Privacy-preserving Multiparty Collaborative Mining with Geometric Data Perturbation

Keke Chen, *Member, IEEE,* and Ling Liu, *Senior Member, IEEE*

***Abstract*—In multiparty collaborative data mining, participants contribute their own datasets and hope to collaboratively mine a comprehensive model based on the pooled dataset. How to efficiently mine a quality model without breaching each party's privacy is the major challenge. In this paper, we propose an approach based on geometric data perturbation and data-mining-service oriented framework. The key problem of applying geometric data perturbation in multiparty collaborative mining is to securely unify multiple geometric perturbations that are preferred by different parties, respectively. We have developed three protocols for perturbation unification. Our approach has three unique features compared to the existing approaches. (1) With geometric data perturbation, these protocols can work for many existing popular data mining algorithms, while most of other approaches are only designed for a particular mining algorithm. (2) Both the two major factors: data utility and privacy guarantee are well preserved, compared to other perturbation-based approaches. (3) Two of the three proposed protocols also have great scalability in terms of the number of participants, while many existing cryptographic approaches consider only two or a few more participants. We also study different features of the three protocols and show the advantages of different protocols in experiments.**

***Index Terms*—privacy preserving data mining, distributed computing, collaborative computing, geometric data perturbation**

I. Introduction

Recent advances in computing, communication, and digital storage technologies have enabled incredible volumes of data to be accessible remotely across geographical and administrative boundaries. There is an increasing demand on collaborative mining over the distributed data stores to find the patterns or rules that benefit all of the participants. For example, multiple retailer stores in the same business section want to pool their data together to determine the characteristics of customer purchases. Cancer research institutes in different geographical areas need to collaboratively find the environmental factors related to certain type of cancer. However, these distributed datasets could also contain sensitive information, such as business sales data and patient clinical records. Therefore, an important challenge for distributed collaborative mining is how to protect each participant's sensitive information, while still finding useful data models (classification models, for example).

Keke Chen is with the Department of Computer Science and Engineering, Wright State University, Dayton, OH, 45435, USA e-mail: keke.chen@wright.edu

Ling Liu is with the College of Computing, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: lingliu@cc.gatech.edu

Manuscript received June 15, 2008; revised December 9, 2009.

The service-oriented infrastructure for collaborative mining of data distributed has become the most popular solution [2], [27], where the data providers are the collaborators who submit their own datasets to the designated data mining service provider for discovering and mining the commonly interested models on the pooled data. This model reduces the high communication cost associated with most cryptographic approaches [17], [13]. In this paper, we will study the problem of privacy preserving multiparty collaborative data mining using *geometric data perturbation* under this service-based framework.

Geometric data perturbation has unique benefits [5], [7] for privacy-preserving data mining. First, many popular data mining models are *invariant* to geometric perturbation. For example, the classifiers: kernel methods (including k-nearest-neighbor (KNN) classifier), linear classifiers, and support-vector-machine (SVM) classifiers [11], are invariant to geometric perturbation in the sense that the classifiers trained on the geometrically perturbed data have almost the same accuracy as those mined with the original raw data. This conclusion is also valid for most popular clustering algorithms based on Euclidean distance[14]. Second, multiple geometric data perturbation can be easily generated with low cost, each of which preserves about the same model accuracy. Thus, an individual data provider needs only to select one perturbation that can provide satisfactory privacy guarantee. Comparing with other existing approaches to privacy preserving data mining, geometric data perturbation significantly reduces the complexity in balancing data utility and data privacy guarantee [2], [8].

When applying geometric data perturbation to multi-party collaborative mining, the above advantages are inherited. In addition, due to the service-based framework, the collaboration can scale up conveniently in most cases, while many cryptographic protocols are limited to a small number of parties [17], [13], [26].

The key challenge for applying geometric data perturbation to multiparty collaborative data mining is to securely unify the perturbations used by different data providers, while each party still gets satisfactory privacy guarantee and the utility of the pooled data is well preserved. There are three important factors that impact the quality of unified perturbation: the privacy guarantee of each dataset, the utility of the pooled data, and the efficiency of the perturbation unification protocol. We consider these factors in developing the following three protocols for perturbation unification: the simple protocol, the negotiation protocol, and the space adaptation protocol. Analytical and experimental results show that the space adaptation protocol

is the most efficient protocol with great scalability, while the negotiation protocol can provide better privacy guarantee with some additional cost.

The rest of the paper proceeds as follows. We give the concepts and related issues in geometric perturbation in Section II for better understanding of the paper. In Section III, we will briefly review the multiparty framework and address the problem of perturbation unification under this framework. In Section IV, we present the three protocols and analyze their cost and privacy guarantee. The related factors in these protocols are further studied in experiments (Section V).

II. Preliminary

In this section, we will introduce basic concepts in geometric perturbation for better understanding of the paper. The primary focus will be on the related issues in the scenario of single data provider − a single data provider releases the perturbed data to the service provider or to the public for mining purpose. By convention we will use capital characters to represent matrices, and bold lower cases to represent vectors.

A. Geometric Perturbation and Privacy Protection

We first briefly describe the basic geometric perturbation method, with which the participants of collaborative mining will perturb their own private dataset before releasing it to other parties.

We define a geometric perturbation as a combination of random rotation perturbation, random translation perturbation, and noise addition. Without loss of generality, it can be represented as $G(X) = RX + \Psi + \Delta$. X denotes the original dataset with N rows and d columns and we sometimes also denote X by $X_{d\times N}$, R is a random orthonormal matrix [21], and Ψ is a random translation matrix. We define a random translation matrix as follows.

Definition 1. *A matrix Ψ is called a translation matrix, if $\Psi = \mathbf{t} \times \mathbf{1}'$, $\mathbf{t} = [t_1, t_2, \ldots, t_d]'$ $(0 \le t_i \le 1,\ 1 \le i \le d)$, and $\mathbf{1} = [1, 1, \ldots, 1]'$.*

$\mathbf{t}$ is randomly generated based on the uniform distribution over [-1, 1]. Δ is a noise matrix with i.i.d. (independent identically distributed, with zero mean and small variance) elements, which is used to perturb the distances so that the perturbation is resilient to certain kind of attacks. If Ψ_1 and Ψ_2 are translation matrices and R is an orthogonal transformation, it is easy to verify that $\Psi_1 +/- \Psi_2$ and $R\Psi_i$ are also translation matrices.

While it is delicate to find an appropriate R in terms of the resilience to attacks, both $\mathbf{t}$ and the noise component of $G(X)$ can be generated independently. In initial investigation, Δ with some general setting, such as Gaussian $N(0, \sigma^2)$ and $\sigma = 0.1$, can provide satisfactory resilience to the identified attacks, and still maintain high model accuracy [7].

In previous work [5], geometric perturbation is specially designed for a family of "geometric transformation invariant classifiers", include KNN, kernel methods, linear classifiers, and SVM classifiers with the commonly used kernels. However, more mining models can be added to this list, including most clustering models. The major benefit of such a transformation is: for a given dataset, all geometric perturbations can give similar model accuracy for these classifiers; thus, an individual data provider needs to select only one perturbation that can provide satisfactory privacy guarantee. We next define what is a "good" perturbation in terms of privacy guarantee.

Privacy Guarantee for Multidimensional Perturbation Geometric perturbation is a multidimensional data perturbation. In contrast to single dimension data perturbation [2], data in all columns are perturbed together in a multidimensional transformation. Thus, the privacy guarantee of an individual data column is correlated to the privacy guarantee of other columns. We define the privacy guarantee of a multidimensional perturbation as follows.

Let C_i be a random variable representing the *normalized* data of column i in the original dataset so that the values across the d columns are comparable $(1 \le i \le d)$. Let O_i be a random variable representing the observed data of column i, which can be the perturbed data or the data reconstructed from the perturbed data by particular attacks. Both C_i and O_i are normalized so that scaling will not artificially increase the privacy guarantee. We use p_i to denote the privacy guarantee on column i and define p_i by the standard deviation of the difference between O_i and C_i, namely $p_i = stdev(O_i - C_i)$. If both O_i and C_i are normalized to [0,1]. Figure 1 gives an intuitive understanding of p_i.

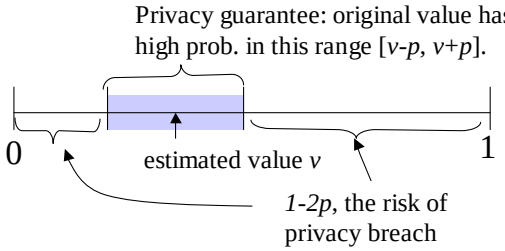

Fig. 1. Understanding privacy guarantee

Furthermore, let the significance of privacy protection for each column have a normalized weight w_i, we define two basic composite metrics by comparing p_i/w_i across all d columns:

$$\Phi_1(p_1, \ldots, p_d, w_1, \ldots, w_d) = \min\{\frac{p_1}{w_1}, \frac{p_2}{w_2}, \ldots, \frac{p_d}{w_d}\}$$

$$\Phi_2(p_1, \ldots, p_d, w_1, \ldots, w_d) = \frac{1}{d}\sum_{i=1}^{d}\frac{p_i}{w_i}$$

Φ_1 is called *Minimum Privacy Guarantee*, which defines the lowest privacy guarantee that a column should have with the given perturbation. Φ_2 is called *Average Privacy Guarantee*, which defines the average privacy guarantee over all d columns. In the sequel, we will use Φ_1 to represent the "privacy guarantee" of a perturbation.

Attack analysis and perturbation optimization: In optimizing geometric perturbation, we also need to consider the resilience to attacks. We identify three categories of attacks to geometric perturbation, according to the amount of external

knowledge the attacker may have [7].
(1) *Naive-estimation attack* is the first category, where attackers have no additional knowledge about the original data, so they simply estimate the original records from the perturbed data.
(2) *Independent Component Analysis (ICA) based attack* is the second category of attacks. When attackers know some column statistics, such as the maximum/minimum values and the probability density function of each column, they can try to reconstruct the original dataset with ICA techniques. The effectiveness of ICA-based attack is determined by the property of the original dataset. We can find a good perturbation resilient to ICA attack in most cases.
(3) *Distance-inference attack* is the third category of attack. If the attacker knows enough number of original data records and their maps in the perturbed dataset, they can use this kind of knowledge to break geometric perturbation. The noise component Δ in geometric perturbation is used to perturb the distance relationship and make the perturbation resilient to the distance-inference attack. Initial experiments show that low noise intensity can satisfactorily reduce the accuracy of distance-inference attack, and still preserve the desired model accuracy.

A randomized perturbation optimization algorithm [7] is designed for finding a good perturbation with satisfactory resilience to the discussed attacks. In general, compared to randomly generated perturbations (the components R, Ψ,Δ are randomly selected), the optimized perturbation can give significantly higher privacy guarantee. This optimization algorithm will also be used in our multiparty protocols.

III. Multiparty Mining Service Framework

In this section we give an overview of the data-mining-as-a-service framework for multiparty collaborative data mining. We will focus the discussion on the issues in applying geometric perturbation to multiparty mining under this framework. First, we will briefly introduce the involved parties in the framework, and then give the major issues of applying geometric perturbation to the multiparty scenario.

A. Overview and Threat Model

Service computing is becoming a major paradigm in distributed computing and business processing. Since data mining is a resource-intensive task, involving highly centralized expertise and computing power, it can be a valuable service supported by the companies or the research institutes that have abundant resources. Interested in finding valuable global models, multiple parties can use such data mining services by providing restrictive sharing of their data. One of the major concerns in collaborative mining is preserving the sensitive information for each participating data provider, while maintaining high quality of the mined models (or the utility of the pooled data).

Figure 2 shows the parties and the possible interactions between them. The service provider (S) is a party who owns abundant computing power, data mining tools and talents and willing to offer their data mining services to the contracted parties through certain service provision scheme. All data providers (notated by P_i) provide their own data which may contain sensitive information and they are willing to collaboratively find global models. Besides the two kinds of parties, sometimes, trusted servers or semi-trusted commodity servers [4] may also be used. However, in this paper, we do not assume trusted parties are present because they are difficult to find in practice.

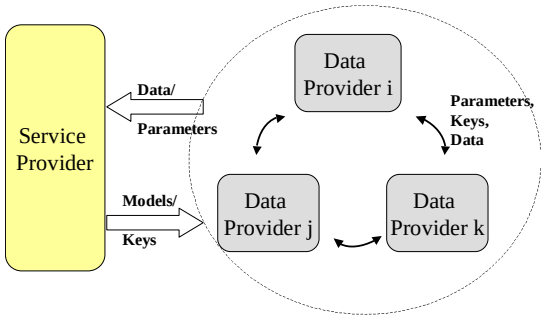

Fig. 2. Service-oriented multiparty privacy preserving data mining.

In this paper we assume a semi-honest threat model for all parties. A party is said to be semi-honest, if she will honestly follow the multiparty interaction protocol agreed upon by all the participants in the protocol, but she might be curious about any potentially private information contained in the intermediate results that she receives.

By assuming a semi-honest threat model, we do not consider the scenarios where either the service provider or any of data providers is malicious. Malicious adversaries may do anything to infer secret information. They can abort the protocol at any time, send spurious messages, spoof messages, collude with other (malicious) parties, etc.

We believe the semi-honest assumption is realistic for secure and privacy preserving multiparty computing. For example, consider the case where credit card companies jointly build data mining models for credit card fraud detection, which need to share the credit card fraud transactions. Such sharing is typically controlled by managing to whom and at what extent such sharing will take place. It is more realistic to assume the players (the credit card companies) are not malicious but semi-honest in nature. More importantly, in this type of scenarios, protocols developed under the semi-honest threat model enable the participants to collaboratively perform data mining models without considering the insider attacks.

In addition, most of existing research on secure and privacy preserving computing assumes semi-honest model where the players may exhibit malicious behavior in limited context given that the admission to participate in a planned distributed collaboration is to some extent controlled in most of the real world applications. Many of the previous multiparty computational protocols are based on this assumption, such as secure multiparty computation [10], multiparty privacy preserving association rule mining [23], [15], multiparty decision tree mining [17], and multiparty k-means clustering [24].

Note that in multiparty environment, *anonymization* might also become a key factor in privacy preservation. In many

cases, the private information becomes valuable to the privacy attacker only when the owner of the private information is identified. However, we will not consider anonymization in the current version.

The semi-honest model in this paper is also relaxed – the assumption of no collusion between any parties is not strictly held. The design of our protocols allows the collusion between data providers. However, the service provider should not collude with any of the data providers. Otherwise, the protocols may become too complicated and costly (we will discuss this later).

In our threat model, passive logging and eavesdropping over the network are possible. Therefore, encryption is needed for transmitting secrets. In the rest of the paper we will focus on the potential privacy breaches via the transmitted datasets and parameters that *a party can normally decrypt and see* in the protocol, which are mainly caused by the *curious service provider*, the *curious data providers*, and the *collusive data providers*.

B. Why Do We Need to Unify Perturbations

With geometric perturbation, each data provider can employ the geometric perturbation algorithm to obtain a locally optimized perturbation regarding to its own dataset. However, if we want to use all datasets for mining, we still need to unify all datasets under one perturbation. We will discuss the reason why we need perturbation unification.

A geometric transformation changes the coordinates of the data points, i.e., transforms data points in one coordinate system to another, while preserving the distance information that is critical to the applicable mining models. When datasets are transformed differently, although the distance information is preserved within a particular dataset, it is not preserved crossing different datasets. There might exist multiple ways to preserve distance between datasets, but we will use perturbation unification in this paper to address this problem.

Let the original vector space V_0 denote a d-dimensional data space.By using the geometric perturbation G_i, we transform the vector space V_0 to any target space V_t. For clear presentation, in the rest of the paper we use the geometric perturbation G_i to represent the transformed vector space V_i, and the "vector space" is equivalent to the "data space" as well. Let $\{X_1, X_2, \ldots, X_k\}$ denote the sub-datasets in V_0, each of which is held by one of the k data providers P_i $(1 \leq i \leq k)$, respectively. Let G_i be the transformation used by the data provider P_i. Clearly, the following are true :

- If $G_i \neq G_j$, directly merging the transformed datasets $G_i(X_i)$ and $G_j(X_j)$ will break the distance relationship between the original datasets X_i and X_j.
- Assume the models M_i and M_i depending on the distance information. If M_i is trained with $G_i(X_i)$, and M_j with $G_j(X_j)$, $G_i \neq G_j$, then M_i and M_j are not compatible due to unpreserved distance relationship between G_i and G_j.

Let G_t be the target unified space. One straightforward method is to make $G_i = G_t$ or indirectly transform G_i to G_t for any party i. It is then equivalent to directly transforming the pooled original data X to $G_t(X)$, as in the single-party scenario. The following protocols use these unification methods.

IV. Protocols for Perturbation Unification

In this section, we develop three protocols: the simple protocol, the negotiation protocol, and the space adaptation protocol. All of them are good candidates for certain application scenarios. We will address the problems and advantages associated with each protocol. In the following discussion, the service provider will also provide a public key for encrypting the data that only the service provider can decrypt. We will skip some common steps for all protocols, steps such as mining on the pooled data at the server side and applying the mined model to new data by the data provider.

A. Simple Protocol

The first protocol is quite simple, yet presenting some basic components that will also be used in other protocols. In this protocol, the data providers use the same randomly generated perturbation to perturb data. The basic issues include (1) how to securely generate the same random perturbation in each site, while preventing the curious service provider knowing the unified perturbation, and (2) how to prevent privacy breach caused by curious data providers.

The first issue can be addressed by the group-key based random perturbation generation. The data providers share the same random seed (the group key) to generate the same perturbation locally. There are abundant literatures on group key management [3], so we will skip the details here. The perturbed data cannot be delivered to the service provider directly, since the network is not secure and other data providers can log the transmitted data and recover the original data with the known perturbation. Thus, the perturbed data has to be encrypted with the public key provided by the service provider before it goes to public [1].

The service provider decrypts the perturbed data with her private key and pool the data together to mine a unified model. The unified model is returned to the data providers. Since the unified model is in the perturbed space, before the data provider applies it to the new data, she needs to transform her new data with the unified perturbation. The mining procedure and the model application procedure will be the same for all protocols. We will skip them in later discussions.

Apparently, there are a few weaknesses with this simple protocol. First of all, random perturbation may not provide same privacy guarantee for all data providers. We will study the difference between the distributions of privacy guarantee provided by random perturbation and by locally optimized perturbation in experiments. Secondly, encryption makes the data exclusively used in the current collaboration, and any of the perturbed datasets cannot be easily shared by the public or reused in other collaborations. Data providers may need to maintain multiple versions of perturbed data for different uses, which increases the maintenance cost.

[1]Note that the data should be encrypted in blocks, i.e., multiple records, otherwise, if different data providers have the same record, the singly encrypted data record can be easily identified.

B. Negotiation Protocol

Bearing the first weakness of the simple protocol in mind, the negotiation protocol aims at improving the overall privacy guarantee for all data providers. Some data providers may not be satisfied with the randomly generated perturbation in the simple protocol in terms of privacy guarantee. In the negotiation protocol, each data provider has a chance to review the candidate perturbation and vote for or against the candidate.

Due to different data distributions of the locally owned dataset, a data provider may prefer a different locally optimal perturbation than other perturbations possibly preferred by another data provider. Chances are slim that one perturbation works optimally for all data providers. The data providers may need to accept some suboptimal perturbations eventually. To evaluate the "satisfaction level" of a unified perturbation to the data provider, we define the following metric.

Definition 2. *Assume the locally optimized perturbation G_i gives privacy guarantee p_i^o for data provider P_i and the unified perturbation G_t gives p_i. The satisfaction level for P_i is defined by*

$$s_i = p_i / p_i^o$$

In the negotiation protocol , the agreement on the unified perturbation is reached by voting and negotiation. Each data provider P_i sets her own "minimum satisfaction level" s_i^{min}, which is the lower bound that a global perturbation is acceptable to the data provider. Then, each of the k data providers nominates her locally optimal perturbation, encrypts it by the group key, and distributes it to the other $k-1$ parties. At each party P_i, the $k-1$ candidate perturbations from other parties are evaluated and labeled with "accepted/rejected" according to the lower bound $s_i^{min} p_i^o$. Let p_{ij} be the privacy guarantee given by the perturbation G_j from P_j. P_i's vote to G_j is defined as follows.

$$q_{ij} = \begin{cases} 1 & p_{ij} \geq s_i^{min} p_i \\ 0 & p_{ij} < s_i^{min} p_i \end{cases}$$

When all parties return 1 to the party P_i, G_i is accepted by all parties. In whatever situation, P_i has to broadcast either her own perturbation is globally agreed or not. If multiple perturbations are agreed by all parties, only the one with lowest party ID is used as the global perturbation. If no perturbation is agreed on, another round of negotiation starts.

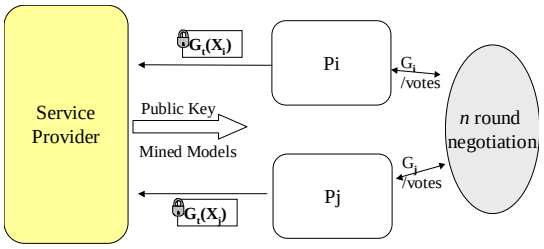

Fig. 3. Negotiation protocol.

The major issue is how efficient the negotiation process is in terms of the setting of local minimum satisfaction level. Apparently, a loose setting, i.e., a low local minimum satisfaction, will lead to fast agreements. Therefore, there is a tradeoff between the level of privacy guarantee and the efficiency of negotiation. We will further study this tradeoff in experiments.

C. Space Adaptation Protocol

The negotiation protocol can increase the overall privacy guarantee of the unified perturbation. However, the interactions between the parties are heavyweight, and, still, the perturbed data has to be encrypted before distribution. This step of encryption also makes the perturbed data exclusively used for the service provider in the current collaboration. Thus, the additional cost in maintaining different version of perturbed datasets still exists. In this section, we propose the third protocol, the space adaptation (SA) protocol, which inherits the convenience of distributing data in the single-party scenario, while also reduces the cost of communication, encryption and maintenance.

The space adaptation approach is based on the fact that geometric perturbations are transformable. We define the transformation of perturbation G_i to G_t as $G_{i\to t}$, the "Space Adpator", if G_t is the target perturbation. G_t can be represented as the composition of G_i and $G_{i\to t}$: $G_t = G_i \circ G_{i\to t}$. Specifically, for a given dataset X,

$$G_t(X) = (G_i \circ G_{i\to t})(X) = G_{i\to t}(G_i(X))$$

Note if G_i or G_t also contains a noise component, this equation becomes an approximation.

Although the overall satisfaction level to the unified perturbation is at the same level of the simple protocol, there are a few advantages by using the space adaptation protocol. In the space adaptation protocol, the data provider can simply distribute $G_i(X)$ without encryption, plus the encrypted space adaptor $G_{i\to t}$ for the particular collaboration. This brings considerable flexibility since $G_i(X)$ can be released to the public and be reused by future collaborations as well. While keeping the locally optimally perturbed data published and unchanged, the data provider just needs to change the space adaptor and encrypt it for different multiparty collaborative applications. With the combination of negotiation protocol, the overall satisfaction level can be improved as well. We will first give the detailed concept of space adaptation and then describe the protocol.

Concept of Space Adaptation As we discussed earlier, the perturbation parameters for the data provider i are $G_i : (R_i, \mathbf{t}_i)$, the translation matrix is $\Psi_i = \mathbf{t}_i \mathbf{1}_{n_i}^t$, and the original sub-dataset is X_i. Let Y_i be the perturbed data. Now, suppose that we want to transform Y_i to $Y_{i\to t}$ in the target space $G_t : (R_t, \mathbf{t}_t)$ that has *no noise component*. The following procedure can be applied. Since $Y_i = G_i(X_i) = R_i X_i + \Psi_i + \Delta_i$, and thus $X_i = R_i^{-1}(Y_i - \Psi_i - \Delta_i)$, the proof of the following equation is trivial:

$$Y_{i\to t} = R_t R_i^{-1} Y_i + (\Psi_t - R_t R_i^{-1} \Psi_i) - R_t R_i^{-1} \Delta_i$$

This equation consists of three components. We define the first component $R_t R_i^{-1}$ as *the rotation component of the*

adaptor $R_{i\to t}$. Apparently, $R_t R_i^{-1}\Psi_i$ is still a translation matrix (referring to the Definition 1, and thus we name the second part $\Psi_t - R_t R_i^{-1}\Psi_i$ as the *the translation component of the adaptor* $\Psi_{i\to t}$. The third part involves the original noise component and we name $\Delta_{it} = R_t R_i^{-1}\Delta_i$ as *the complementary noise component*.

Proposition 1. *Removing the complementary noise component in the target space G_t is equivalent to inheriting the noise component Δ_i from the original space G_i.*

PROOF SKETCH. Since Δ_i consists of i.i.d. elements with $N(0,\sigma^2)$, we have $E[R_t R_i^{-1}\Delta_i] = 0$ and covariance matrix

$$\begin{aligned} cov[R_t R_i^{-1}\Delta_i] &= R_t R_i^{-1} cov[\Delta_i](R_t R_i^{-1})^t \\ &= R_t R_i^{-1}\sigma^2\mathbf{I}(R_i^{-1})^t R_t^t = \sigma^2\mathbf{I} \quad (1) \end{aligned}$$

i.e., the transformed noise component has the same distribution with Δ_i. As this component is used to complement (de-randomize) the random noise in G_i, removing this component will exactly inherit the noise component of G_i. □

Therefore, we can reformulate space adaptation as follows:

$$Y_{i\to t} = R_{i\to t} Y_i + \Psi_{i\to t} \quad (2)$$

Where $R_{i\to t} = R_t R_i^{-1}$ and $\Psi_{i\to t} = \Psi_t - R_t R_i^{-1}\Psi_i$. We define the two components $< R_{i\to t}, \Psi_{i\to t} >$ as the space adaptor $G_{i\to t}$ from G_i to G_t. Clearly, by knowing data provider i's perturbed data $Y_i = G_i(X_i)$ and its space adaptor $G_{i\to t}$, one can transform the data to the target space.

With space adaptation, we split the perturbation into two parts, the perturbed data and the space adaptors that are used to transform the perturbed data to the global perturbation. This split brings two unique advantages: 1) the perturbed data can be safely released to any of the parties in terms of privacy preservation; 2) only small encryption cost is needed to safely transmit the space adaptors.

Protocol With space adaptation, now each data provider needs to publish two components: the perturbed data $G_i(X_i)$ and the space adaptor $G_{i\to t}$. The perturbed data is generated by using the locally optimized privacy guarantee, which is only known by the data provider. Therefore, the data can be directly published without encryption. The space adaptor can be used to recover the original perturbation by other data providers, since every data provider knows the unified perturbation. Therefore, the space adaptor $G_{i\to t}$ is only allowed to be known by the service provider. In other words, $G_{i\to t}$ has to be encrypted with the service provider's public key.

1) The same group-key based procedure that we have presented in the simple protocol is applied to setup the randomly generated unified perturbation G_t;
2) Each data provider generates the perturbation that is locally optimized for their own data, notated by G_i. With G_t and G_i, $G_{i\to t}$ can be calculated according to the definition;
3) Each data provider publishes $G_i(X_i)$ and transmits the encrypted $G_{i\to t}$ to the service provider;
4) The service provider decrypts the encrypted space adaptors and applies it to the corresponding perturbed data, which transforms the data to the unified space G_t. Then, the service provider can pool the datasets and train a unified model.

Figure 4 shows the components and interactions in the space adaptation protocol.

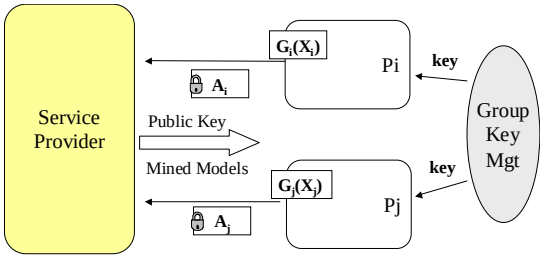

Fig. 4. Space adaptation protocol.

D. Performance Analysis

One of the major issues in multiparty computation is the cost, including the communication cost and the encryption(decryption) cost. In addition, we consider the reusability of the perturbed data, which saves the cost of future use of the dataset, as a part of performance analysis. In the following analysis, we calculate the cost of communication and encryption based on the data unit, e.g., a floating-point unit.

Table I summarizes these metrics for the three protocols. Assume that each party provides approximately the same number of records, say n records, each record has d dimensions, and there are k data providers in total. Let the average cost of local optimization for the dataset of above size be π. In the negotiation protocol, with certain setting the average number of negotiation rounds is r.

For the simple protocol, the communication cost consists of the maintenance of group key [3], which is $O(k)$, and the transmission cost of data, which is $O(knd)$. Since this protocol does not locally optimize perturbations, there is no cost in local optimization. Encryption is required to transmit the data, which cost $O(knd)$ in total. The perturbed data are exclusively used in a single collaboration task. The data provider may need to provide another locally optimized perturbed data for public access.

For the negotiation protocol, it requires maximum r rounds to get the agreed perturbation, r agreed by the parties. In each round, k perturbation parameters ($O(d^2)$ for each) are broadcast, which contributes to the communication cost $O(k^2d^2)$. The perturbation parameter is also encrypted in the broadcast, i.e., $O(kd^2)$. In each round, each data provider will also perform local optimization once. Similar to the simple protocol, the perturbed data is not reusable.

The space adaptation protocol is the most efficient one. The communication cost is the same as the simple protocol, since both depend on group key for generating the unified perturbation. Each data provider needs to generate local optimal perturbation once, and only the space adaptors ($O(d^2)$ units) need to be encrypted. Furthermore, the perturbed data can be reused for other purpose, which greatly decreases the maintenance cost.

	communication	local optimization	encryption	reusable
simple	$O(k(1+nd))$	0	$O(knd)$	no
negotiation	$O(rk^2d^2+knd)$	$rk\pi$	$O(rkd^2+knd)$	no
space adaptation	$O(k(1+nd))$	$k\pi$	$O(kd^2)$	yes

TABLE I
COST ANALYSIS FOR THE THREE PROTOCOLS

Overall, the negotiation protocol has the highest cost, which might also results in lower scalability. We will study the scalability issue in terms of the setting of satisfaction level.

E. Discussion on Risk of Privacy Breach

We first give a conceptual order of the overall satisfaction level of privacy guarantee provided by the three protocols, and then analyze the risk of privacy breach for each protocol. Finally, we discuss whether any collusion will increase the risk of privacy breach either between data providers or between the service provider and data providers.

The simple protocol and the space adaptation protocol employ a randomly generated unified perturbation, while the negotiation protocol optimizes the unified perturbation to some extent. Therefore, the ordering of overall satisfaction level can be roughly represented as follows.

$$s_{Simple}, s_{SA} < s_{Negotiation}$$

Note that the negotiation protocol can be used to generate the agreed perturbation for the space adaptation protocol to increase the overall satisfaction level, which, however, will limit the scalability of the protocol.

Risk of Privacy Breach

The risk of privacy breach for different protocols can be investigated through two types of adversaries: One is the curious data providers and the other is the curious service provider. We look at each protocol separately.

In the simple protocol, the data provider transmits encrypted perturbed data to the service provider, thus curious data providers cannot figure out any useful information from eavesdropping. The random perturbation is locally generated with the same algorithm, according to the shared seed, i.e., the group key sent by the service provider. If each party honestly follows the protocol, the curious data providers cannot find any information from the shared perturbation. The service provider can see all perturbed datasets submitted by the data providers. Since random perturbation does not guarantee all parties get high satisfaction level, the risk of privacy breach caused by the curious service provider might be higher for some data provider than others. The individual risk can be evaluated by the satisfaction level locally. If the data provider is not comfortable with certain satisfaction level, she can refuse to attend the collaboration.

The negotiation protocol enables multi-round voting to reach an agreed perturbation. In each round of negotiation, a data provider sees the perturbation parameters preferred by other data providers and their boolean votes to her perturbation. Since the perturbed data from other parties are all encrypted, without knowing the perturbed data, the curious data provider cannot utilize the perturbation parameters and boolean votes to breach privacy. As the result of negotiation, the unified perturbation is approved by all parties. Therefore, the risk from the curious service provider is greatly reduced, compared to the simple privacy.

If the space adaptation protocol uses random perturbation as the agreed perturbation, the risk of privacy breach from the curious service provider is similar to the simple protocol. Now, the data providers can see the published perturbed data, which was perturbed with locally optimized perturbation. Thus, the risk from curious data providers, as well as any unknown public privacy attackers, is minimized. The space adapters are all encrypted so that curious data providers cannot utilize them.

Discussion on Collusion

For all three protocols we discussed, we do not allow the collusion between the service provider and any of the data providers. If this type of collusion happens, the current protocols will not work. The service provider can exactly recover all original datasets, if she knows the unified perturbation G_t, which can be provided by the colluded data provider. Can we revise the protocols and make them resilient to this type of collusion? Probably, but the cost may increase dramatically. The key point to make this collusion ineffective is to prevent data providers knowing the unified perturbation G_t as well. To achieve this, we may need a trusted server to take care of the perturbation unification. Since the data provider does not know G_t, she would not be able to use the mined model locally. Therefore, the trusted party may also need to involve in model application. Without a better solution, simply revised protocols will put too much burden to the trusted party. We will leave this challenging issue for the future study.

However, with the three protocols, the collusion between two data providers will not increase the risk of privacy breach of other data providers. In the simple protocol and the negotiation protocol, one data provider cannot see another's perturbed data since the data are encrypted. Therefore, collusion between two data providers brings no additional information of the third data provider. In the space adaptation protocol, two components are published: one is the data perturbed with a locally optimized perturbation, known by no party except the data owner, and the other is the encrypted space adaptor, which can be decrypted by only the data owner and the service provider. Collusion will not provide additional information of the third data provider. Therefore, collusion between data providers will not increase the risk of privacy breach in the space adaptation protocol as well.

V. EXPERIMENTS

In this section, we present four sets of experiments on evaluating the effectiveness of the proposed three protocols

	Simple	Negotiation	Space adaptation
Curious data providers	none	none	very low
Curious service provider	random	low	random/low
Other attackers	none	none	very low

TABLE II
RISK OF PRIVACY BREACH FROM DIFFERENT ADVERSARIES FOR THE THREE PROTOCOLS

(simple, negotiation, and space adaptation). The first set of experiments shows the difference between locally optimized perturbations and randomly selected perturbations; The second set studies the relationship between the setting of minimum satisfaction level and the efficiency of negotiation in the negotiation protocol; The third set compares the satisfaction level between the protocols; Finally, the fourth set of experiments shows the preservation of model accuracy by using these protocols.

A. Setting of Experiments

The perturbation optimization algorithm used by each data provider uses the fastICA implementation[2] to test the resilience of the candidate perturbation to the ICA-based attacks [7]. We use two representative classifiers: KNN classifer and SVM with radial basis function kernel to show model accuracy preservation. The SVM implementation is from LIBSVM[3], and in our KNN implementation, we also use the kd-tree implemented in ANN library[4] to efficiently search the nearest neighbors.

Twelve UCI machine learning datasets are used in experiments. Each dataset are duplicated 10 times to generate a larger dataset. Then, we randomly split them into several random-sized sub-datasets, simulating the distributed datasets from the data providers. In our experiments, we also simulate two special partition distributions: the class-biased partition and the uniform partition (as illustrated in Figure 5 and 6) for the distributed datasets. In some experiments, we will choose to show the detailed results of a few featured datasets: Diabetes dataset that has an unclear geometric class boundary (KNN with accuracy about 73%), Shuttle dataset that has geometrically well-separated three major classes and a few tiny classes (KNN with accuracy about 99%), and Votes dataset that is a boolean dataset.

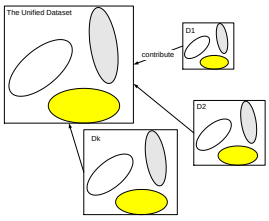

Fig. 5. Uniform partition of the pooled data

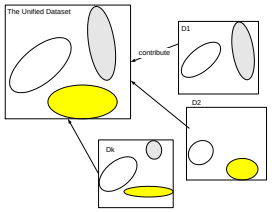

Fig. 6. Class-biased partition of the pooled data

The proposed protocols will use two types of algorithms to generate perturbations: randomized and optimized. Randomly generated perturbation means the three components R, Ψ and Δ are randomly generated. The rotation component R is generated from the QR decomposition [19] of a uniform random matrix; the elements of Ψ is uniformly selected from the range [-1,1]; the elements in Δ are i.i.d drawn from $N(0, 0.1^2)$. The optimization algorithm [7] mainly optimizes the rotation component R, while the other two components are generated in the same way as in the randomized method. The simple protocol will share a randomly generated perturbation by sharing the same randomization seed, which is generated from the shared group key with some hashing function. In the negotiation protocol, each party will generate its own locally optimized perturbation as the baseline for calculating the satisfaction level. In the space adaptation protocol, each party calculates the shared target perturbation, G_t, with the group key, and then generates its own locally optimized G_i. The rotation component and the translation component of the adaptor $G_{i\to t}$ can be calculated with G_t and G_i using the formula:

$$R_{i\to t} = R_t R_i^{-1} \tag{3}$$

$$\Psi_{i\to t} = \Psi_t - R_t R_i^{-1} \Psi_i \tag{4}$$

B. Random Perturbation vs. Optimized Perturbation

Local optimization gives significantly better perturbation than randomly generated perturbations in terms of privacy guarantee [5], [7]. Certainly, all parties will like to preserve the gain of privacy guarantee, i.e., preferring their locally optimized perturbations, when attending the multiparty collaborative mining. In this section, we show the difference between optimized perturbations and randomly generated perturbations in terms of cost and benefit, justifying that some costly protocols such as the negotiation protocol has its value in certain applications.

We use the three typical datasets in the experiments. For each dataset, we generate 1000 perturbations with randomization and with optimization, respectively. Since the optimization process is also randomized, against the attacks to the algorithm itself, the privacy guarantee of the generated perturbations will also show certain randomness. In Figure 7, 8 and 9, the x-axis is the minimum privacy guarantee of the perturbation as defined in Section II-A, and the y-axis is the number of perturbations having the corresponding privacy guarantee. These three figures show that optimized perturbations often have significantly higher privacy guarantee than randomly generated perturbations.

C. Efficiency of Negotiation

Since the interactions between parties are straightforward in the simple protocol and the space adaptation protocol, the

[2]http://www.cis.hut.fi/ projects/ica/fastica/
[3]http://www.csie.ntu.edu.tw/ cjlin/libsvm/
[4]http://www.cs.umd.edu/ mount/ANN/

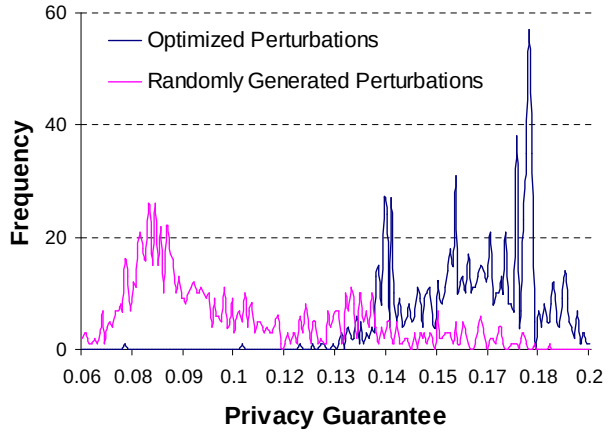

Fig. 7. Sample distribution of privacy guarantee (Shuttle).

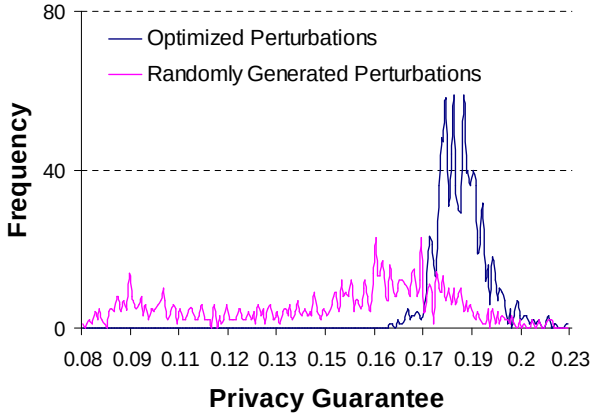

Fig. 8. Sample distribution of privacy guarantee (Diabetes)

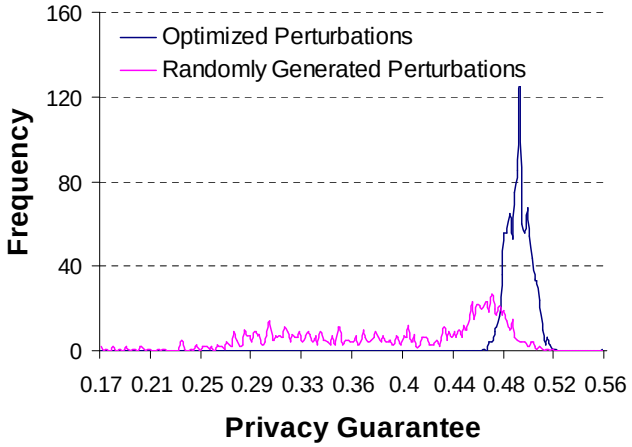

Fig. 9. Sample distribution of privacy guarantee (Votes)

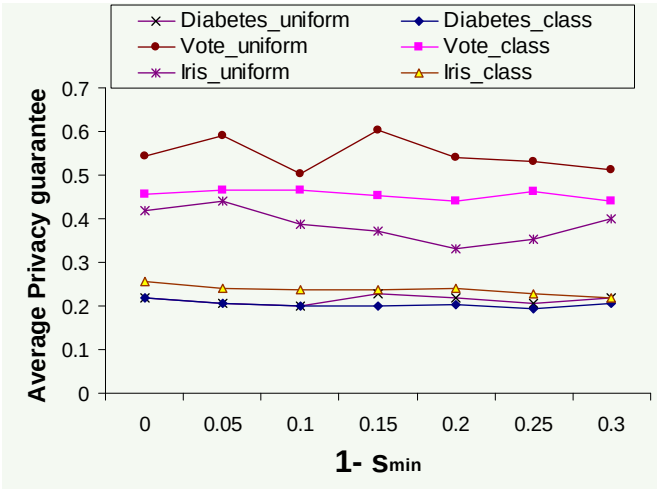

Fig. 12. Privacy guarantee vs. relaxation of minimum satisfaction level

formal analysis of time complexity are sufficient. However, the negotiation protocol involves the number of negotiation rounds, which the formal analysis cannot determine. We believe that the setting of minimum satisfaction level s_{min} has an intuitive impact on the success rate of negotiation. Besides that, we also notice that partition distribution can affect the efficiency of negotiation. We choose to show the results of the three typical datasets here with the setting of five-party collaborative mining (five data providers). Figure 10 and 11 show the average results of 10 tests for each dataset, each partition distribution, and each setting of minimum satisfaction level. We use 50 rounds as the upper limit of the number of rounds doing negotiation, i.e., if the parties cannot agree on any good perturbation in 50 rounds, we simply stop it.

$$\text{success rate} = \frac{\text{\# successful negotiations}}{50}$$

Note that class-based partition has more impact on the Votes dataset (boolean) than the other two datasets, possibly due to the type of data. In addition, with a little relaxation on the minimum satisfaction level, the negotiation protocol is pretty efficient. For example, for uniform partition, if s_{min} is relaxed from 1 to 0.8, the success rate rises from almost 0 to around 60% to 90%. On the other hand, it is a little more difficult to agree on a good perturbation for class-based partition, since each subset has very different distributions, which should results in different optimal perturbation. In particular, the success rate for Votes data increases slowly from 0 to 20% when s_{min} is relaxed to 0.8. Most importantly, Figure 12 shows that the relaxation of minimum satisfaction level does not significantly affect the average of the privacy guarantees from all parties, which implies it will be safe and efficient to relax the minimum satisfaction level in a small range, e.g., [0.8, 1].

So far we have not touched the scalability issue. The negotiation protocol seemly works pretty efficient for a small number of data providers. What if the number of parties increases? Figure 14 shows that with increasing number of parties, the effect to the performance of negotiation is not trivial. In general, class-based partition results in much worse performance. For example, the increase of parties makes the agreement of Votes/class-based partition quickly become very

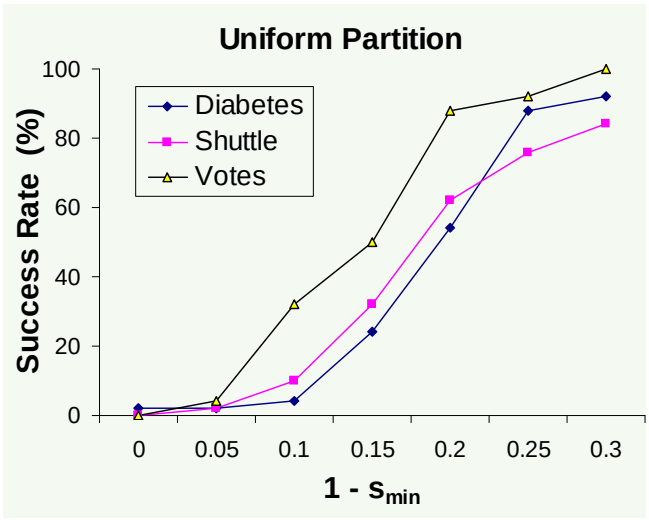

Fig. 10. Success rate of negotiation with uniform partition.

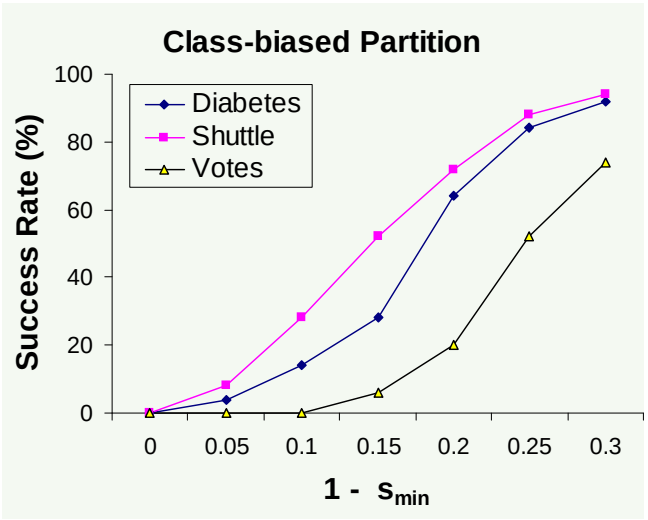

Fig. 11. Success rate of negotiation with class-biased partition.

difficult. In contrast, with uniform partition, the negotiation for the Votes data is still quite efficient, a successful negotiation happens in about 2-3 rounds on average at ten parties. Overall, the efficiency of negotiation is seemly determined by both the class distribution of the pooled dataset and the partition distribution of the distributed datasets.

D. Satisfaction level to Unified Perturbation

One of the major metrics in perturbation unification is the satisfaction level of privacy guarantee. We have used it in the evaluation of the negotiation protocol. We will further compare the overall satisfaction levels between the negotiation protocol and the other two protocols, which are based on randomly generated unified perturbations.

Figure 13 shows the comparison. We show the average of min/max satisfaction level together with the average satisfaction level among all parties. For the negotiation protocol, if we set the minimum satisfaction level to 0.8 for five parties, the resultant satisfaction level is quite high – on average it is above 0.9 which confirms the observed pattern in Figure 12. At lease one party gets perfect satisfaction level 1.0 or even higher. A satisfaction level higher than 1 can happen because the local perturbation optimization algorithm is a hill climbing algorithm, which does not guarantee to get the best perturbation – In some cases, a perturbation from another party might be better than the one optimized locally. Interestingly, although the minimum satisfaction level from randomly generated unified perturbations is lower than that from the negotiation protocol, the average are reasonably high. Moreover, most of the parties keep more than 50% of their original privacy guarantee, and some parties may even get satisfaction level higher than 1. A global perturbation may give higher privacy guarantee than a locally optimized perturbation in both negotiation or random cases, which indicates some space for us to improve in the future in terms of privacy guarantee.

E. Preservation of Data Utility

We finalize the experiments with the study of data utility for the two representative classifiers: KNN classifier and SVM classifier with RBF kernel. One of the major tradeoffs in privacy preserving data mining is that between data utility (or model accuracy in the classification case) and privacy guarantee. However, most of this paper has been focused on the efficiency of protocols and privacy guarantee (or risk of privacy breach). In fact, all protocols we discussed so far do not involve factors that can significantly downgrade the quality of the pooled dataset. In other words, data utility should be ideally as good as that in the single-party perturbation. The nuance may come from the perturbation of the noise component in the space adaptation protocol. According to Eq. 1, space adaptation will not change the intensity of noise component either. We will study this in experiments.

The pooled datasets generated by the simulation of protocols are used to train the two kinds of classifiers, KNN and SVM. The numbers in Figure 15 and 16 show the deviation from the standard accuracy which is obtained with the original unperturbed dataset. These numbers are the average of 10 rounds of randomized protocol simulation – in each round, data is split randomly (according to different partition distributions), and all local optimizations are done in a randomized manner. A negative number means that the actual accuracy is reduced. We use “SP”, “NP” and “SAP” to represent the three protocols, respectively. The results of different partition distributions for the space adaptation protocol are labeled with “SAP-Uniform” and “SAP-Class”. The result shows that partition distributions and protocols do not make significantly different impact on the accuracy, while SAP may have slightly more negative impact on some datasets. Therefore, the only factor is data itself, i.e., the class distribution, which is different from dataset to dataset.

VI. Related Work

Data perturbation changes the data in such a way that it is difficult to estimate the original values from the perturbed

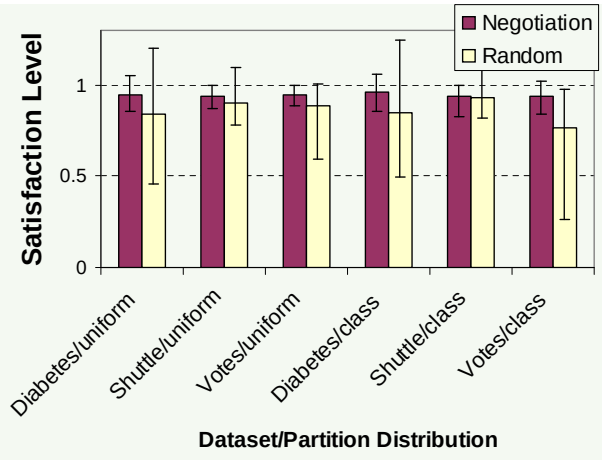

Fig. 13. The satisfaction level to the unified perturbation.

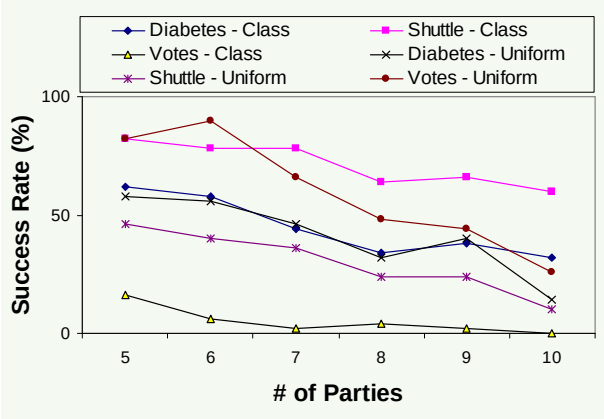

Fig. 14. Success rate vs. the number of parties.

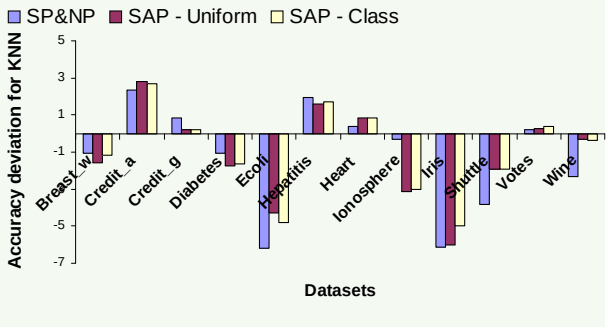

Fig. 15. The average deviation of model accuracy for KNN classifier.

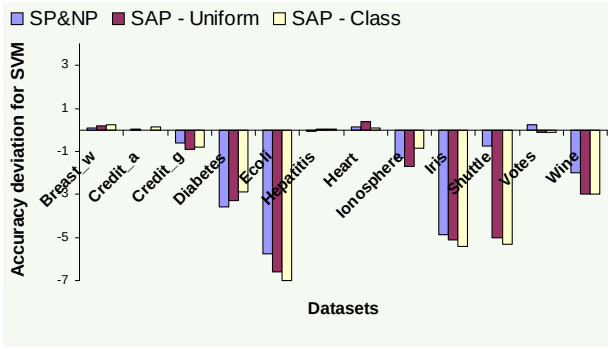

Fig. 16. The average deviation of model accuracy for SVM(RBF) classifier.

data, while some of the properties of the dataset critical to data mining are still preserved. Recently data perturbation techniques have become popular for privacy-preserving data mining [2], [9], [1], [22], [5], due to the relatively low cost to deploy them compared to the cryptographic techniques [17], [23], [24], [15], [13]. However, there are a few challenges in the data-perturbation based privacy-preserving data mining. First, it is commonly recognized that it is critical but difficult to balance data utility (affecting the model accuracy in the classification case) and data privacy. Second, potential attacks to the data perturbation methods are not sufficiently considered in previous research. A few works have started to address the privacy breaches to randomization approaches, by applying data reconstruction techniques [7], [12], [16] or the domain knowledge [8]. Third, some approaches, such as randomization approach [2], require to develop new data mining algorithms to mine the perturbed data, which raises extra difficulty in applying these techniques. To address these challenges, it is critical to understand the intrinsic relationship between data mining models and the perturbation techniques.

The previous work [5], [7] has investigated the perturbation techniques from the perspective of the specific data mining models. The authors observed that different data mining tasks/models actually care about different properties of the dataset, which could be statistical information, such as the column distribution and the covariance matrix, geometric properties, such as distance, and so on. Clearly, it is almost impossible to preserve all of the information in the original dataset in data perturbation. Thus, perturbation techniques should focus on preserving only the task-specific information in the dataset that is critical to the specific data mining task/model, in order to bring better flexibility in optimizing data privacy guarantee. The initial study on the geometric perturbation approach to data classification [5] has shown that the *task/model-specific data perturbation* can provide better privacy guarantee and better model accuracy. Liu et al. [18] also discussed the scenarios where a general multiplicative data perturbation is applied. However, such perturbation may not preserve the model accuracy well for the classifiers we have mentioned.

Data perturbation is particularly good for a single data owner publishing his/her own data. It may raise particular issues when applied to multiparty collaborative mining. Recent research [27], [2] also mentioned the service-oriented framework for collaborative privacy-preserving data mining with data perturbation. Another branch of multiparty privacy preserving data mining is derived from the basic idea of secure multiparty computation (SMC) [25]. The generic SMC protocols are costly and thus only applicable for small data. Lindell et al. [17] proposes a protocol for efficiently computing

mutual information from two-party distributed sources, which is the basis of ID3 decision tree algorithm [20]. Jagannathan et al. [13] proposes the cryptographic protocol for two-party secure Kmeans clustering. There a few more protocols are proposed [24], [26] for different data mining algorithms on vertically partitioned datasets. However, all of them are attached to certain data mining algorithm and not easy to extend to other data mining algorithms. Furthermore, most of them are two-party protocols. By increasing the number of parties, either the communication cost will increase exponentially or the original protocol does not work anymore. In contrast, our geometric perturbation based approach can be applied to multiple categories of mining algorithms with good scalability. We have reported some of the preliminary result, primarily on space adaptation [6]. In this paper, we present more protocols with a comprehensive evaluation.

VII. Conclusion and Future Work

Geometric perturbation has shown to be an effective perturbation method in single-party privacy preserving data publishing. In this paper, we present the geometric perturbation approach to multiparty privacy-preserving collaborative mining. The main challenge is to securely unify the perturbations used by different participants without much loss of privacy guarantee and data utility. We designed three protocols and analyzed the features and the cost of each protocol. The main factors and tradeoffs are also studied in the experiments. Overall, the space adaptation protocol provides a better balance between scalability, flexibility of data distribution, and the overall satisfaction level of privacy guarantee. For a small number of collaborative parties, we can also use the negotiation protocol which can provide better overall satisfaction level with some more communication cost.

The three protocols described in this paper represent our first effort on applying geometric data perturbation to multiparty privacy-preserving mining. Our work continues along several dimensions. First, it is known that, often, when the attacker knows where the breached information comes from, the damage becomes more substantial. We are interested in investigating the anonymization factor in the protocol design to further enhance the privacy preservation. Second, our current protocols assume that the service provider and the data providers do not collude. We are interested in investigating the challenging situation where this assumption is relaxed. Third, as the experimental result shows, the negotiation protocol can improve the overall privacy guarantee significantly. Therefore, it is meaningful to improve the negotiation protocol by seeking better balance between the satisfaction level and the efficiency of the protocol. Finally, in the current framework, we consider only the setting of one service provider and multiple data providers. We are interested in studying the privacy and security issues in the situation where multiple service providers collaboratively providing the privacy preserving mining service to multiple data providers.

Acknowledgement

This work is partially sponsored by grants from NSF CyberTrust program and NSF Computer Systems program, a grant from AFOSR, and a grant from Intel Research Council.

References

[1] C. C. Aggarwal and P. S. Yu, "A condensation approach to privacy preserving data mining," in *Proceedings of Intternational Conference on Extending Database Technology (EDBT)*, vol. 2992. Springer, 2004, pp. 183–199.

[2] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of ACM SIGMOD Conference*, 2000.

[3] Y. Amir, Y. Kim, C. Nita-rotaru, and G. Tsudik, "On the performance of group key agreement protocols," *ACM Transactions on Information and System Security*, vol. 7, no. 3, August 2004.

[4] D. Beaver, "Commodity-based cryptography," in *ACM Symposium on Theory of Computing*, 1997.

[5] K. Chen and L. Liu, "A random rotation perturbation approach to privacy preserving data classification," in *Proceedings of International Conference on Data Mining (ICDM)*, 2005.

[6] K. Chen and L. Liu, "Space adaptation: Privacy-preserving multiparty collaborative mining with geometric perturbation," in *Proceedings of IEEE Conference on Principles on Distributed Computing*, 2007.

[7] K. Chen and L. Liu, "Towards attack-resilient geometric data perturbation," in *SIAM Data Mining Conference*, 2007.

[8] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting privacy breaches in privacy preserving data mining," in *Proceedings of ACM Conference on Principles of Database Systems (PODS)*, 2003.

[9] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," in *Proceedings of ACM SIGKDD Conference*, 2002.

[10] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. Strauss, and R. N. Wright, "Secure multiparty computation of approximations," in *ICALP '01: Proceedings of the 28th International Colloquium on Automata, Languages and Programming,*. Springer-Verlag, 2001, pp. 927–938.

[11] T. Hastie, R. Tibshirani, and J. Friedmann, *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[12] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proceedings of ACM SIGMOD Conference*, 2005.

[13] G. Jagannathan and R. N. Wright, "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," in *Proceedings of ACM SIGKDD Conference*, 2005.

[14] A. Jain, M. Murty, and P. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, pp. 264–323, 1999.

[15] K. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *IEEE Transactions on Knowledge and Data Engineering*, 2004.

[16] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Proceedings of International Conference on Data Mining (ICDM)*, 2003.

[17] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Journal of Cryptology*, vol. 15, no. 3, pp. 177–206, 2000.

[18] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, 2006.

[19] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, 2000.

[20] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[21] L. Sadun, *Applied Linear Algebra: the Decoupling Principle*. Prentice Hall, 2001.

[22] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, 2002.

[23] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proceedings of ACM SIGKDD Conference*, 2002.

[24] J. Vaidya and C. Clifton, "Privacy preserving k-means clustering over vertically partitioned data," in *Proceedings of ACM SIGKDD Conference*, 2003.

[25] A. C. Yao, "How to generate and exhange secrets," in *IEEE Symposium on Foundations of Computer Science*, 1986.

[26] H. Yu, J. Vaidya, and X. Jiang, "Privacy-preserving svm classification on vertically partitioned data," in *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*. Springer, 2006.

[27] N. Zhang, S. Wang, and W. Zhao, "A new scheme on privacy-preserving data classification," in *Proceedings of ACM SIGKDD Conference*, 2005.

Dr. Keke Chen is an Assistant Professor in the Department of Computer Science and Engineering at Wright State University, Dayton OH, USA, since 2008. He received his PhD degree in Computer Science from the College of Computing at Georgia Tech, Atlanta GA, USA, in 2006. Keke's research focuses on distributed data intensive scalable computing, including distributed privacy-preserving collaborative mining, web search, databases, data mining and visualization. From 2002 to 2006, Keke worked with Dr. Ling Liu in the Distributed Data Intensive Systems Lab at Georgia Tech, where he developed a few well-known research prototypes, such as the VISTA visual cluster rendering and validation system, the iVIBRATE framework for large-scale visual data clustering, the "Best K" cluster validation method for categorical data clustering, and the geometric data perturbation approach for service-oriented privacy-preserving data mining. From 2006 to 2008, he was a senior research scientist in Yahoo! Search&Ads Science, working on research issues in international web search relevance and developing advanced data mining algorithms for large distributed datasets on the Cloud.

Dr. Ling Liu is a Professor in the College of Computing at Georgia Institute of Technology. She directs the research programs in Distributed Data Intensive Systems Lab (DiSL), examining various aspects of data intensive systems, ranging from distributed systems, network computing, wireless and mobile computing, to Internet data management and storage systems, with the focus on performance, security, privacy, and energy efficiency in building large scale Internet systems and services. Dr. Liu has published over 200 International journal and conference articles in the areas of distributed systems, Internet data management, and information security. Her research group has produced a number of open source software systems, among which the most popular ones are WebCQ, XWRAPElite, PeerCrawl. Dr. Liu is currently on the editorial board of several international journals, including IEEE Transactions on Service Computing (TSC), International Journal of Peer-to-Peer Networking and Applications (Springer), Wireless Network (WINET, Springer). Dr. Liu is a recipient of the best paper award of ICDCS 2003, the best paper award of WWW 2004, the 2005 Pat Goldberg Memorial Best Paper Award, the best data engineering paper award of Int. conf. on Software Engineering and Data Engineering 2008, and a recipient of IBM faculty award in 2003, 2006 2008. Dr. Liu's research is primarily sponsored by NSF, AFOSR, IBM, and Intel.