $E_i(v)$, and v'.
3. $E_0(E_1(\cdot)) = E_1(E_0(\cdot))$.

An example of commutative encryption function is Pohlig-Hellman exponentiation cipher [19],

$$E_i(v) = (h(v))^{e_i} \mod p, \tag{11}$$

with the corresponding decryption function

$$D_i(c) = c^{d_i} \mod p, \tag{12}$$

where p is a prime number, e_i and d_i are keys that satisfy $e_i d_i \equiv 1 \mod (p-1)$, and h is a strong-collision-resistant hash function from V to all quadratic residues modulo p.

For a dataset V_i and encryption function E_i, we define $E_i(V_i)$ to be the set of $E_i(v|v \in V_i)$, which is represented by a sequence of all $E_i(v|v \in V_i)$ with lexicographical order. Given Property 3 of E_i, we have

$$\begin{aligned} & E_1(E_0(V_0)) \cap E_0(E_1(V_1)) \\ =& E_0(E_1(V_0)) \cap E_0(E_1(V_1)) \quad (13)\\ =& E_0(E_1(V_0 \cap V_1)). \quad (14) \end{aligned}$$

Since a party may change its input dataset, to avoid confusion, we use $\langle |V_i'|, V_i' \rangle$ to denote the input from P_i to its local processing module. If a party P_i detects an inconsistency between the two input from the other party (thereby convicts the other party as an adversary), the local processing module of P_i terminates execution immediately and exits the information sharing process.

```
1: Secretly exchange input dataset size $|V_0'|$ and $|V_1'|$,
2: If $|V_0'| > |V_1'|$, $P_0$ becomes $P_s$ and $P_1$ becomes $P_c$,
   and vice versa. If $|V_0'| = |V_1'|$, $P_0$ and $P_1$ are assigned
   as $P_s$ and $P_c$ randomly.
3: $P_c$ sends $E_c(V_c')$ to $P_s$,
4: $P_s$ sends $E_s(E_c(V_c'))$ to $P_c$ using the order of $E_c(V_c')$,
5: $P_s$ sends $E_s(V_s')$ to $P_c$,
6: $P_c$ computes $E_c(E_s(V_0' \cap V_1'))$. Since $E_s(E_c(V_c'))$ re-
   ceived by $P_c$ in Step 4 is in the same order as $E_c(V_c')$
   generated by $P_c$ in Step 3, $P_c$ can thereby finds the
   correspondingly $V_0' \cap V_1'$. $P_c$ then sends $V_0' \cap V_1'$ to $P_s$.
```

Figure 3: Protocol A: Designed for Systems with Weakly Malicious Adversaries

```
1: Secretly exchange input dataset size $|V_0'|$ and $|V_1'|$,
2: Exchange encrypted input dataset $E_0(V_0')$ and $E_1(V_1')$,
3: Encrypt the received message and secretly exchange
   $E_0(E_1(V_1'))$ and $E_1(E_0(V_0'))$,
4: Each party now obtains $E_0(E_1(V_0' \cap V_1'))$ and decrypts
   it. Both parties exchange $E_1(V_0' \cap V_1')$ and $E_0(V_0' \cap V_1')$.
```

Figure 4: Protocol B: Designed for Systems with Strongly Malicious Adversaries

Figure 3 and Figure 4 show the pseudo-code for our Protocol A and Protocol B, which are designed for systems with weakly malicious adversaries and strongly malicious adversaries, respectively. In both protocols, we use a simultaneous secret exchange primitive which exchanges two secret messages from two (possibly malicious) parties such that either both parties know the secret of the other party, or no party can know the secret of the other party. This primitive has been realized by many protocols [4,8,17,20].

5 Analysis of Protocol A

We first show that Protocol A is secure when both parties are honest or semi-honest.

Theorem 5.1. *When Protocol A is used, if both parties are honest or semi-honest, each party learns $V_0 \cap V_1$, the size of the dataset of the other party, and nothing else after information sharing.*

Proof. (sketch) Since all parties follow the protocol strictly without changing their input datasets, we have $V_i' = V_i$.

In the protocol, P_s receives $|V_c|$, $E_c(V_c)$, and $V_0 \cap V_1$, P_c receives $|V_s|$, $E_s(E_c(V_c))$ and $E_s(V_s)$. We will prove that the view of either party in the protocol (the information it receives from the other party) is computationally indistinguishable from a view generated from its own dataset, $V_0 \cap V_1$ and the size of the dataset of the other party.

Let C be a sequence of $|V_c|$ lexicographically-ordered random variables chosen uniformly from the value range of $E_i(\cdot)$. We can construct a view $\langle |V_c|, C, V_0 \cap V_1 \rangle$ based on $|V_c|$ and $V_0 \cap V_1$. Due to property 2 of $E_i(\cdot)$, $\langle |V_c|, C, V_0 \cap V_1 \rangle$ and $\langle |V_c|, E_c(V_c), V_0 \cap V_1 \rangle$ are computationally indistinguishable. Thus, P_s learns $V_0 \cap V_1$, $|V_c|$, and nothing else after information sharing.

We now construct a view to simulate the view of P_c. Let C_s be a set of $(|V_s| - |V_0 \cap V_1|)$ data points chosen uniformly from $V \backslash V_c$. Let E'_s be a commutative encryption function (whose key is) randomly generated such that E_c and E'_s also satisfy the three properties as E_0 and E_1. We construct a view $\langle |V_s|, E'_s(E_c(V_c)), E'_s((V_0 \cap V_1) \cup C_s) \rangle$ based on $V_0 \cap V_1$, V_c, and $|V_s|$. Due to property 2, the constructed view is computationally indistinguishable to $\langle |V_s|, E_s(E_c(V_c)), E_s(V_s) \rangle$. Thus, P_c learns $V_0 \cap V_1$, $|V_s|$, and nothing else after information sharing. □

We now analyze the cases where weakly malicious adversaries exist in the system. Let s_{D}^0 be a defensive countermeasure which will neither change the input dataset nor revise the local processing module (i.e., to keep honest). We derive an upper bound on the amount of privacy disclosure as follows.

Theorem 5.2. *When the adversary is weakly malicious, let s_{A} be the optimal attacking method for the adversary. When Protocol A is used, there is $l_{\mathrm{a}}(s_{\mathrm{A}}, s_{\mathrm{D}}^0) = 1$ and $l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}}^0) \leq \sqrt{p/|V|}$, where p is the probability that a data point $v \in V$ appears in V_i.*

Proof. (sketch) Since the defending party keeps honest, we have $V'_0 = V_0$. First, we show that the adversary cannot compromise any private information when it becomes P_s in the protocol. As we can see, P_s receives $E_c(V'_c)$ in step 3 and $V'_0 \cap V'_1$ in step 6. The adversary cannot compromise privacy from $E_c(V'_c)$ due to the property of the encryption function $E_c(\cdot)$. We note that if P_1 can infer private information from $V'_0 \cap V'_1$ (i.e., $V_0^{\mathrm{P}} \cap (V'_0 \cap V'_1) \neq \phi$), the information sharing fails because P_0 does not obtain the correct intersection. Following the definition of weakly malicious adversary, P_1 would prefer keeping honest. Thus, the adversary cannot compromise any private information when it becomes P_s.

We now show that the adversary can only compromise private information in $((V'_0 \backslash V_0) \cap V_1)$ when it becomes P_c. In the protocol, P_c sends out $E_c(V'_c)$ in step 3 and $V'_0 \cap V'_1$ in step 6. In order to compromise private information, P_c may perform either one or both of the following two intrusions: 1) changing its input dataset V_c, and 2) deviate from the protocol in step 6. After step 6, P_c does not receive any more information. Thus, the only private information P_c can obtain is $((V'_c \backslash V_c) \cap V_s)$. Note that if $V_c \subseteq V'_c$, P_c can still compute $V_c \cap V_s = V_c \cap (V'_c \cap V_s)$ and send this correct intersection set to P_s in step 6.

Since $|V'_1|$ and V'_1 have to be consistent, the attacking method is to generate V'_1 such that $V_1 \subseteq V'_1$. We now compute $l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}}^0)$. Note that $|V'_1|$ has to be determined before $|V'_0|$ is known by P_1. Thus, the optimal $|V'_1|$ must maximize $\mathrm{Exp}_{V_0}[l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}}^0)]$. We have

$$l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}}^0) = \Pr\{|V'_1| < |V_0|\} \mathrm{Exp}_{V_0} \left[\frac{|(V'_1 \backslash V_1) \cap V_0|}{|V_0|} \right]. \tag{15}$$

With some mathematical manipulation, we have $l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}}^0) \leq \sqrt{p/|V|}$. □

The above theorem indicates that when the defending party keeps honest, the privacy leakage of our protocol is relatively small for weakly malicious adversaries. In practice, $|V|$ can be in the order of 10^9 while $|V_i|$ is in the order of 10^3. In this case, the expected number of data points compromised by the adversary is in the order of $10^{-4.5}$ or less.

Theorem 5.3. *The communication overhead of our protocol is $(|V_0| + |V_1| + \min(|V_0|, |V_1|) + |V_0 \cap V_1| + k) \log(|V|)$, where k is a constant value.*

Compared to that of the most efficient existing protocol which is secure against semi-honest adversaries [2], the overhead of our protocol is only $k \log(|V|)$ more, which occurs in the first step.

We now compare the communication overhead of our protocol with that of the protocols which are both absolutely accurate and absolutely secure against weakly malicious adversaries. A lower bound on the communication overhead of such protocols is derived as follows.

Theorem 5.4. *There does not exist any protocol which satisfies all the following three conditions simultaneously.*

1. *If both parties follow the protocol properly without changing their input datasets, at the end of the execution of the protocol, both parties obtain $V_0 \cap V_1$, the dataset size of the other party, and nothing else,*

2. *The communication overhead of the protocol is less than $2(|V'_0| + |V'_1|) \log(|V|)$,*

3. *For any weakly malicious adversary, there is $l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}}^0) = 0$, where s_{A} is the optimal attacking method for the adversary.*

Please refer to [27] for the proof of this theorem. As we can see, when $|V_0|$ and $|V|$ are large, our protocol has a communication overhead substantially lower than these protocols (by at least $\max(|V_0|, |V_1|) \log(|V|)$) with little privacy disclosure introduced.

6 Analysis of Protocol B

We first show that Protocol B is secure when both parties are honest, semi-honest, or weakly malicious. Since no protocol can achieve both absolute accuracy and absolute security when strongly malicious adversaries exist, we analyze the tradeoff between accuracy and privacy when Protocol B is used in a system with strongly malicious adversaries.

6.1 Systems with Semi-honest and Weakly-malicious Adversaries

Theorem 6.1. *When Protocol B is used, if both parties are honest, semi-honest, or weakly malicious, each party learns $V_0 \cap V_1$, the size of the dataset of the other party and nothing else after information sharing.*

Proof. (sketch) We first consider the case when both parties are honest or semi-honest. In this case, since all parties follow the protocol strictly without changing their input datasets, we have $V_i' = V_i$. The protocol is symmetric in that each party learns exactly the same information about the dataset of the other party. Without loss of generality, we consider the information obtained by P_1. P_1 receives $|V_0|$, $E_0(V_0)$, $E_0(E_1(V_1))$, and $E_1(V_0 \cap V_1)$ after information sharing. In the proof of Theorem 5.1, we proved that the view of $\langle |V_0|, E_0(V_0), E_0(E_1(V_1)) \rangle$ can be simulated by a view constructed from $V_0 \cap V_1$, V_1, and $|V_0|$. As we can see, $E_1(V_0 \cap V_1)$ can also be generated from $V_0 \cap V_1$. Thus, the view of P_1 is computationally indistinguishable to a view constructed from $V_0 \cap V_1$, V_1, and $|V_0|$. As such, when the Protocol B is used, each party learns $V_0 \cap V_1$, the size of the dataset of the other party and nothing else after information sharing.

When weakly malicious adversaries exist in the system, an adversary P_i can only infer private information from the dataset it receives in step 4 (i.e., $E_i(V_0 \cap V_1)$). As we can see, both parties obtain $|E_0(E_1(V_0 \cap V_1))| = |V_0 \cap V_1|$ after step 3. As such, if the adversary can infer private information from $E_i(V_0 \cap V_1)$, it cannot obtain the correct intersection $V_0 \cap V_1$. Thus, when Protocol B is used, the system is secure against weakly malicious adversaries. □

As we demonstrated in Section 4, tradeoff has to be made between accuracy and privacy when strongly malicious adversaries exist in the system. In order to analyze such tradeoff, we propose a game theoretic formulation of the information sharing system as follows.

6.2 Game Theoretic Formulation

To deal with the systems with strongly malicious adversaries, we model the information sharing system as a noncooperative game $G(S_{\mathrm{A}}, S_{\mathrm{D}}, u_{\mathrm{A}}, u_{\mathrm{D}})$ between the two parties where S_{A} and S_{D} are the set of attacking methods and defensive countermeasures, respectively, and u_{A} and u_{D} are the utility functions (i.e., objective functions) for the adversary and the defending party, respectively. The game is non-cooperative as neither party knows whether the other party is an adversary. The utility function for the adversary is the objective function we defined in Section 3. In particular, for a strongly malicious adversary with $\sigma = 1$, we have

$$u_{\mathrm{A}}(s_{\mathrm{A}}, s_{\mathrm{D}}) = l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}}). \tag{16}$$

In order to define the utility function for the defending party, we first need to identify the goals of the defending party. The defending party has two goals in information sharing. One goal is to share information and obtain $V_0 \cap V_1$. We assume that the defending party has to guarantee a success probability of $1 - \epsilon$ for the information sharing if the other party is also a defending party. The other goal is to prevent its private data in V_0 from being compromised by the adversary. As such, we define the utility function for the defending party as

$$u_{\mathrm{D}}(s_{\mathrm{A}}, s_{\mathrm{D}}) = \begin{cases} -\infty, & \text{if } \Pr\{l_{\mathrm{a}}(s_{\mathrm{D}}, s_{\mathrm{D}}) = 0\} > \epsilon, \\ -l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}}), & \text{otherwise}, \end{cases} \tag{17}$$

where $l_{\mathrm{a}}(s_{\mathrm{D}}, s_{\mathrm{D}})$ is the accuracy measure when both parties are defending parties.

Our goal is to derive a Nash equilibrium of the game which contains both the optimal attacking method and the optimal defensive countermeasure. In order to do so, we need to formulate the space of all possible attacking methods and defensive countermeasures. Recall that as we mentioned in Section 2, both attacking methods and defensive countermeasures need to determine the (possible changed) input dataset and the (possibly revised) local processing module. Besides, an attacking method also needs to generate $\tilde{V}_0$ based on the information obtained in information sharing. In this section, we first consider a simple case where both attacking methods and defensive countermeasures do not revise the local processing module. We derive a Nash equilibrium of the game based on this simple case. In the next section, we will prove that neither party can benefit by revising its local processing module. As such, the Nash equilibrium derived in this section will not change when the parties are allowed to revise their local processing modules.

6.3 Simple Attacking Methods and Defensive Countermeasures

6.3.1 Simple Attacking Methods

Due to our classification of adversaries, a strongly malicious adversary has $1/2 \leq \sigma \leq 1$. Nevertheless, we consider the worst cases where the adversary has $\sigma = 1$. That is, the only goal of the adversary is to intrude the privacy of the defending party.

Since Protocol B is secure if the adversary is semi-honest, in order to compromise the privacy of the other party, the adversary must change its input dataset. Since the intersection set may contain data points manipulated by the defending party, the adversary also needs to decide if a data point in $V_0' \cap V_1'$ should be included in $\tilde{V}_0$. We analyze the attacking methods for determining V_1' and $\tilde{V}_0$ respectively as follows.

- Change input dataset. The adversary P_1 can compromise the private information in V_0^{P} by changing its input dataset to V_1'. As we can see, if the defending party keeps honest, the adversary will obtain the private information in $V_0 \cap V_1'$ after information sharing. Due to Protocol B, $|V_1'|$ has to be determined before any information about V_0 can be obtained. Without loss of generality, we assume that $|V_1'|$ is a function of $|V_1|$, denoted by $k_1(|V_1|)$. Due to our system assumption, the adversary has no previous knowledge about any data point in V_0. As such, the optimal method for the adversary to generate V_1' is to choose V_1' randomly from $V \backslash V_1$. Without loss of generality, we model the attacking method on changing the input dataset as to determine $k(|V_1|)$.

- Generate $\tilde{V}_0$ from $V_0' \cap V_1'$. Since neither party may revise its local processing module, the only information that an adversary can obtain from information sharing is $V_0' \cap V_1'$. To benefit its own interest, the adversary has only two methods to generate $\tilde{V}_0$.

  - $\tilde{V}_0 = V_0' \cap V_1'$.
  - $\tilde{V}_0 = \phi$.

  That is, $\tilde{V}_0$ either contains all data points in the intersection set, or none of them. This can be easily observed from the definition of $l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}})$.

6.3.2 Simple Defensive Countermeasures

The defensive countermeasure contains the method of changing the two inputs $\langle |V_0'|, V_0' \rangle$ to the local processing module. Due to the protocol, $|V_0'|$ has to be determined before any information about V_1 can be obtained. Without loss of generality, we assume that $|V_0'|$ is a function of $|V_0|$, denoted by $k_0(|V_0|)$. The only information that P_0 can obtain before choosing V_0' is the size of the input dataset of P_1. As such, we assume that V_0' is a function of V_0 and $|V_1'|$ and is represented by $f(V_0, |V_1'|)$ where $f(V_0, |V_1'|) \subseteq V$ and $|f(V_0, |V_1'|)| = k_0(|V_0|)|$. We model the defensive countermeasure as $\langle k(|V_0|), f(V_0, |V_1'|) \rangle$.

6.4 Theorem of Nash Equilibrium

Let $h = \lfloor |V_0|/\mu \rfloor + 1$. Let V_d be a dataset with the same distribution as V_i. Recall that p is the probability that a data point in V appears in V_i. We derive the Nash equilibrium of the game as follows.

Theorem 6.2. *The optimal defensive countermeasure* $\langle k_0(|V_0|), f(V_0, |V_1'|) \rangle$ *is*

$$k_0(|V_0|) = \begin{cases} |V_0| + h, & \textit{if } h + g(|V_0| + h) \leq \epsilon |V| \\ |V_0|, & \textit{otherwise} \end{cases} \tag{18}$$

$$f(V_0, |V_1'|) = \begin{cases} V_0 \cup \mathcal{U}(V \backslash V_0, h), \textit{if } k = |V_0| + h, \\ V_0, \textit{if } k = |V_0| \textit{ and } |V_1'| < N_S, \\ \mathcal{U}(V_0, N_S \cdot |V_0|/|V_1'|), \textit{otherwise} \end{cases} \tag{19}$$

where $g(\cdot)$ *satisfies*

$$g(i) = \sum_{j=1}^{|V|} \Pr\{|V_d| = j\} \cdot \mathrm{Exp}[|f(V_d, i) \backslash V_d| + |V_d \backslash f(V_d, i)|], \tag{20}$$

$\mathcal{U}(V, j)$ *is the set of* j *data points chosen uniformly at random from* V*, and* N_S *is the largest integer that satisfies*

$$g(|V_0|) + \sum_{j=N_S}^{|V|} \left[\frac{(p|V|)^j e^{-p|V|}}{j!} (1 - \frac{N_S}{k_0(j)}) \cdot |V_0| \right] \leq \epsilon |V|. \tag{21}$$

An optimal attacking method $k_1(|V_1|)$ *is* $k_1(|V_1|) = N_S$*. The above optimal attacking method and defensive countermeasure form a Nash equilibrium of the game.*

Proof. (sketch) We will prove the theorem in three steps. First, we will prove that the error rate does not exceed the upper bound ϵ. Second, we will show that when $k_0(|V_0|) = |V_0| + h$, we have $l_{\mathrm{p}} = 0$. In the last step, we will prove the optimality of the strategy when $k_0(|V_0|) = |V_0|$.

- Error rate is controlled below ϵ.

  We first consider the case when $k_0(|V_0|) = |V_0| + h$. In this case, no matter what $|V_1'|$ is, we have

  $$f(V_0, \cdot) = V_0 \cup \mathcal{U}(V \backslash V_0, h). \tag{22}$$

  If P_1 is a defending party, since $|V| \gg |V_0|$, the error rate is

  $$\begin{aligned} \epsilon_0 &\approx \frac{h + \mathrm{Exp}[|V_1' \backslash V_1| + |V_1 \backslash V_1'|]}{|V|} && (23) \\ &= \frac{h + g(|V_0| + h)}{|V|} && (24) \\ &\leq \epsilon. && (25) \end{aligned}$$

  That is, the error rate is no more than ϵ.

  When $k_0(|V_0|) = |V_0|$, the error rate is

  $$\begin{aligned} \epsilon_0 \approx \frac{1}{|V|}(g(|V_0|) + \sum_{j=N_S}^{|V|} [\Pr\{|V_1| = j\} \cdot && (26) \\ (1 - N_S/k_0(j)) \cdot |V_0|]) && (27) \\ \leq \epsilon. && (28) \end{aligned}$$

  Thus, the error rate is also no more than ϵ.

- When $k_0(|V_0|) = |V_0| + h$, $l_{\mathrm{p}} = 0$.

  When $k_0(|V_0|) = |V_0| + h$, we have $V_0' = V_0 \cup \mathcal{U}(V\backslash V_0, h)$. Recall that $\tilde{V_0}$ is either $V_0' \cap V_1'$ or an empty set. If $\tilde{V_0} = V_0' \cap V_1'$, we have

$$\alpha(s_{\mathrm{A}}, s_{\mathrm{D}}) = \frac{|V_0 \cap V_0'|}{|V_0|} = 1, \quad (29)$$

$$\beta(s_{\mathrm{A}}, s_{\mathrm{D}}) = 1 - \frac{|V_0' \backslash V_0|}{|V_0'|} < \frac{\mu}{\mu + 1}. \quad (30)$$

  As such, we have $u_A < 0$. Thus, the adversary has to choose $\tilde{V_0} = \phi$. That is, we have $l_{\mathrm{p}} = 0$.

- When $k_0(|V_0|) = |V_0|$, the attacking method and the defensive countermeasure form a Nash equilibrium.

  The basic idea of the proof is to show that when the defending party does not change its defensive countermeasure, the adversary cannot compromise any more private information by using a manipulated dataset with size larger than N_S, which can be easily observed from $f(V_0, |V_1'|)$. When the adversary does not change its attacking method, the defending party cannot preserve more private information because otherwise the error rate would be larger than ϵ. As such, the state defined in the theorem is a state where no party can benefit by changing its attacking method or defensive countermeasure unitarily. The detailed proof of this step is mainly mathematical manipulations. Due to space limitations, we omit the detailed proof here.

□

7 Extensions to Complicated Methods

In this section, we will prove that when Protocol B is used, neither the adversary nor the defending party can benefit by revising its local processing module.

7.1 Adversary

Theorem 7.1. *When Protocol B is used, the adversary cannot increase the expected value of its utility function by revising its local processing module.*

Proof. (sketch) First, the adversary will not deviate from the protocol in step 1 and 2 because by doing so, the adversary is actually changing its input dataset. Recall that we assume all parties are rational. As such, the adversary will not revise step 4 either. The reason is that after this step, the adversary cannot obtain any more information about the dataset of the other party. We now show that the adversary will not deviate from the protocol in step 3.

In step 3, the adversary P_1 sends $E_1(E_0(V_0'))$ to the defending party P_0. P_0 then uses $E_1(E_0(V_0'))$ to compute

$$E_1(E_0(V_0')) \cap E_0(E_1(V_1')) = E_0(E_1(V_0' \cap V_1')), \quad (31)$$

which will be decrypted to $E_1(V_0' \cap V_1')$ and sent to P_1 in step 4. Thus, we only need to prove that by changing $E_1(E_0(V_0'))$, the adversary cannot increase

$$\begin{aligned} &|E_1(V_0' \cap V_1')| = |E_0(E_1(V_0' \cap V_1'))| \\ =&|E_1(E_0(V_0')) \cap E_0(E_1(V_1'))|. \end{aligned} \quad (32)$$

Recall that the adversary cannot change $|E_1(E_0(V_0'))|$ because by doing so, the defending party will detect an inconsistency between $|E_1(E_0(V_0'))|$ and $|V_0'|$ and quit the information sharing. As such, we need to prove that the adversary cannot change $E_1(E_0(v_0|v_0 \in V_0'))$ to collide with $E_0(E_1(v_1|v_1 \in V_1'))$. This can be inferred from property 2 of the commutative encryption function. □

7.2 Defending Party

Theorem 7.2. *When Protocol B is used, the defending party cannot increase the expected value of its utility function by revising its local processing module.*

Proof. (sketch) First, the defending party will not deviate from the protocol in step 1 and 2 because it can change its input instead. We remark that the defending party also will not revise step 3 because by doing so, it cannot obtain the information sharing result (i.e., $V_0' \cap V_1'$). As such, we now prove that the defending party will not deviate from the protocol in step 4.

In step 4, the defending party P_0 sends $E_1(V_0' \cap V_1')$ to the adversary P_1. P_1 then decrypts $E_1(V_0' \cap V_1')$ to $V_0' \cap V_1'$, which is the result of information sharing. Since the defending party obtains $|V_1'|$ before step 2, we only need to prove that before step 4, the defending party does not know anything more than $|V_1'|$ about V_1'. If so, the defending party will not revise step 4. Rather, it will change its input in step 2.

As we can see, the defending party has received $E_1(V_1')$ and $E_1(E_0(V_0'))$ since step 2. Thus, we need to prove that given V_0', $E_0(\cdot)$, $|V_1'|$, $E_1(V_1')$, and $E_1(E_0(V_0'))$, there does not exist any polynomial time algorithm with time complexity $O(k)$ and output $v \in V$ such that

$$\left| \Pr\{v \in V_1'\} - \frac{|V_1'|}{|V|} \right| > \frac{1}{poly(k)}, \quad (33)$$

where $poly(\cdot)$ is a polynomial function. This can be inferred from property 2 of the commutative encryption function. □

8 Numerical Results

Numerical measurement has not been commonly used to demonstrate system security because all possible attacking methods cannot be exhausted in a simulation. Nevertheless, we propose to use numerical measurements in our case. The reason is that in the theoretical analysis, we already derive the Nash equilibrium of the game, which is a state where neither party can benefit by unitarily changing

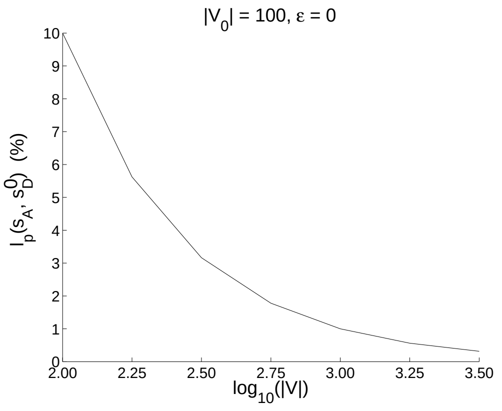

Figure 5: Weakly malicious adversaries

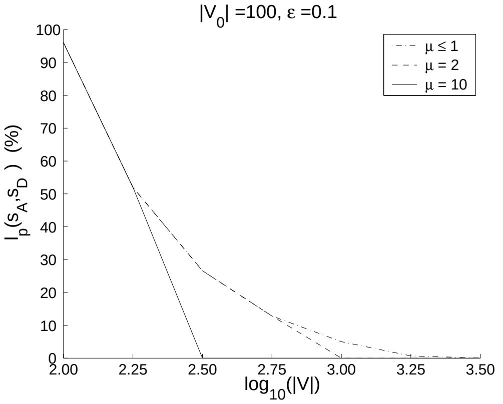

Figure 6: Strongly malicious adversaries

its attacking method or defensive countermeasure. The numerical results shown actually demonstrate the privacy disclosure in this state, and thus can be used to demonstrate the real privacy protection performance of systems using our protocols.

We evaluate the system performance in terms of the maximum expected number of private data compromised by the adversary, which is $l_{\mathrm{p}}(s_{\mathrm{A}}, s_{\mathrm{D}})$, where s_{A} and s_{D} are the optimal attacking strategy and the optimal defensive countermeasure, respectively. The error rate of information sharing when both parties are defending parties is fixed to be $\epsilon = 0$ for systems with weakly malicious adversaries and $\epsilon = 0.1$ for systems with strongly malicious adversaries. With $|V_0| = 100$, we demonstrate the relationship between the amount of privacy disclosure and the size of the population set (i.e., $|V|$).

For systems with weakly malicious adversaries, the maximum amount of privacy disclosure when Protocol A is used is shown in Figure 5. As we can see from the figure, the privacy leakage of our protocol is very small when $|V|$ is large. In particular, when $|V|$ is in the order of 10^3, the expected number of data points compromised by the adversary is no larger than 1.

For systems with strongly malicious adversaries, when Protocol B is used, the maximum amount of privacy disclosure is shown in Figure 6. As we can see from the figure, the higher $|V|$ or μ is, the less private data points are compromised by the adversary. In particular, no privacy disclosure occurs when $\mu \geq 2$ and $|V| \geq 1000$.

9 Conclusion

In this paper, we have addressed issues related to privacy protection in information sharing, which has become an important and common application in distributed systems. Most of the previous studies investigated the problem and proposed solutions based on the assumption that all parties are honest or semi-honest. While it is sometimes useful, this assumption substantially underestimates the capability of adversaries and thus does not always hold in practical situations. We considered a space of more powerful adversaries which include not only honest and semi-honest adversaries but also those who are weakly malicious and strongly malicious. For weakly malicious adversaries, we design an efficient protocol and show that the protocol can preserve privacy effectively. For strongly malicious adversaries, we propose a game theoretic formulation of the system and derive a Nash equilibrium of the game. We evaluate the performance of defensive countermeasure in the Nash equilibrium and show that with an acceptable loss of accuracy, the privacy of the defending entity can be effectively preserved in many systems.

Again, we would like to remark that in this paper, we are not promoting specific protocols. Instead, we show that simple and efficient solutions can be developed to deal with malicious adversaries. Specifically, we show simple solutions can be effective if we 1) constrain the adversary goal to be weakly malicious, or 2) allow making a tradeoff between accuracy and privacy.

Many extensions to our work exist, including 1) extending the information sharing function from intersection to other operations, and 2) dealing with multiple parties in the system, including dealing with correlated attacks from multiple adversaries. Our results can be readily applied to some information sharing functions including 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.

References

[1] R. Agrawal, D. Asonov, and R. Srikant. Enabling sovereign information sharing using web services. In *Proceedings of the 23rd ACM SIGMOD international conference on Management of data*, pages 873–877, 2004.

[2] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *Proceedings of the 22nd ACM SIGMOD international conference on Management of data*, pages 86–97, 2003.

[3] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[4] M. Blum. How to exchange (secret) keys. *ACM Transactions on Computer Systems*, 1(2):175–193, 1983.

[5] 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*, pages 222–233, 2004.

[6] 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*, pages 1–8, 2002.

[7] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[8] S. Even, O. Goldreich, and A. Lempel. A randomizing protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.

[9] M. J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Advances in Cryptography: Proceedings of Eurocrypt 2004*, 2004.

[10] O. Goldreich. *The foundations of cryptography*, volume 2. Cambridge University Press, 2004.

[11] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the 19th annual ACM conference on Theory of computing*, pages 218–229. ACM Press, 1987.

[12] B. A. Huberman, M. Franklin, and T. Hogg. Enhancing privacy and trust in electronic communities. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 78–86, 1999.

[13] N. Jefferies, C. Mitchell, and M. Walker. A proposed architecture for trusted third party services. In *Cryptography Policy and Algorithms Conference*, volume 1029 of *Springer LNCS*, pages 98–104. Springer Verlag, 1995.

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

[15] 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.

[16] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in cryptology*, pages 36–54, 2000.

[17] M. Luby, S. Micali, and C. Rackoff. How to simultaneously ecchange a secret bit by flipping a symmertically-biased coin. In *Proceedings of the 24th Annual Symposium on the Foundations of Computer Science*, pages 11–12, 1983.

[18] M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the 31st annual ACM symposium on Theory of computing*, pages 245–254, 1999.

[19] S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms in gf(p) and its cryptographic significance. *IEEE Transactions on Information Theory*, 24(1):106–111, 1978.

[20] M. O. Rabin. How to exchange secrets by oblivious transfer. Technical Report Technical Memo TR-81, Aiken Computer Laboratory, Harvard University, 1981.

[21] A. Shamir, R. L. Rivest, and L. M. Adleman. Mental poker. In *Mathematical Gardner*, pages 37–43. Wadsworth, Belmont, California, 1981.

[22] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 639–644, 2002.

[23] J. Vaidya and C. Clifton. Privacy preserving naive bayes classifier for vertically partitioned data. In *Proceedings of the 4th SIAM Conference on Data mining*, pages 330–334, 2004.

[24] J. Vaidya and C. Clifton. Privacy-preserving top-k queries. In *Proceedings of the 21st International Conference on Data Engineering*, 2005.

[25] A. C. Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 160–164. IEEE, 1982.

[26] A. C. Yao. How to generate and exchange secrets. In *Proceedings of the 27th Annual Symposium on Foundations of computer science*, pages 162–167. IEEE Press, 1986.

[27] N. Zhang and W. Zhao. Privacy preserving in distributed information sharing systems. Technical Report available at http://people.cs.tamu.edu/nzhang/ppdirs.pdf, Texas A&M University, 2005.