A Game Theoretic Perspective Toward Practical Privacy Preserving Data Mining

Kamalika Das
University of Maryland, Baltimore County
kdas1@cs.umbc.edu

Kun Liu
IBM Almaden Research Center
kun@us.ibm.com

Hillol Kargupta
University of Maryland, Baltimore County & AGNIK
hillol@cs.umbc.edu

Abstract

Analysis of privacy-sensitive data in a multi-party environment often assumes that the parties are well-behaved and they abide by the protocols. Parties compute whatever is needed, communicate correctly following the rules, and do not collude with other parties for exposing third party sensitive data. This paper argues that most of these assumptions fall apart in real-life applications of privacy-preserving distributed data mining (PPDM). It offers a more realistic formulation of the PPDM problem as a multi-party game where each party tries to maximize its own objectives. The paper uses this game-theoretic framework for doing equilibrium-analyses of existing PPDM algorithms. It then modifies these algorithms using the concept of mechanism design and shows how introduction of penalty forces dishonest rational participants to follow the protocol. It illustrates this using the secure sum computation protocol. Finally, this paper discusses the open questions in this work and future research directions.

1. Introduction

Advanced analysis of multi-party privacy-sensitive data plays an important role in many cross-domain applications that require large-scale information integration. The data mining community has responded to this challenge by developing a new breed of distributed data mining algorithms that are privacy preserving. These algorithms attempt to analyze multi-party data for detecting underlying patterns without necessarily divulging the raw privacy-sensitive data to any of the parties. However, most of these privacy preserving data mining algorithms make assumptions regarding the behavior of participating entities, such as, they always follow the protocol and never not try to collude or sabotage the process. These kind of assumptions fall apart in real life. For example, let us consider the US Department of Homeland Security funded PURSUIT project[1] for privacy preserving distributed data integration and analysis of network traffic data from different organizations. The goal here is to detect "macroscopic" patterns from network traffic of different organizations for revealing common threats against those organizations. However, participating entities in a consortium like PURSUIT may not all be ideal. Some may decide to behave like a "leach"—exploit the benefit of the system without contributing much. Some may try to collude with other parties for exposing the private data of a party. In this paper we suggest an alternate perspective for privacy preserving data mining by relaxing some of the existing assumptions. We model large-scale multi-party privacy preserving data mining as a game where each participant (player) tries to maximize its benefit or utility score by optimally choosing the strategies during the execution of the protocol. Modeling using game theory only helps us to analyze the nature of the data mining algorithm. So, to make the participants behave in a desired manner, we use mechanism design to modify existing privacy preserving algorithms to introduce incentive or penalty-based schemes. We show in our analysis that these algorithms work better to achieve the desired goal and are more robust to participant behavior. We illustrate this using the secure sum computation protocol [3]. We modify the standard secure sum algorithm by incorporating an asynchronous distributed penalizing scheme. We analyze the equilibrium states of both versions of the algorithm and show that the semi-honest assumption of the standard secure sum protocol is suboptimal whereas the modified algorithm gives optimal performance.

The remainder of this paper is organized as follows. Section 2 discusses some key concepts of game theory and then reviews applications of game theory in privacy and security. Section 3 describes multi-party PPDM from a game theoretic perspective. Section 4 illustrates the framework using multi-party secure sum computation as an example. Sec-

[1] http://www.agnik.com/DHSSBIR.html

tion 5 gives the optimal solution using a distributed penalty function mechanism. Section 6 presents the experimental results. Finally, Section 7 concludes this paper with a discussion on future directions of research.

2. Background and Related Work

Application of game theory to privacy and security is a relatively new area of research. In this section we review some existing work in this area. Before that we discuss some key concepts in game theory.

A game is an interaction or a series of interactions between players, which assumes that 1) the players pursue well defined objectives (they are *rational*) and 2) they take into account their knowledge or expectations of other players' behavior (they *reason strategically*).

Definition 2.1 (Strategic Game) *A strategic game consists of (i) a finite set P: the set of players, (ii) for each player $i \in P$ a nonempty set A_i: the set of actions available to player i, and (iii) for each player $i \in P$ a preference relation $\succeq_i$ on $A = \times_{j \in P} A_j$: the preference relation of player i.*

The preference relation $\succeq_i$ of player i can be specified by a utility function $u_i : A \rightarrow \mathbb{R}$ (also called a payoff function), in the sense that for any $a \in A, b \in A$, $u_i(a) \geq u_i(b)$ whenever $a \succeq_i b$. The values of such a function is usually referred to as utilities (or payoffs). Here a or b is called the *action profile*, which consists of a set of actions, one for each player. Therefore, the utility (or payoff) of player i depends not only on the action chosen by herself, but also the actions chosen by all the other players. Mathematically, for any action profile $a \in A$, let a_i be the action chosen by player i and a_{-i} be the list of actions chosen by all the other players except i, the utility of player i is $u_i(a) = u_i(\{a_i, a_{-i}\})$.

One of the fundamental concepts in game theory is the Nash equilibrium:

Definition 2.2 (Nash Equilibrium) *A Nash equilibrium of a strategic game is an action profile $a^* \in A$ such that for every player $i \in P$ we have*

$$u_i(\{a_i^*, a_{-i}^*\}) \geq u_i(\{a_i, a_{-i}^*\}) \text{ for all } a_i \in A_i.$$

Therefore, Nash equilibrium defines a set of actions (an action profile) that captures a steady state of the game in which no player can do better by unilaterally changing her action (while all other players do not change their actions).

When the game involves a sequence of interactive actions of the players, and each player can consider her plan of action whenever she has to make a decision, the *strategic game* becomes an *extensive game*. In that situation, the *action* a_i for player i, is replaced by σ_i, the *strategy* for that player, which is a complete algorithm for playing the game, implicitly including all actions of that player for every possible situation throughout the game. The utility function also assigns a payoff to player i for each joint strategies of all the players, *i.e.*, $u_i(\{\sigma_i, \sigma_{-i}\})$.

Halpern and Teague [5] considered the problem of secret sharing and multiparty computation among rational agents. Abraham et al. [1] introduced the k-resilient Nash equilibrium and offered a k-resilient algorithm for solving Shamir's secret sharing [11] problem. Kunreuther et al. [9] and Kearns et al. [8] proposed a practical security problem called the *Interdependent Security (IDS)* in airline companies and proposed a game theory-based solution. Dalvi et al. [4] looked at classification applications as a game between the classifier and the malicious user trying to produce false negatives and developed algorithms for an optimal classifier given the optimal strategies of the attacker. More recently, Agarwal et al. [2] addressed the generalized problem of *honest* information sharing where the idea is to make sure that all the entities get to know only the *correct* result of the query without any additional information. The authors propose a centralized auditing device whose task is to penalize entities if they are caught deviating. Jiang et al. [6] provide an alternative solution to the traditional semi-honest adversary model by proposing an accountable computing framework in which malicious nodes can be detected in polynomial time.

3 Multi-Party PPDM as Games

In a multi-party PPDM environment, each party has certain responsibilities in terms of performing their part of the computations, communicating correct values to other nodes and protecting the privacy of the data. Depending on the characteristics of these entities and their objectives, they either perform their duties or not. Sometimes, they even collude with others to reveal others' private information. Let $\sigma_i = (M_i, R_i, S_i, G_i)$ be the strategy that party (node)i adapts in terms of computation (M_i), communication (receive ((R_i) and send (S_i)), and collusion (G_i)with a group of G_i nodes in the network. Further let $c_{i,m}(M_i)$ be the utility of performing M_i, and similarly we can define $c_{i,r}(R_i)$, $c_{i,s}(S_i)$ and $c_{i,g}(G_i)$. Then the overall utility of node i will be a linear or nonlinear function of utilities obtained by the choice of strategies in the respective dimensions of computation, communication and collusion. Without loss of generality, we consider an utility function which is a weighted linear combination of all of the above dimensions:

$$\begin{aligned} u_i(\{\sigma_i, \sigma_{-i}\}) &= w_{i,m} c_{i,m}(M_i) + w_{i,r} c_{i,r}(R_i) \\ &\quad + w_{i,s} c_{i,s}(S_i) + w_{i,g} c_{i,g}(G_i), \end{aligned}$$

where $w_{i,m}, w_{i,r}, w_{i,s}, w_{i,g}$ represent the how important the specific action is for the node to determine its strategy. In the next section, we would illustrate our formalizations us-

ing one of the most popular PPDM algorithms, the secure sum computation.

4 Case Study: Multi-Party Secure Sum Computation

Secure Sum Computation Suppose there are n individual sites, each with a value $v_j, j = 1, 2, \ldots, n$. It is known that the sum $v = \sum_{j=1}^{n} v_j$ (to be computed) takes an integer value in the range $0, 1, \ldots, N-1$. We want to compute this sum following the secure computation protocol described in [10, 3].

Collusion Analysis The secure sum computation algorithm assumes semi-honest parties who are only interested in the end result and do not indulge in collusion. Since this assumption is not practical, Let us assume that there are k $(k \geq 2)$ nodes acting together secretly to achieve a fraudulent purpose. Let v_i be an honest node who is worried about her privacy. We also use v_i to denote the value in that node. Let v_{i-1} be the immediate predecessor of v_i and v_{i+1} be the immediate successor of v_i. The possible collusion that can arise are:

- If $k = n - 1$, then the exact value of v_i will be disclosed.
- If $k \geq 2$ and the colluding nodes include both v_{i-1} and v_{i+1}, then the exact value of v_i will be disclosed.
- If $n - 1 > k \geq 2$ and the colluding nodes contain neither v_{i-1} nor v_{i+1}, or only one of them, then v_i is disguised by $n - k - 1$ other nodes' values.

The first two cases need no explanation. Now let us investigate the third case. Without loss of generality, we can arrange the nodes in an order such that $v_1 v_2 \ldots v_{n-k-1}$ are the honest sites, v_i is the node whose privacy is at stake and $v_{i+1} \ldots v_{i+k}$ form the colluding group. We have

$$\underbrace{\sum_{j=1}^{n-k-1} v_j}_{\text{denoted by X}} + \underbrace{v_i}_{\text{denoted by Y}} = \underbrace{v - \sum_{j=i+1}^{i+k} v_j}_{\text{denoted by W}},$$

where W is a constant and is known to all the colluding nodes. Now, it is clear that the colluding nodes will know v_i is not greater than W, which is some extra information contributing to the utility of the collusions. To take a further look, the colluding nodes can compute the posteriori probability of v_i and further use that to launch a maximum a posteriori probability (MAP) estimate-based attack. It can be shown that, this posteriori probability is:

$$\begin{aligned} f_{posterior}(v_i) &= \frac{1}{(m+1)^{(n-k-1)}} \times \sum_{j=0}^{r} (-1)^j C_j^{(n-k-1)} \\ &\quad \times C_{(n-k-1)+(r-j)(m+1)+t-1}^{(r-j)(m+1)+t}, \end{aligned}$$

where $v_i \leq W$, $r = \lfloor \frac{W-v_i}{m+1} \rfloor$ and $t = W - v_i - \lfloor \frac{W-v_i}{m+1} \rfloor (m+1)$. When $v_i > W$, $f_{posterior}(v_i) = 0$. Due to space constraints, we have not included the proof of this result here. Interested readers can find a detailed proof in [7].

Note that, when computing this posteriori probability, we model the colluding nodes' belief of each unknown v_j $(j = 1, \ldots, n-k-1)$ as a uniform distribution over an interval $\{0, 1, \ldots, m\}$. This assumption has its roots in the principle of maximum entropy, which models all that is known and assumes nothing about what is unknown, in that case, the only reasonable distribution would be uniform.

Overall Utilities The derived posteriori probability can be used to quantify the utility of collusion, *e.g.*, $g(v_i) = Posteriori - Prior = f_{posterior}(v_i) - \frac{1}{m+1}$. We see here that this utility depends on $W - v_i$ and the size of the colluding group k. Now we can put together the overall utility function for the game of multi-party secure sum computation:

$$\begin{aligned} u_i(\{\sigma_i, \sigma_{-i}\}) &= w_{i,m} c_{i,m}(M_i) + w_{i,r} c_{i,r}(R_i) \\ &\quad + w_{i,s} c_{i,s}(S_i) + w_{i,g} \sum_{j \in P - G_i} g(v_j), \end{aligned}$$

where P is the set of all nodes and G_i is the set of nodes colluding with node i.

Let us now consider a special instance of the overall utility where the node performs all the communication and computation related activities as required by the protocol. This results in a function: $u_i(\{\sigma_i, \sigma_{-i}\}) = w_{i,g} \sum_{j \in P - G_i} g(v_j)$, where the utilities due to communication and computation are constant and hence can be neglected for determining the nature of the function. Figure 1 shows a plot of the overall utility of multi-party secure sum as a function of the distribution of the random variable $W - v_i$ and the size of the colluding group k. It shows that the utility is maximum for a value of k that is greater than 1. Since the strategies opted by the nodes are dominant, the optimal solution corresponds to the Nash equilibrium. This implies that in a realistic scenario for multi-party secure sum computation, nodes will have a tendency to collude. Therefore the non-collusion $(k = 1)$ assumption of the classical secure multi-party sum is sub-optimal. The next section describes a new mechanism that leads to an equilibrium state corresponding to no collusion.

5 Achieving Nash Equilibrium with No Colluding Nodes

To achieve a Nash equilibrium with no collusions, the game players can adopt a punishment strategy to threaten potential deviators. One may design a mechanism to penalize colluding nodes in a number of ways:

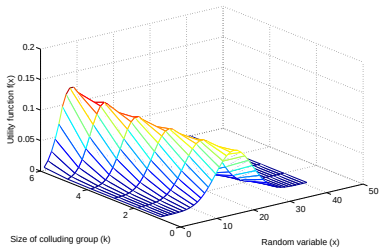

Figure 1. Overall utility for classical secure sum computation. The optimal strategy takes a value of $k > 1$

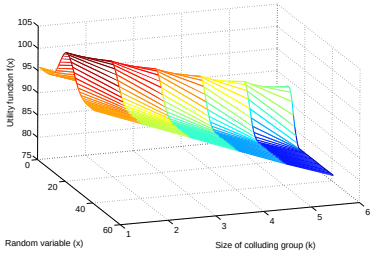

Figure 2. Overall utility for secure sum computation with punishment strategy. The optimal strategy takes a value of $k = 1$.

1. Policy I: Remove the node from the application environment because of protocol violation. Although it may work in some cases, the penalty may be too harsh since usually the goal is to have everyone participate in the process and faithfully contribute to the data mining process.
2. Policy II: Introduce a general penalizing scheme based on one's belief about whether there are violators. This policy does not try to identify violators, but tries to bring down the overall utility of the system, thereby relying on the rational behavior of the players to change for good in the lack of any advantage. Let k' (an estimate of k, actual number of dishonest nodes) be the estimate of threat to the system. Then for policy II, the modified utility function is given by $\tilde{u}_i(\{\sigma_i, \sigma_{-i}\}) = u_i(\{\sigma_i, \sigma_{-i}\}) - * \alpha k'$, where $\alpha > 0$ The last term in the equation accounts for the penalty imposed by the honest nodes. Obviously such a penalizing scheme works for repeated games, where bad nodes turn good in successive rounds of the game.

Figure 2 shows a plot of the modified utility function for secure sum with policy II. It shows that the globally optimal strategies are all for $k = 1$. The strategies that adopt collusion always offer a sub-optimal solutions which would lead to moving the global optimum to the case where $k = 1$.

As an illustrative example, consider a three-party secure sum computation with the payoff listed in Table 1. When there is no penalty, all the scenarios with two bad nodes and one good node offer the highest payoff for the colluding bad nodes. So the Nash equilibrium in the classical secure sum computation is the scenario where the participating nodes

A	B	C	Payoff (No Penalty)	Payoff (Policy I)	Payoff (Policy II)
Good	Good	Good	(3, 3, 3)	(3, 3, 3)	(3, 3, 3)
Good	Good	Bad	(3, 3, 3)	(2, 2, 0)	(2, 2, 2)
Good	Bad	Bad	(3, 4, 4)	(0, 0, 0)	(2, 2, 2)
Bad	Bad	Bad	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)

Table 1. Payoff table for three-party secure sum computation.

are likely to collude. However, in both cases with penalty, no node can gain anything better by deviating from good to bad when all others remain good. Therefore, the equilibrium corresponds to the strategy where none of the nodes collude. Note that, the three-party collusion is not very relevant in secure sum computation since there are all together three parties and there is always a good node (the initiator) who wants to only know the sum.

Implementing the Penalty Mechanism without Having to Detect Collusion: In order to implement the penalizing protocol, one may use a central mediator who can monitor the behavior of all nodes (see, *e.g.*, [2]). However, it requires a a trusted central authority and global synchronization which might create a bottleneck in a distributed system. Instead, an asynchronous distributed control. can be realized by *cheap talk*, a pre-play communication concept from game theory. The idea is based on the assumption that collusion requires consent from multiple parties. So a party with intention of collusion might get caught while sending out collusion invitation randomly in the network if those invitations reach some honest parties. The new protocol will therefore have a pre-play phase where "lobbying agents" (well-behaved nodes or advocacy groups) will make participants aware of the fact that one will be penalized if any collusion is detected. This "lobbying" does not affect the utility function. It simply makes everyone aware of that. *It does not require a perfect collusion detection.* A real threat with an estimated high-enough value of the collusion-size (k') will push everyone toward proper behavior.

Secure Sum with Penalty We propose a secure sum with penalty (SSP) protocol that achieves no collusion. Consider a network of n nodes where a node can either be *good* (honest) or *bad* (colluding). Before the secure sum protocol starts, during cheap talk, the good nodes set their estimate of bad nodes in the network $k' = 0$ and bad nodes send invitations for collusions randomly to nodes in the network. Every time a good node receives such an invitation, it increments its estimate of k'. Bad nodes respond to such collusion invitations and form collusions. If a bad node does not receive any response, it behaves as a good node. To penalize nodes that collude, good nodes split their local data into $\alpha k'$ random shares and in every round they send only one of their $\alpha k'$ shares. Therefore, each sum computation consists of $O(\alpha k')$ rounds of communication for every complete sum computation.

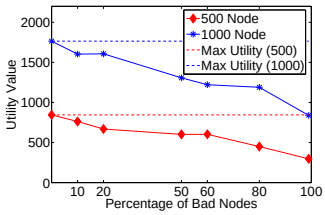

Figure 3. Utility vs. Collusion-size.

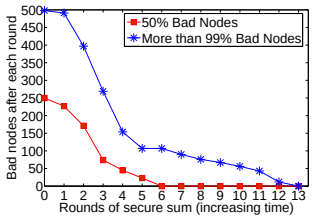

Figure 4. Rate of decrease of bad nodes

6 Experimental Results

We empirically verify our claim that the SSP protocol leads to an equilibrium state where there is no collusion. The utility function used for the experiments is the one described in Policy II. The penalty in this case is the excess amount of communication and computation needed. In the first experiment we demonstrate for different sizes of the network (500 nodes and 1000 nodes) that the utility is maximum when the collusion is minimum (Figure 3). The maximum utility in the figure corresponds to the classical secure sum computation without collusion. In our second experiment we verify that the number of bad nodes decreases with successive rounds of SSP (See Figure 4). Each bad node has a random utility threshold that is assigned during the setup. If the computed utility falls below a node's threshold, the node decides to change its strategy and becomes a good node for the subsequent rounds. The time taken to have a no collusion scenario depends on the initial number of bad nodes in the network, which indirectly decides the estimate of k'.

7 Conclusions

This paper questions some of the common assumptions in multi-party PPDM and shows that if nobody is penalized for cheating, rational participants tends to behave dishonestly. This paper takes a game-theoretic approach to analyze this phenomenon and presents Nash equilibrium analysis of the well-known multi-party secure sum computation. A cheap-talk based mechanism design to implement a penalty is proposed to offer a more robust protocol that does not rely on semi-honest behavior of the participants.

A number of questions, however, yet remain to be answered. The nature of the optimization function reveals that there is an optimum utility of the function for a certain size of the colluding group. A maximization of the objective function to estimate k' might be able to provide interesting results in terms of the convergence time of the algorithm. Also, the current analysis assumes a homogeneous system where all players attach the same importance to the different costs. Studying a heterogeneous scenario would give us a better insight about the performance of the SSP algorithm. In the current version of the SSP algorithm the amount of penalty is a function of the initial estimate of the individual's belief regarding the number of dishonest players in the system. If $k' \ll k$, then the system would take a long time to converge to the Nash equilibrium of honest behavior or might not even converge. If we could provide a numerical computation of how much penalty is needed for the system to converge to eventual honest behavior, then we could provide a bound on the time of convergence for the players of the game.

References

[1] I. Abraham, D. Dolev, R. Gonen, and J. Halpern. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing (PODC)*, Denver, Colorado, USA, July 2006.

[2] R. Agrawal and E. Terzi. On honesty in sovereign information sharing. In *International Conference on Extending Database Technology*, pages 240–256, 2006.

[3] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Zhu. Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations*, 4(2), 2003.

[4] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, New York, NY, USA, 2004. ACM Press.

[5] J. Halpern and V. Teague. Rational secret sharing and multiparty computation: extended abstract. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 623 – 632, Chicago, IL, USA, 2004.

[6] W. Jiang and C. Clifton. Transforming semi-honest protocols to ensure accountability. In *ICDMW '06: Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, pages 524–529, Washington, DC, USA, 2006. IEEE Computer Society.

[7] H. Kargupta, K. Das, and K. Liu. A game theoretic approach toward multi-party privacy-preserving distributed data mining. Technical Report TR-CS-0701, UMBC, April 2007.

[8] M. Kearns and L. Ortiz. Algorithms for interdependent security games. *Advances in Neural Information Processing Systems*, 2004.

[9] H. Kunreuther and G. Heal. Interdependent security. *Journal of Risk and Uncertainty*, 26(2-3):231–249, 2003.

[10] B. Schneier. *Applied Cryptography*. John Wiley & Sons, 2nd edition, 1995.

[11] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.