Enabling Multi-level Trust in Privacy Preserving Data Mining

Yaping Li
Minghua Chen

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2008-156
http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-156.html

December 13, 2008

Enabling Multi-level Trust in Privacy Preserving Data Mining

Yaping Li *, Minghua Chen #

*Department of Electrical Engineering and Computer Sciences, University of California at Berkeley
Berkeley, CA 94720, USA
*yaping@cs.berkeley.edu

#Department of Information Engineering, The Chinese University of Hong Kong
Shatin, NT, Hong Kong

#minghua@ie.cuhk.edu.hk

***Abstract*— Privacy Preserving Data Mining (PPDM) addresses the problem of developing accurate models about aggregated data without access to precise information in individual data record. A widely studied *perturbation-based PPDM* approach introduces random perturbation to individual values to preserve privacy before data is published. Previous solutions of this approach are limited in their tacit assumption of a single-level trust on data miners.**

In this work, we relax this assumption and expand the scope of perturbation-based PPDM to Multi-Level Trust (MLT-PPDM). In our setting, the more trusted a data miner is, the less perturbed copy of the data it can access. Under this setting, a malicious data miner may have the access to differently perturbed copies of the same data through various means, and may combine these diverse copies to jointly infer additional information about the original data that the data owner does not intend to release. Preventing such *diversity attacks* is the key challenge of providing MLT-PPDM service. We address this challenge by properly correlating perturbation across copies at different trust levels. We prove that our solution is robust against diversity attacks with respect to our privacy goal. That is, for data miners who have the access to an arbitrary collection of the perturbed copies, our solution prevent them from jointly reconstructing the original data more accurately than the best effort using any individual copies in the collection. Our solution allows a data owner to generate perturbed copies of its data for arbitrary trust levels on-demand. This feature offers data owners maximum flexibility.

I. Introduction

A widely employed and accepted Privacy Preserving Data Mining (PPDM) approach, which based on data perturbation, tacitly assumes a single-level of trust on data miners. This approach introduces uncertainty about individual values before data is published or released to third parties for data mining purposes [1], [2], [3], [4], [5], [6], [7]. Under the single trust level assumption, a data owner generates only one perturbed copy of its data with fixed amount of uncertainty. This assumption is limited in various applications where a data owner trusts the data miners at different levels.

We present below a two trust level scenario as a motivating example.

- The government or a business might do internal (most trusted) data mining, but they may also want to release the data to the public, and might perturb it more. The mining department which receives the less perturbed internal copy also has access to the more perturbed public copy. It would be desirable that this department does not have *more* power in reconstructing the original data by utilizing both copies than when it has only the internal copy.
- Conversely, if the internal copy is leaked to the public, then obviously the public has all the power of the mining department. However, it would be desirable if the public can not reconstruct the original data *more* accurately when it uses both copies than when it uses only the leaked internal copy.

This new dimension of *Multi-Level Trust* (MLT) poses new challenges for perturbation based PPDM. In contrast to the single-level trust scenario where only one perturbed copy is released, now multiple differently perturbed copies of the same data are available to data miners at different trusted levels. The more trusted a data miner is, the less perturbed copy it can access; it may also have the access to the perturbed copies available at lower trust levels. Moreover, a data miner could access multiple perturbed copies through various other means, e.g., accidental leakage or colluding with others.

By utilizing *diversity* across differently perturbed copies, the data miner may be able to produce a more accurate reconstruction of the original data than what is allowed by the data owner. We refer to this attack as *diversity attack*. It includes the colluding attack scenario where adversaries combine their copies to attack as a group; it also includes the scenario where an adversary utilizes public information to perform the attack on its own. Preventing diversity attacks is the key challenge facing the MLT-PPDM problem.

In this paper, we address this challenge in enabling MLT-PPDM services. In particular, we focus on the additive perturbation approach where random noise is added to the original data, and provide a systematic solution. Through a one-to-one mapping, our solution allows a data owner to generate distinctly perturbed copies of its data according to different trust levels. Defining trust levels and determining such mappings are beyond the scope of this paper.

A. Contributions

We make the following contributions:

- We expand the scope of perturbation based PPDM to multi-level trust, by relaxing the implicit assumption of a single-level trust in exiting work. MLT-PPDM introduces

another dimension of flexibility which allows data owners to generate differently perturbed copies of its data for different trust levels.
- We identify a key challenge in enabling MLT-PPDM services. In MLT-PPDM, data miners may have the access to multiple perturbed copies. By combining multiple perturbed copies, data miners may be able to perform diversity attacks to reconstruct the original data more accurately than what is allowed by the data owner. Defending such attacks is challenging, which we highlight through a case study in Section IV.
- We address this challenge by properly correlating perturbation across copies at different trust levels. We prove that our solution is robust against diversity attacks. We propose several algorithms for different targeting scenarios. We demonstrate the effectiveness of our solution through numerical evaluation.
- Our solution allows data owners to generate perturbed copies of their data at arbitrary trust levels on-demand. This property offers data owners maximum flexibility.

B. Related Work

Privacy Preserving Data Mining (PPDM) was first proposed in [2] and [8] simultaneously. To address this problem, researchers have since proposed various solutions that fall into two broad categories based on the level of privacy protection they provide. The first category of the Secure Multiparty Computation (SMC) approach provides the strongest level of privacy; it enables mutually distrustful entities to mine their collective data without revealing anything except for what can be inferred from an entity's own input and the output of the mining operation alone [8], [9]. In principle, any data mining algorithm can be implemented by using generic algorithms of SMC [10]. However, these algorithms are extraordinarily expensive in practice, and impractical for real use. To avoid the high computational cost, various solutions that are more efficient than generic SMC algorithms have been proposed for specific mining tasks. Solutions to build decision trees over the horizontally partitioned data were proposed in [8] . For vertically partitioned data, algorithms have been proposed to address the association rule mining [9], k-means clustering [11], and frequent pattern mining problems [12]. The work of [13] uses a secure coprocessor for privacy preserving collaborative data mining and analysis.

The second category of the partial information hiding approach trades privacy with improved performance in the sense that malicious data miners may infer certain properties of the original data from the disguised data. Various solutions in this category allow a data owner to transform its data in different ways to hide the true values of the original data while at the same time still permits useful mining operations over the modified data. This approach can be further divided into three categories: (a) k-anonymity [14], [15], [16], [17], [18], [19], (b) retention replacement (which retains an element with probability p or replaces it with an element selected from a probability distribution function on the domain of the elements) [20], [21], [22], and data perturbation (which introduces uncertainty about individual values before data is published) [1], [2], [3], [4], [23], [5], [6], [7].

The data perturbation approach includes two main classes of methods: additive [1], [2], [4], [5], [7] and matrix multiplicative [3], [6] schemes. These methods apply mainly to continuous data. In this paper, we focus solely on the additive perturbation approach where noise is added to data values.

Another relevant line of research concerns the problem of privately computing various set related operations. Two party protocols for intersection, intersection size, equijoin, and equijoin size were introduced in [24] for honest-but-curious adversarial model. Some of the proposed protocols leak information [25]. Similar protocols for set intersection have been proposed in [26], [27]. Efficient two party protocols for the private matching problem which are both secure in the malicious and honest-but-curious models were introduced in [28]. Efficient private and threshold set intersection protocols were proposed in [29]. While most of these protocols are equality based, algorithms in [25] compute arbitrary join predicates leveraging the power of a secure coprocessor. Tiny trusted devices were used for secure function evaluation in [30].

C. Paper Layout

The rest of the paper is organized as follows. We go over preliminaries in Section II. We formulate the problem, and define our privacy goal in Section III. In Section IV, we present a simple but important case study. It highlights the key challenge in achieving our privacy goal, and presents the intuition that leads to our solution. In Section V, we formally present our solution, and prove that it achieves our privacy goal. Algorithms that target at different scenarios are also proposed, and their complexities are studied. We carry out numerical evaluation in Section VI to verify our theoretical analysis. Section VII concludes the paper.

II. Preliminaries

A. Jointly Gaussian

Let G_1 through G_L be L Gaussian random variables. They are said to be *jointly Gaussian* if and only if each of them is a linear combination of multiple independent Gaussian random variables. Equivalently, G_1 through G_L are jointly Gaussian if and only if any linear combination of them is also a Gaussian random variable.

A vector formed by jointly Gaussian random variables is called a jointly Gaussian vector. For a jointly Gaussian vector $\mathbb{G} = [G_1, \ldots, G_L]^T$, its probability density function (PDF) is as follows: for any real vector g,

$$f_{\mathbb{G}}(g) = \frac{1}{\sqrt{(2\pi)^L \det(K_{\mathbb{G}})}} e^{-(g-\mu_{\mathbb{G}})^T K_{\mathbb{G}}^{-1}(g-\mu_{\mathbb{G}})/2},$$

where $\mu_{\mathbb{G}}$ and $K_{\mathbb{G}}$ are the mean vector and covariance matrix of $\mathbb{G}$, respectively.

If multiple random variables are jointly Gaussian, then conditional on some of them, the rest variables are still jointly Gaussian. This is a key property of jointly Gaussian variables. We utilize this property in Section V-C.

Note that not all Gaussian random variables are jointly Gaussian. For example, let G_1 be a zero mean Gaussian random variable with a positive variance, and define G_2 as

$$G_2 = \begin{cases} G_1, & \text{if } |G_1| \leq 1; \\ -G_1, & \text{otherwise.} \end{cases}$$

where $|G_1|$ is the absolute value of G_1. It is straightforward to verify that G_2 is Gaussian, but $G_1 + G_2$ is not. Therefore, G_1 and G_2 are not jointly Gaussian.

B. Additive Perturbation

Single-level trust PPDM via data perturbation has been widely studied in literature. In this setting, a data owner implicitly trusts all recipients of its data uniformly and distributes a single perturbed copy of the data.

A widely used and accepted way to perturb a data is by additive perturbation [1], [2], [4], [5], [7]. This approach adds to the original data, X, some random noise, Z, to obtain the perturbed copy, Y, as follows:

$$Y = X + Z. \tag{1}$$

We assume that X, Y, and Z are all N-dimension vectors where N is the number of attributes in X. Let x_j, y_j, and z_j be the j^{th} entry of X, Y, and Z respectively.

The original data X follows a distribution with mean vector μ_X and covariance matrix K_X. The covariance K_X is an $N \times N$ positive semi-definite matrix given by

$$K_X = E\left[(X - \mu_X)(X - \mu_X)^T\right], \tag{2}$$

which is a diagonal matrix if the attributes in X are uncorrelated.

The noise Z is assumed to be independent of X and is a jointly Gaussian vector with zero mean and covariance matrix K_Z chosen by the data owner. In short, we write it as $Z \sim N(0, K_Z)$. The covariance matrix K_Z is an $N \times N$ positive semi-definite matrix given by

$$K_Z = E\left[ZZ^T\right]. \tag{3}$$

It is straightforward to verify the mean vector of Y is also μ_X, and its covariance matrix, denoted by K_Y, is

$$K_Y = K_X + K_Z.$$

The perturbed copy Y is published or released to data miners. Equation 1 models both the cases where a data miner sees a perturbed copy of X, and where it knows the true values of certain attributes. The latter scenario is considered in recent work [7] where the authors show that sophisticated filtering techniques utilizing the true value leaks can help recover X.

In general, given Y, a malicious data miner's goal is to reconstruct X by filtering out the added noise. The authors of [4] point out that the attributes in X and the added noise should have the same correlation, otherwise the noise can be easily filtered out. This observation essentially requires to choose K_Z to be proportional to K_X [4], i.e., $K_Z = \sigma_Z^2 K_X$ for some constant σ_Z^2 denoting the perturbation magnitude.

TABLE I
KEY NOTATIONS

Symbol	Description
X	original data
Y_i	perturbed copy of X of trust level i
Z_i	noise added to X to generate Y_i
N	number of attributes in X
M	number of trust levels
$\mathbb{Y}$	a vector of all M perturbed copies
$\mathbb{Z}$	a vector of noisy Z_1 to Z_M
$\hat{X}(\mathbb{Y})$	LLSE estimate of X given $\mathbb{Y}$
K_X	covariance matrix of X
$K_{\mathbb{Z}}$	covariance matrix of $\mathbb{Z}$

C. Linear Least Squares Error Estimation

Given a perturbed copy of the data, a malicious data miner may attempt to reconstruct the original data as accurately as possible. Among the family of linear reconstruction methods, where estimates can only be linear functions of the perturbed copy, *Linear Least Squares Error* (LLSE) estimation has the minimum square errors between the estimated values and the original values.

According to the classical orthogonal principle in probability theory, the LLSE estimate of X given Y, denoted by $\hat{X}(Y)$, is the *projection* of X onto the set of linear functions of Y. The estimation error $X - \hat{X}(Y)$ is *orthogonal* to the observation Y, which means $X - \hat{X}(Y)$ is zero-mean and is uncorrelated to Y. Working out the math gives

$$\hat{X}(Y) = K_{XY} K_Y^{-1} (Y - \mu_X) + \mu_X, \tag{4}$$

where K_{XY} (K_Y resp.) is the covariance matrix of X and Y (Y resp.). K_{XY} is given by

$$\begin{aligned} K_{XY} &= E[(X - \mu_X)(Y - E[Y])^T] \\ &= E[(X - \mu_X)((X - \mu_X) + (Z - 0))^T] \\ &= K_X + 0. \end{aligned}$$

Note in the above derivation, we compute $E[(X - \mu_X)Z^T] = E[(X - \mu_X)]E[Z^T] = 0$, since X and Z are independent.

The square estimation errors between the LLSE estimates and the original values of the attributes in X are the diagonal terms of the covariance matrix of $X - \hat{X}(Y)$. An important property of LLSE estimation is that it simultaneously minimizes all these estimation errors.

III. PROBLEM FORMULATION

In this section, we present the problem settings, describe our threat model, state our privacy goal, and identify the design space. Table I lists the key notations used in the paper.

A. Problem Settings

In the MLT-PPDM problem we consider in this paper, a data owner trusts data miners at different levels and generates a series of perturbed copies of its data for different trust levels. This is done by adding varying amount of noise to the data.

Under the multi-level trust setting, data miners at higher trust levels can access less perturbed copies which data miners

at lower trust levels can not access, and possibly vice versa. In some scenarios, such as the motivating example we give at the beginning of Section I, data miners at higher trust levels may also have the access to the perturbed copies at more than one trust levels. Data miners at different trust levels may also collude to share the perturbed copies among them. As such, it is common that data miners can have the access to more than one perturbed copies.

Specifically, we assume that the data owner wants to release M perturbed copies of its data X, which is an $N \times 1$ vector with mean μ_X and covariance K_X as defined in Section II-B. These M copies can be generated in various fashions. They can be jointly generated all at once. Alternatively, they can be generated at different times upon receiving new requests from data miners, in an on-demand fashion. The latter case gives data owners maximum flexibility.

Let $\mathbb{Y} = [Y_1^T, \ldots, Y_M^T]^T$ be the vector of all perturbed copies $Y_i (1 \leq i \leq M)$ where T denotes transpose. Let $\mathbb{Z} = [Z_1^T, \ldots, Z_M^T]^T$ be the vector of noise. Let H be an $(N \cdot M) \times N$ matrix as follows:

$$H = \begin{bmatrix} I_N \\ \vdots \\ I_N \end{bmatrix},$$

where I_N represents an $N \times N$ identity matrix.

We have the relationship between $\mathbb{Y}$, X and $\mathbb{Z}$ as follows:

$$\mathbb{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_M \end{bmatrix} = \begin{bmatrix} I_N \\ \vdots \\ I_N \end{bmatrix} X + \begin{bmatrix} Z_1 \\ \vdots \\ Z_M \end{bmatrix} = HX + \mathbb{Z}, \quad (5)$$

where $Z_i, 1 \leq i \leq M$ are independent of X. To be robust against advanced filtering attacks, individual noise terms in Z_i added to different attributes in X should have same the correlations as the attributes themselves, otherwise Z_i can be easily filtered out [4]. As such, we have

$$K_{Z_i} = \sigma_{Z_i}^2 K_X, \text{ and } K_{Y_i} = (1 + \sigma_{Z_i}^2) K_X,$$

where $\sigma_{Z_i}^2$ is a constant of the perturbation magnitude. The data owner chooses a value for $\sigma_{Z_i}^2$ according to the trust level associated with the target perturbed copy Y_i.

B. Threat Model

We assume malicious data miners who always attempt to reconstruct a more accurate estimate of the original data given perturbed copies. We hence use the terms data miners and adversaries interchangeably throughout this paper. In MLT-PPDM, adversaries may have the access to a subset of the perturbed copies of the data. The adversaries' goal is to reconstruct the original data as accurately as possible based on all available perturbed copies they have access to.

The reconstruction accuracy heavily depends on the adversaries' knowledge. We make the same assumption as the one in [4] that adversaries have the knowledge of the statistics of the original data X and the noise $\mathbb{Z}$, i.e., mean μ_X, and covariance matrices K_X and $K_{\mathbb{Z}}$.

In addition, we assume adversaries only perform linear estimation attacks, where estimates can only be linear functions of the perturbed data Y. It is known that if X follows a jointly Gaussian distribution, then LLSE estimation achieves the minimum estimation error among both linear and nonlinear estimation methods. For X with general distribution, LLSE estimation has the minimum estimation error among all linear estimation methods. Various recent work in perturbation based PPDM, such as [4] and [5], makes this assumption of linear estimation. See reference [7] for a comprehensive review.

Noticed $K_{X\mathbb{Y}} = K_X H^T$ and $K_{\mathbb{Y}} = H K_X H^T + K_{\mathbb{Z}}$, the LLSE estimate $\hat{X}(\mathbb{Y})$ of X given $\mathbb{Y}$ can be expressed as:

$$\begin{aligned} \hat{X}(\mathbb{Y}) &= K_{X\mathbb{Y}} K_{\mathbb{Y}}^{-1} (\mathbb{Y} - E[\mathbb{Y}]) + \mu_X \\ &= K_X H^T \left[H K_X H^T + K_{\mathbb{Z}} \right]^{-1} (\mathbb{Y} - H\mu_X) \\ &\quad + \mu_X. \end{aligned} \quad (6)$$

In our setting, $\hat{X}(\mathbb{Y})$ is the most accurate estimate of X that an adversary can possibly make. The corresponding estimation errors of attributes in X are the diagonal terms of the covariance matrix of $\hat{X}(\mathbb{Y}) - X$. Using Equation 6, we can compute the covariance matrix as follows:

$$\begin{aligned} & E\left[\left(\hat{X}(\mathbb{Y}) - X \right) \left(\hat{X}(\mathbb{Y}) - X \right)^T \right] \\ = \; & K_X - K_X H^T K_{\mathbb{Y}}^{-1} H K_X = \left[K_X^{-1} + H^T K_{\mathbb{Z}}^{-1} H \right]^{-1}. \end{aligned} \quad (7)$$

For an adversary who observes only a single copy $Y_i \, (1 \leq i \leq M)$ and gets a LLSE estimate $\hat{X}(Y_i)$, the covariance matrix of $\hat{X}(Y_i) - X$ has a simple form as follows:

$$\begin{aligned} & E\left[\left(\hat{X}(Y_i) - X \right) \left(\hat{X}(Y_i) - X \right)^T \right] \\ = \; & K_X - K_X K_{Y_i}^{-1} K_X = \frac{\sigma_{Z_i}^2}{\sigma_{Z_i}^2 + 1} K_X. \end{aligned} \quad (8)$$

C. Definitions

1) Distortion: To facilitate future discussion on utility and privacy, we define the concept of perturbation $\mathcal{D}$ between two data as the average expected square difference between them. For example, the distortion between the original data X and the perturbed copy Y as define in Section II-B is given by:

$$\mathcal{D}(X, Y) = \frac{1}{N} \sum_{j=1}^{N} E[(y_j - x_j)^2].$$

It is easy to see that $\mathcal{D}(X, Y) = \mathcal{D}(Y, X)$.

Based on the above definition, we refer to a perturbed data Y_2 to be *more perturbed* than Y_1 with respect to X if and only if $\mathcal{D}(X, Y_2) > \mathcal{D}(X, Y_1)$.

2) Privacy and Utility: Single-level Trust : With respect to the original data X, the privacy of a perturbed copy Y represents how well the true values of X is hidden in Y and the utility of Y represents the amount of information about X contained in Y.

A more perturbed copy of the data does not necessarily have more privacy or less utility since the added noise may be intelligently filtered out. Consequently, we define the privacy of a perturbed copy by taking into account an adversary's power in reconstructing the original data. We define the

privacy of Y with respect to X to be $\mathcal{D}(X, \hat{X}(Y))$, i.e. the distortion between X and the LLSE estimate $\hat{X}(Y)$. We refer to a perturbed data Y_2 to preserve *more privacy* than Y_1 with respect to X if and only if $\mathcal{D}(X, \hat{X}(Y_2)) > \mathcal{D}(X, \hat{X}(Y_1))$.

As the distortion between $\hat{X}(Y)$ and X increases, Y's usefulness decreases. A larger distortion hides the original values better (and thus preserve more privacy), but it also hides more information about the original data. Consequently, we define the *utility* of Y with respect to X, to be the inverse of the distortion between X and $\hat{X}(Y)$, i.e. $1/\mathcal{D}(X, \hat{X}(Y))$. We refer to perturbed data Y_2 to have *less utility* than Y_1 with respect to X if and only if $\mathcal{D}(X, \hat{X}(Y_2)) > \mathcal{D}(X, \hat{X}(Y_1))$.

3) Privacy and Utility: Multi-level Trust: We now define privacy and utility for the multi-level trust case in the same spirit of the single-level trust case.

For a vector $\mathbb{Y} = [Y_1^T, \cdots, Y_M^T]^T$ of M perturbed copies of X, the privacy of $\mathbb{Y}$ represents how well the true values of X is hidden in the multiple perturbed copies $\mathbb{Y}$, and the utility of Y represents the amount of information about X contained in $\mathbb{Y}$. The privacy of $\mathbb{Y}$, with respect to X, is defined as $\mathcal{D}(X, \hat{X}(\mathbb{Y}))$, the distortion between X and its LLSE estimate $\hat{X}(\mathbb{Y})$. The utility of $\mathbb{Y}$ with respect to X is defined as the inverse of $\mathbb{Y}$'s privacy with respect to X.

Based on the definitions of privacy and utility, we see that one is uniquely determined by the other. As such, we use the terms privacy and utility interchangeably in this paper.

D. Privacy Goal and Design Space

In MLT-PPDM setting, a data owner releases distinctly perturbed copies of its data to multiple data miners. One key goal of the data owner is to control the amount of information about its data that adversaries may derive.

We assume that the data owner wants to distribute a total of M different perturbed copies of its data, i.e., $Y_i (1 \leq i \leq M)$, each for a trust level i. The assumption of M is for ease of analysis. As it will become clear later that our solution of the on-demand generation allows a data owner to generate as many different copies as it wishes.

The data owner can easily control the amount of the information about its data an attacker may infer from a single perturbed copy. Utilizing Equation 8, we express the privacy of Y_i, i.e. $\mathcal{D}(X, \hat{X}(Y_i))$, as follows:

$$\begin{aligned} & \mathcal{D}(X, \hat{X}(Y_i)) \\ = & \frac{1}{N} \sum_{i=1}^{N} E\left[(\hat{x}_i(Y_i) - x_i)^2 \right] \\ = & \frac{1}{N} Tr\left(E\left[\left(\hat{X}(Y_i) - X \right) \left(\hat{X}(Y_i) - X \right)^T \right] \right) \\ = & \frac{\sigma_{Z_i}^2}{\sigma_{Z_i}^2 + 1} \frac{1}{N} Tr\left(K_X \right), \end{aligned} \tag{9}$$

where $Tr(\cdot)$ represents the trace of a matrix.

The data owner can easily control the privacy and utility of individual copy Y_i by setting $\sigma_{Z_i}^2$ according to trust level i through a one-to-one mapping. Defining trust levels and such mappings are beyond the scope of this paper.

However, such control alone is not sufficient in the face of diversity attacks. Adversaries that can access copies at different trust levels enjoy the diversity gain when they combine multiple distinctly perturbed copies to estimate the original data. We discuss one such case in Section IV-B.1.

Ideally, the amount of information about X that adversaries can jointly infer from multiple perturbed copies should be no more than that of the best effort using any individual copies.

Formally, we say the privacy goal is achieved with respect to M perturbed copies $Y_i, 1 \leq i \leq M$, if the following statement holds. For an *arbitrary* subset $\mathbb{Y}_C$ of $\{Y_i, 1 \leq i \leq M\}$,

$$\mathcal{D}(X, \hat{X}(\mathbb{Y}_C)) = \min_{\xi \in \mathbb{Y}_C} \mathcal{D}(X, \hat{X}(\xi)). \tag{10}$$

where $\mathbb{Y}_C$ is the set of perturbed copies an adversary uses to reconstruct the original data.

Intuitively, achieving the privacy goal requires that given the copy with the highest utility among any subset of these M perturbed copies, the remaining copies in that subset contain no extra information about X.

To achieve this goal, the available design space is noise $\mathbb{Z}$. We already determine that individual noise $Z_i, 1 \leq i \leq M$ must follow $N(0, \sigma_{Z_i}^2 K_X)$. In the rest of the paper, we show by properly correlating noise $Z_i, 1 \leq i \leq M$, the desired privacy goal can be achieved.

IV. Case Study

In this section, we study a basic case corresponding to the motivating example we described at the beginning of Section I. In the case, a data miner has access to two differently perturbed copies of the same data, each for a different trust level. We present the challenges in achieving the privacy goal in Equation 10 with two false starts. As we develop a solution to this basic base, we show the key ideas in solving the more general case of arbitrarily fine granularity of trust levels.

A. An Illustrative Case

For ease of illustration, we assume that data contain only one attribute. We assume that the data owner has already distributed a perturbed copy Y_2 of the original data X where

$$Y_2 = X + Z_2.$$

The Gaussian noise $Z_2 \sim N(0, \sigma_2^2)$ is independent of X.

The data owner now wishes to produce another perturbed copy Y_1. It generates Gaussian noise $Z_1 \sim N(0, \sigma_1^2)$, and adds it to X to obtain Y_1 as

$$Y_1 = X + Z_1.$$

The new noise Z_1 is also independent of X (but could be designed to be correlated with Z_2). We consider the case where the data owner chooses $\sigma_2^2 > \sigma_1^2$ so that Y_1 is less perturbed than Y_2.

The privacy goal in Equation 10 requires that

$$\mathcal{D}(X, \hat{X}(Y_1, Y_2)) = \mathcal{D}(X, \hat{X}(Y_1)). \tag{11}$$

To see this, note that $\min(X, \mathcal{D}(\hat{X}(Y_1)), \mathcal{D}(X, \hat{X}(Y_2)))$ can be simplified to $\mathcal{D}(X, \hat{X}(Y_1))$, i.e., the less perturbed copy gives better estimate.

B. Two False Starts

In this section, we illustrate the challenges in achieving the privacy goal by two false starts.

1) Independent Noise: The first intuitive attempt is to generate the two perturbed copies independently. The added noise in the two perturbed copies is not only independent to the original data, but also independent to each other.

In the case we consider, the above solution generates Z_1 to be independent of X and Z_2 respectively. Consequently, adversaries have two perturbed copies as follows:

$$\begin{cases} Y_1 = X + Z_1 \\ Y_2 = X + Z_2 \end{cases}$$

where X, Z_1 and Z_2 are mutually independent. The adversaries perform a joint LLSE estimation to obtain $\hat{X}(Y_1, Y_2)$. Straightforward computation shows that

$$\mathcal{D}(X, \hat{X}(Y_1, Y_2)) = \frac{\sigma_X^2}{1 + \sigma_X^2/\sigma_1^2 + \sigma_X^2/\sigma_2^2}.$$

This value is strictly smaller than the error of the estimate based on either Y_1 or Y_2, which is for $i = 1, 2$,

$$\mathcal{D}(X, \hat{X}(Y_i)) = \frac{\sigma_X^2}{1 + \sigma_X^2/\sigma_i^2}.$$

Thus, Equation 11 is not satisfied and the desired privacy goal not achieved.

Intuitively, this is because the two copies of the data are generated independently, each containing some innovative information of the original data that is absent from the other. When estimation is performed jointly, the innovative information from both copies can be utilized, resulting in a smaller estimation error and thus a more accurate estimate.

2) Linearly Dependent Noise: In light of the incorrectness of the first solution, one might consider a second approach to generate new noise so that it is linearly dependent to the existing one.

In the case we consider, the above approach may generate $Z_1 = \frac{\sigma_1}{\sigma_2} Z_2$. It is easy to verify that $Z_1 \sim N(0, \sigma_1^2)$. However, $Y_1 = X + Z_1$ again fails to achieve the privacy goal.

To see this, notice that the adversaries who have the access to both copies can reconstruct X perfectly as follows:

$$X = \frac{\sigma_2 Y_1 - \sigma_1 Y_2}{\sigma_2 - \sigma_1} = \frac{\sigma_2(X + Z_1) - \sigma_1(X + Z_2)}{\sigma_2 - \sigma_1}.$$

The estimation error is zero, and Equation 11 is not satisfied.

C. Proposed Solution

Intuitively, Equation 11 requires that given Y_1, observing the more perturbed Y_2 does not improve the estimation accuracy.

One way to satisfy Equation 11 is to generate Z_1 so that $Y_1 = X + Z_1$ and $Z_2 - Z_1$ are independent. To see why, we re-write Y_2 as

$$Y_2 = Y_1 + (Z_2 - Z_1).$$

If Y_1 and $Z_2 - Z_1$ are independent, then Y_2 is nothing but a perturbed observation of Y_1. All information in Y_2 useful for estimating X is inherited from Y_1. Consequently, given Y_1, Y_2 provides no extra innovative information to improve the estimation accuracy, and Equation 11 is satisfied.

Since X and Z_1 (resp. Z_2) are independent, Y_1 and $Z_2 - Z_1$ are independent if Z_1 and $Z_2 - Z_1$ are independent. The following theorem gives a sufficient and necessary condition for Z_1 and Z_2 to satisfy that Z_1 and $Z_2 - Z_1$ are independent.

Theorem 1: Assume $Z_1 \sim N(0, \sigma_1^2)$, $Z_2 \sim N(0, \sigma_2^2)$, and $\sigma_1^2 < \sigma_2^2$. Z_1 and $Z_2 - Z_1$ are independent if and only if Z_1 and Z_2 are jointly Gaussian and their covariance matrix is

$$\begin{bmatrix} \sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_2^2 \end{bmatrix}. \tag{12}$$

Proof: Refer to Appendix A. ■

The following theorem states that Z_1 and $Z_2 - Z_1$ being independent is a sufficient condition for Equation 11 to hold.

Theorem 2: Given that $Z_1 \sim N(0, \sigma_1^2)$ and $Z_2 \sim N(0, \sigma_2^2)$, and $\sigma_1^2 < \sigma_2^2$, if Z_1 and $Z_2 - Z_1$ are independent, then Equation 11 holds.

Proof: Refer to Appendix B. ■

This sufficient condition is key in achieving the privacy goal in this simple case, as well as in the general cases, on which we elaborate in Section V.

Following the above analysis, our solution to this simple case is as follows:

- Given σ_1^2 and σ_2^2, construct the covariance matrix of Z_1 and Z_2 as in Equation 12. Derive the joint distribution of Z_1 and Z_2.
- Compute the conditional distribution of Z_1 given Z_2. Generate Z_1 according to this conditional distribution.
- Generate the desired $Y_1 = X + Z_1$.

In this way, Z_1 and $Z_2 - Z_1$ are guaranteed to be independent; hence, Equation 11 is satisfied.

V. Solution to General Cases

We now show that solutions to the general cases of arbitrarily fine trust levels follow naturally from that to the two trust level case studied in Section IV.

A. Shaping the Noise

1) Independent Noise Revisited: In Section IV, we show that adding independent noise to generate two differently perturbed copies, although convenient, fails to achieve our privacy gaol. The increase in the number of independently generated copies aggravates the situation; the estimation error actually goes to zero as this number increases indefinitely. In turn, the attackers can perfectly reconstruct the original data. We formalize this observation in the following theorem.

Theorem 3: Let $\mathbb{Y} = [Y_1^T, \ldots, Y_M^T]^T$ be a vector containing M perturbed copies. Assume that $\mathbb{Y}$ is generated from the original data X as follows:

$$\mathbb{Y} = HX + \mathbb{Z},$$

where $H = [I_N, \ldots, I_N]^T$, and $\mathbb{Z} = [Z_1^T, \ldots, Z_M^T]^T$ with $Z_i \sim N(0, \sigma_{Z_i}^2 K_X)$ is the noise vector.

If noises $Z_i, 1 \leq i \leq M$ are mutually independent, then the square errors between the LLSE estimate X and $\hat{X}(\mathbb{Y})$ are the diagonal terms of the following matrix

$$\left(1 + \sum_{i=1}^{M} \frac{1}{\sigma_{Z_i}^2}\right)^{-1} K_X.$$

As M goes to infinity, the estimation errors go to zero eventually, so does the distortion $\mathcal{D}(X, \hat{X}(\mathbb{Y}))$.

Proof: Refer to Appendix C. ∎

We conclude that noise $Z_i, 1 \leq i \leq M$ should not be generated independently.

2) Properly Correlated Noise: We show by the case study that the key to achieving the desired privacy goal is to have noise $Z_i, 1 \leq i \leq M$ properly correlated. To this end, we further develop the pattern found in the 2×2 noise covariance matrix in Equation 12 into a *corner-wave* property for a multi-dimensional noise covariance matrix. This property becomes the cornerstone of Theorem 4 which is a generalization of Theorem 1 and 2.

Corner-wave Property Theorem 4 states that for M perturbed copies, the privacy goal in Equation 10 is achieved if the noise covariance matrix $K_{\mathbb{Z}}$ has the corner-wave pattern as shown in Equation 14. Specifically, we say that an $M \times M$ square matrix has the corner-wave property if, for every i from 1 to M, the following entries have the same value as the $(i,i)^{th}$ entry:

- all entries to the right of the $(i,i)^{th}$ entry in row i,
- all entries below the $(i,i)^{th}$ entry in column i.

The distribution of the entries in such a matrix looks like corner-waves originated from the lower right corner.

Theorem 4: Let $\mathbb{Y} = [Y_1^T, \ldots, Y_M^T]^T$ represent an arbitrary number of perturbed copies. Assume that $\mathbb{Y}$ is generated from the original data X as follows:

$$\mathbb{Y} = HX + \mathbb{Z},$$

where $H = [I_N, \ldots, I_N]^T$, and $\mathbb{Z} = [Z_1^T, \ldots, Z_M^T]^T$ with $Z_i \sim N(0, \sigma_{Z_i}^2 K_X)$ is the noise vector. Without loss of generality, we further assume

$$\sigma_{Z_i}^2 < \sigma_{Z_{i+1}}^2, \quad \forall i = 1, \ldots, M-1. \tag{13}$$

Then the following equation holds[1]

$$\mathcal{D}(X, \hat{X}(\mathbb{Y})) = \min_{i=1,\ldots,M} \mathcal{D}(X, \hat{X}(Y_i)) = \frac{\sigma_{Z_1}^2}{\sigma_{Z_1}^2 + 1} \frac{1}{N} Tr(K_X),$$

if $\mathbb{Z}$ is a jointly Gaussian vector and its covariance matrix $K_{\mathbb{Z}}$ is given by

$$K_{\mathbb{Z}} = \begin{bmatrix} \sigma_{Z_1}^2 K_X & \sigma_{Z_1}^2 K_X & \cdots & \sigma_{Z_1}^2 K_X \\ \sigma_{Z_1}^2 K_X & \sigma_{Z_2}^2 K_X & \cdots & \sigma_{Z_2}^2 K_X \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{Z_1}^2 K_X & \sigma_{Z_2}^2 K_X & \cdots & \sigma_{Z_M}^2 K_X \end{bmatrix}. \tag{14}$$

Proof: Refer to Appendix D. ∎

Moreover, for any subset of these M perturbed copies, the covariance matrix of the corresponding noise also has the corner-wave property, and thus the privacy goal is achieved. We summarize this observation in Corollary1.

Corollary 1: If the privacy goal in Equation 10 is achieved with respect to M perturbed data $Y_1, \ldots, Y_M$, then the goal is also achieved with respect to any subset of $\{Y_1, \ldots, Y_M\}$.

Based on Theorem 4 and Corollary 1, one way to achieve the privacy goal in Equation 10 is to ensure that noise $\mathbb{Z}$ is a jointly Gaussian vector and follows $N(0, K_{\mathbb{Z}})$ where $K_{\mathbb{Z}}$ is given by Equation 14. We consider two scenarios when generating noise $\mathbb{Z}$ and the corresponding perturbed copies $\mathbb{Y}$. We discuss these two scenarios in the following two sections.

B. Batch Generation

In the first scenario, the data owner determines the M trust levels a priori, and generates M perturbed copies of the data in one batch. In this case, all trust levels are predefined and $\sigma_{Z_1}^2$ to $\sigma_{Z_M}^2$ are given when generating the noise. We refer to this scenario as the *batch generation*.

We propose two batch algorithms. Algorithm 1 generates noise $Z1$ to Z_M in parallel while Algorithm 2 sequentially.

1) Algorithm 1: Parallel Generation: Without loss of generality, we assume $\sigma_{Z_i}^2 < \sigma_{Z_{i+1}}^2$ where $1 \leq i \leq M-1$. Algorithm 1 generates the components of noise $\mathbb{Z}$, Z_1 to Z_M, simultaneously based on the following probability distribution function, for any real $(N \cdot M)$-dimension vector v,

$$f_{\mathbb{Z}}(v) = \frac{1}{\sqrt{(2\pi)^M \det(K_{\mathbb{Z}})}} e^{-\frac{1}{2} v^T K_{\mathbb{Z}}^{-1} v}, \tag{15}$$

where $K_{\mathbb{Z}}$ is given by Equation 14.

Algorithm 1 then constructs $\mathbb{Y}$ as $HX + \mathbb{Z}$ and outputs it. We refer to Algorithm 1 as *parallel generation*.

Algorithm 1 : Parallel Generation

1: // Input: X, K_X, and $\sigma_{Z_1}^2$ to $\sigma_{Z_M}^2$
2: // Output: $\mathbb{Y}$
3: Construct $K_{\mathbb{Z}}$ with K_X and $\sigma_{Z_1}^2$ to $\sigma_{Z_M}^2$, according to Equation 14
4: Generate $\mathbb{Z}$ with $K_{\mathbb{Z}}$, according to Equation 15
5: Generate $\mathbb{Y} = HX + \mathbb{Z}$
6: Output $\mathbb{Y}$

Algorithm 1 has a large memory requirement when generating a significant number of perturbed versions. Notice that it relies on generating an $M \times N$ Gaussian vector which requires a memory of size $O(M^2 \cdot N^2)$ to store the corresponding covariance matrix. The memory requirement grows quadratically with M for a fixed N. Algorithm 1 serves as a baseline algorithm for the next two algorithms.

2) Algorithm 2: Sequential Generation: The large memory requirement of Algorithm 1 motivates us to seek for a memory efficient solution. Instead of parallel generation, sequentially generating noise Z_1 to Z_M, each of which a Gaussian vector of N dimension, requires only a memory of size $O(N^2)$. The validity of the alternative procedure is based on the insight in the following theorem.

[1]In the equation, $Tr(\cdot)$ denotes the trace of a matrix.

Theorem 5: Consider $\mathbb{Z} = [Z_1^T, \ldots, Z_M^T]^T$ where $Z_i \sim N(0, K_{Z_i})$ with $K_{Z_i} = \sigma_{Z_i}^2 K_X$. Without loss of generality, further assume

$$\sigma_{Z_i}^2 < \sigma_{Z_{i+1}}^2, \quad \forall i = 1, \ldots, M-1. \tag{16}$$

Then $\mathbb{Z}$ is a jointly Gaussian vector and $K_{\mathbb{Z}}$ has the form in Equation 14, if and only if Z_1, and $(Z_i - Z_{i-1})$ are mutually independent.

Proof: Refer to Appendix E. ∎

Based on Theorem 5, Algorithm 2 sequentially generates M independent noise Z_1, and $(Z_i - Z_{i-1})$ for i from 2 to M. Noise Z_i is then simply $(Z_i - Z_{i-1}) + Z_{i-1}$ for i from 2 to M. Finally Algorithms 2 generates the perturbed copies Y_1 to Y_M by adding the corresponding noise. We refer to Algorithm 2 as *sequential generation*.

Algorithm 2 : Sequential Generation

// Input: X, K_X, and $\sigma_{Z_1}^2$ to $\sigma_{Z_M}^2$
// Output: Y_1 to Y_M
Construct $Z_1 \sim N(0, \sigma_{Z_1}^2 K_X)$
Generate $Y_1 = X + Z_1$
Output Y_1
for i from 2 to M **do**
    Construct noise $\xi \sim N(0, (\sigma_{Z_i}^2 - \sigma_{Z_{i-1}}^2) K_X)$
    Generate $Y_i = Y_{i-1} + \xi$
    Output Y_i
end for

We now explain intuitively why the mutual independence requirement for Z_1, and $(Z_i - Z_{i-1})$ for i from 2 to M is sufficient to achieve our privacy goal in Equation 10.

We rewrite Y_i as $X + Z_1 + \sum_{j=2}^{i} (Z_j - Z_{j-1})$. Since X, Z_1 and $Z_j - Z_{j-1}$ for $j = 2, \ldots, M$ are mutually independent, $Y_i, 2 \le i \le M$ are perturbed observations of Y_1. Intuitively all information in them that are useful for estimating X is inherited from Y_1. As such, given Y_1, $Y_i, 2 \le i \le M$ provides no extra innovative information to improve the estimation accuracy. Similar analysis applies to any subset of Y_1 to Y_M. Hence, Equation 10 is satisfied. This intuition is similar to the explanation for the case study in Section IV.

3) Disadvantages: The main disadvantage of the batch generation approach is that it requires a data owner to foresee all possible trust levels a priori.

This obligatory requirement is not flexible and sometimes impossible to meet. One such scenario for the latter arises in our case study. After the data owner already released a perturbed copy Y_2, a new request for a less distorted copy Y_1 arrives. The sequential generation algorithm cannot handle such requests since the trust level of the new request is lower than the existing one. In today's ever changing world, it is desirable to have technologies that adapt to the dynamics of the society. In our problem setting, generating new perturbed copies on-demand would be a desirable feature.

C. On Demand Generation

As opposed to the batch generation, new perturbed copies are introduced on demand in this second scenario. Since the requests may be arbitrary, the trust levels corresponding to the new copies would be arbitrary as well. The new copies can be either lower or higher than the existing trust levels. We refer this scenario as *on-demand* generation. Achieving the privacy goal in this scenario will give data owners the maximum flexibility in providing MLT-PPDM services.

We assume $L (L < M)$ existing copies of Y_1 to Y_L. We also assume that the data owner, upon requests, generates additional $M - L$ copies of Y_{L+1} to Y_M. Thus there will be M copies in total. Note in this subsection $\sigma_{Z_1}^2$ to $\sigma_{Z_M}^2$ can be in any order. Finally, we define vectors $\mathbb{Z}'$ and $\mathbb{Z}''$ as

$$\mathbb{Z}' = \begin{bmatrix} Z_1 \\ \vdots \\ Z_L \end{bmatrix} \text{ and } \mathbb{Z}'' = \begin{bmatrix} Z_{L+1} \\ \vdots \\ Z_M \end{bmatrix}.$$

According to Theorem 4, the data owner should generate new noise $\mathbb{Z}''$ in such a way that the covariance matrix of $\mathbb{Z} = [\mathbb{Z}'^T \mathbb{Z}''^T]^T$ has corner-wave property, and they are jointly Gaussian.

The desired covariance matrix $K_{\mathbb{Z}}$ can be constructed according to Equation 14 (after properly ordering Z_1 to Z_M according to $\sigma_{Z_1}^2$ to $\sigma_{Z_M}^2$).

It is known that for two Gaussian vectors $\mathbb{Z}'$ and $\mathbb{Z}''$ to be jointly Gaussian, it is sufficient and necessary for the conditional distribution of $\mathbb{Z}''$ given that $\mathbb{Z}'$ takes any value v_1 to be a Gaussian with mean

$$K_{\mathbb{Z}''\mathbb{Z}'} K_{\mathbb{Z}'}^{-1} v_1 \tag{17}$$

and covariance

$$K_{\mathbb{Z}''} - K_{\mathbb{Z}''\mathbb{Z}'} K_{\mathbb{Z}'}^{-1} K_{\mathbb{Z}''\mathbb{Z}'}^T, \tag{18}$$

where $K_{\mathbb{Z}'}$ is the covariance matrix of $\mathbb{Z}'$, $K_{\mathbb{Z}''\mathbb{Z}'}$ is the desired covariance matrix between $\mathbb{Z}''$ and $\mathbb{Z}'$, and $K_{\mathbb{Z}''}$ is the desired covariance matrix of $\mathbb{Z}''$.

Note $K_{\mathbb{Z}'}$ is known to the data owner, and $K_{\mathbb{Z}''\mathbb{Z}'}$ and $K_{\mathbb{Z}''}$ can be extracted from the desired covariance matrix $K_{\mathbb{Z}}$. We turn the above analysis into Algorithm 3 as follows:

Algorithm 3 : On Demand Generation

// Input: X, K_X, $\sigma_{Z_1}^2$ to $\sigma_{Z_M}^2$, and values of $\mathbb{Z}'$: v_1
// Output: New copies $\mathbb{Z}''$
Construct $K_{\mathbb{Z}}$ with K_X and $\sigma_{Z_1}^2$ to $\sigma_{Z_M}^2$, according to Equation 14
Extract $K_{\mathbb{Z}''\mathbb{Z}'}$ and $K_{\mathbb{Z}''}$ from $K_{\mathbb{Z}}$
Construct $K_{\mathbb{Z}'}$ with K_X and $\sigma_{Z_1}^2$ to $\sigma_{Z_L}^2$, according to Equation 14
Generate $\mathbb{Z}''$ as a Gaussian with mean and variance in Equation 17 and 18, respectively
for i from $L+1$ to M **do**
    Generate $Y_i = X + Z_i$
    Output Y_i
end for

Algorithm 3 requires $O(M^2 \cdot N^2)$ memory to store the covariance matrix $K_{\mathbb{Z}}$. Table II compares the three proposed algorithms.

TABLE II
COMPARISON OF APPLICABILITIES AND REQUIRED MEMORY SIZE OF THREE PROPOSED ALGORITHMS.

	Batch Generation	On-demand Generation	Required memory size
Algorithm 1	✓		$O(M^2N^2)$
Algorithm 2	✓		$O(N^2)$
Algorithm 3	✓	✓	$O(M^2N^2)$

VI. EXPERIMENTS

A. Methodology and Settings

We design two experiments to explore answers to the following questions numerically:

- How severe can LLSE-based diversity attacks be, given perturbed copies at different trust levels are generated independently?
- How effective is our proposed scheme against LLSE-based diversity attacks?
- How does an adversary's knowledge affect the power of such attacks?

We use synthetic data for ease of carrying out the experiments and evaluating their performance in a fully controlled manner. Our approach generates a mean vector μ_X and a covariance matrix K_X, then the synthetic data X based on μ_X and K_X. We use a process similar to the one in [4] to generate a covariance matrix.

To generate perturbed copies Y_i at different trust levels i, we generate Gaussian noise Z_i according to $N(0, \sigma^2_{Z_i} K_X)$, and add them to X . The constant $\sigma^2_{Z_i}$ represents the perturbation magnitude determined by the data owner according to the trust level i. The noise for different trust levels are generated either independently, or in a properly correlated manner following our proposed solution in Section V.

Data miners can access one or more perturbed copies Y_i, either according to application scenario setting or by collusion among themselves. Recall our assumption that data miners perform joint LLSE estimation to reconstruct X. We study two classes of data miners with different knowledge about the original data and noise:

- the first class of adversaries have perfect knowledge, i.e., the exact values of μ_X, K_X and $\sigma^2_{Z_i}$ for every trust level i;
- the second class of adversaries have partial knowledge, i.e., the exact values of $\sigma^2_{Z_i}$ for every trust level i but not μ_X and K_X.

To perform LLSE estimation, data miners with partial knowledge estimate μ_X and K_X using their perturbed copies. For each Y_i, its mean is simply μ_X, and its covariance matrix is $(1 + \sigma^2_{Z_i})K_X$. Knowing the exact values of $\sigma^2_{Z_i}$, a data miner can estimate μ_X and K_X using the sample mean and sample covariance matrix of Y_i. Accuracy of such estimation depends on the sample size; the larger the sample size, the more accurate the estimation on μ_X and K_X can be.

In our experiments, we use normalized estimation error as the performance metric. For LLSE estimate of X based on $\mathbb{Y}$, i.e. $\hat{X}(\mathbb{Y})$, we define its normalized estimation error as

$$\frac{\mathcal{D}(X, \hat{X}(\mathbb{Y}))}{K_X}.$$

It takes values between 0 and 1. The smaller it is, the more accurate the LLSE estimation is. It generally decreases as more perturbed copies are used in the LLSE estimation.

B. Experiment 1: Independent Noise

We assume data miners can access M perturbed copies Y_i, $1 \le i \le M$. Each Y_i is for a different trust level with the corresponding $\sigma^2_{Z_i}$, $1 \le i \le M$ randomly generated in $[0.25, 2]$. This implies that the perturbation magnitude at a trust level is at most twice of the data variance. This setting represents the most severe attacking scenario where data miners jointly estimate X using all available M perturbed copies. We evaluate the corresponding normalized estimation error for the two attacker classes with perfect and partial knowledge.

We assume that data miners with partial knowledge estimate μ_X and K_X with different sample sizes. In particular, we assume that they have $100N^2$, $200N^2$ and $300N^2$ samples, where N^2 is the number of entries in K_X.

Figure 1 shows the normalized estimation errors as a function of the number of perturbed copies. The results of the experiments clearly show that the diversity gain in joint estimation helps reduce the normalized estimation error dramatically. When data miners have perfect knowledge, the normalized estimation error decreases monotonically as M increases. This trend indicates a perfect reconstruction of X when M goes to infinity. It also confirms empirically our statement of Theorem 3.

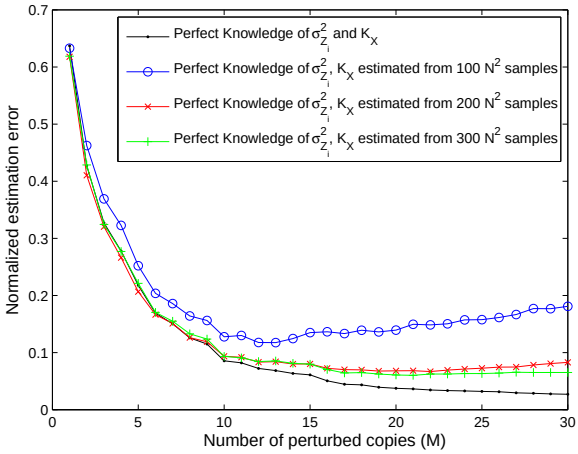

Fig. 1. Average estimation error as a function of the number of independently generated perturbed copies.

Contrarily, the curve flattens and even slightly increases as M becomes large for the cases where the attackers with partial knowledge. This is because the estimation error depends not only on the number of perturbed copies, but also on the precision of μ_X and K_X.

With a small M, the diversity gain in combining different perturbed copies dominates, resulting in decreasing estimation error. However, such gain diminishes after obtaining enough copies. Meanwhile, the estimation based on inaccurately estimated m_X and K_X is no longer optimal. Consequently, the estimation accuracy no longer improves as M increases. Figure 1 also shows that adversaries having more samples perform better in estimating μ_X and K_X, resulting in improved overall accuracy.

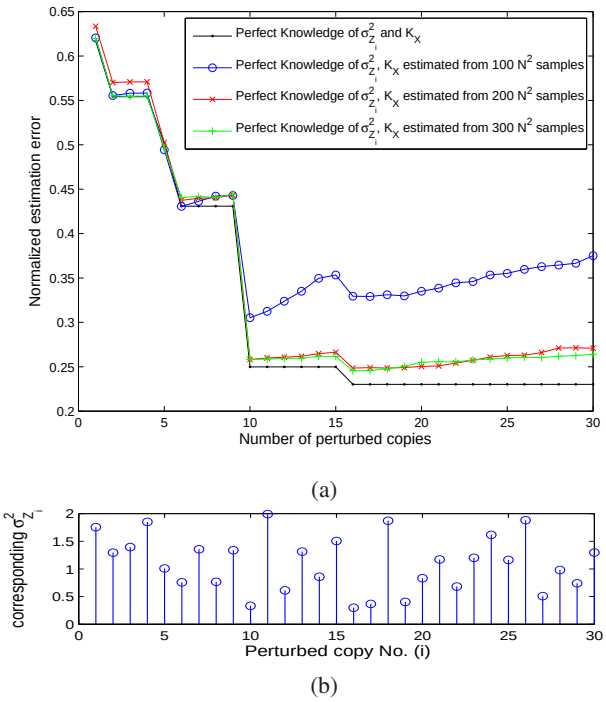

Fig. 2. (a) Average estimation error as a function of the number of perturbed copies generated using our proposed Algorithm 3. (b) Perturbation magnitude $\sigma^2_{Z_i}$ as a function of perturbed copy number i.

C. Experiment 2: Privacy Goal Achieved

We generate M perturbed copies one by one upon requests, using our proposed algorithm A3. Each request is at a different trust level with corresponding $\sigma^2_{Z_i}$ randomly generated in $[0.25, 2]$. Figure 2.(b) shows $\sigma^2_{Z_i}$ as a function of perturbed copy number i. A new copy is generated using our proposed Algorithm 3.

We again study the most severe attacking scenario where data miners jointly estimate X using all available M perturbed copies. Since the perturbed copies are released one by one, the number of available perturbed copies also increases one by one.

We first consider the case where data miners have perfect knowledge. Comparing Figure 2.(a) and 2.(b), we find that the estimation error drops only when a perturbed copy with minimum perturbation magnitude so far becomes available. Our observation implies that the joint estimation based on all existing copies is only as good as the estimation based on the copy with the minimum privacy, and there is no diversity gain in performing the LLSE estimation jointly. Moreover, we have verified that the estimation error matches our analytical result in Theorem 4.

For the case where data miners have only partial knowledge, we observe that the LLSE estimation error is strictly larger than the case where data miners have perfect knowledge, which is consistent with our observations in Experiment 1.

In summary, the privacy goal in Section III-D is achieved in this most severe attacking scenario. We have also verified that the goal is also achieved in other attacking scenario where adversaries have the access to an arbitrary subset of the M perturbed copies.

VII. Discussion and Future Work

In this work, we expand the scope of additive perturbation based PPDM to multi-level trust (MLT), by relaxing an implicit assumption of a single-level trust in exiting work. MLT-PPDM allows data owners to generate differently perturbed copies of its data for different trust levels.

The key challenge lies in preventing the data miners from combining copies at different trust levels to jointly reconstruct the original data more accurate than what is allowed by the data owner.

We address this challenge by properly correlating noise across copies at different trust levels. We prove that if we design the noise covariance matrix to have corner-wave property, then data miners will have no diversity gain in their joint reconstruction of the original data. We verify our claim and demonstrate the effectiveness of our solution through numerical evaluation.

Last but not the least, our solution allows data owners to generate perturbed copies of its data at arbitrary trust levels on-demand. This property offers the data owner maximum flexibility.

We believe that multi-level trust privacy preserving data mining can find many applications. Our work takes the initial step to enable MLT-PPDM services. Many interesting and important directions are worth exploring. For example, it is not clear how to expand the scope of other approaches in the area of partial information hiding, such as random rotation based data perturbation, k-anonymity , and retention replacement, to multi-level trust. It is also of great interest to extend our approach to handle evolving data streams.

Embraced with the assumption that adversaries carry out only linear attacks, our work does not take into account the adversaries that apply nonlinear techniques to derive the original data. This is the limitation of our work, as well as most existing work on perturbation based PPDM. Studying the MLT-PPDM problem under a relaxed setting where adversaries may also carry out nonlinear attacks is certainly an interesting future direction.

References

[1] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proc. of the 20th ACM Symposium on Principles of Database Systems*, Santa Barbara, California, May 2001, pp. 247–255.

[2] R. Agrawal and R. Srikant, "Privacy preserving data mining," in *ACM SIGMOD Conference on Management of Data*, Dallas, Texas, May 2000, pp. 439–450.

[3] K. Chen and L. Liu, "Privacy preserving data classification with rotation perturbation," in *Fifth IEEE International Conference on Data Mining*, 2005.

[4] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proc. of the 24th ACM SIGMOD Conference on Management of Data*, 2005.

[5] F. Li, J. Sun, S. Papadimitriou, G. Mihaila, and I. Stanoi, "Hiding in the crowd: Privacy preservation on evolving streams through correlation tracking," in *Proc. of the 23th Int'l Conference on Data Engineering*, 2007.

[6] 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, pp. 92–106, January 2006.

[7] S. Papadimitriou, F. Li, G. Kollios, and P. S. Yu, "Time series compressibility and privacy," in *33th Int'l Conf. on Very Large Databases (VLDB), Vienna, Austria*, 2007.

[8] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *CRYPTO*, 2000, pp. 36–54. [Online]. Available: citeseer.nj.nec.com/lindell00privacy.html

[9] J. Vaidya and C. W. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proc. of the 8th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, July 2002. [Online]. Available: citeseer.nj.nec.com/492031.html

[10] O. Goldreich, "Secure multi-party computation," Final (incomplete) draft, version 1.4, 2002.

[11] J. Vaidya and C. Clifton, "Privacy-preserving k-means clustering over vertically partitioned data," in *Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[12] A. W.-C. Fu, R. C.-W. Wong, and K. Wang, "Privacy-preserving frequent pattern mining across private databases," in *Proc. of the 5th Int'l Conference on Data Mining (ICDM'05)*, 2005.

[13] B. Bhattacharjee, N. Abe, K. Goldman, B. Zadrozny, V. R. Chillakuru, M. del Carpio, and C. Apte, "Using secure coprocessors for privacy preserving collaborative data mining and analysis," in *Proc. of the 2nd international workshop on Data management on new hardware*, 2006.

[14] C. C. Aggarwal and P. S. Yu, "A condensation approach to privacy preserving data mining," *Lecture Notes in Computer Science*, vol. 2992/2004, pp. 183–199, 2004.

[15] E. Bertino, B. C. Ooi, Y. Yang, and R. H. Deng, "Privacy and ownership preserving of outsourced medical data," in *Proc. of the 21st Int'l Conference on Data Engineering (ICDE'05)*, April 2005.

[16] D. Kifer and J. E. Gehrke, "Injecting utility into anonymized datasets," in *Proc. of the 25th ACM SIGMOD Conference on Management of Data*, 2006.

[17] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in *Proc. of the 22nd IEEE International Conference on Data Engineering*, 2006.

[18] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS)*, vol. 10, 2002.

[19] X. Xiao and Y. Tao, "Personalized privacy preservation," in *Proc. of the 25th ACM SIGMOD Conference on Management of Data*, 2006.

[20] R. Agrawal, R. Srikant, and D. Thomas, "Privacy preserving olap," in *Proc. of the 24th ACM SIGMOD Conference on Management of Data*, 2005.

[21] W. Du and Z. Zhan, "Using randomized response techniques for privacy-preserving data mining," in *Proc. of the 9th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, August 2003.

[22] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," in *Proc. of the 8th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, July 2002.

[23] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Proc. of the Third IEEE International Conference on Data Mining*, 2003.

[24] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private databases," in *Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data*, San Diego, CA, June 2003.

[25] R. Agrawal, D. Asonov, M. Kantarcioglu, and Y. Li, "Sovereign joins," in *Proceedings of the 22nd International Conference on Data Engineering*, Atlanta, Georgia, April 2006.

[26] C. Clifton, M. Kantarcioglu, X. Lin, J. Vaidya, and M. Zhu, "Tools for privacy preserving distributed data mining," *SIGKDD Explorations*, vol. 4, no. 2, pp. 28–34, Jan. 2003. [Online]. Available: citeseer.ist.psu.edu/637102.html

[27] B. A. Huberman, M. Franklin, and T. Hogg, "Enhancing privacy and trust in electronic communities," in *Proc. of the 1st ACM Conference on Electronic Commerce*, Denver, Colorado, November 1999, pp. 78–86.

[28] M. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Advances in Cryptology — EUROCRYPT 2004*. Springer-Verlag, May 2004, pp. 1–19. [Online]. Available: citeseer.ist.psu.edu/article/freedman04efficient.html

[29] L. Kissner and D. Song, "Privacy-preserving set operations," in *Advances in Cryptology — CRYPTO 2005*. [Online]. Available: citeseer.ist.psu.edu/739924.html

[30] A. Iliev and S. Smith, "More efficient secure function evaluation using tiny trusted third parties," Department of Computer Science, Dartmouth University, Dartmouth Computer Science Technical Report TR2005-551, 2005.

APPENDIX

A. Proof of Theorem 1

Proof: We first prove the if part of the theorem. From the covariance matrix of Z_1 and Z_2, we know that $E[Z_1 Z_2] = \sigma_1^2$. Therefore,

$$E[Z_1(Z_2 - Z_1)] = E[Z_1 Z_2] - E[Z_1^2] = \sigma_1^2 - \sigma_1^2 = 0, \quad (19)$$

suggesting that Z_1 and $Z_2 - Z_1$ are linearly independent.

Meanwhile, by definition of jointly Gaussian, $Z_2 - Z_1$ is also a Gaussian random variable. For Gaussian variables Z_1 and $Z_2 - Z_1$, linear independence implies independence.

We now prove the only if part of the theorem. We observe that $Z_2 = Z_1 + (Z_2 - Z_1)$ is sum of two independent Gaussian random variables. Thus, Z_2 and Z_1 are jointly Gaussian by definition, and we also have $E[Z_2 Z_1] = E[Z_1 Z_2] = \sigma_1^2$. It follows that their covariance matrix is as follows:

$$\begin{bmatrix} \sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_2^2 \end{bmatrix}.$$

■

B. Proof of Theorem 2

Proof: By Theorem 5, if Z_1 and Z_2 satisfy that Z_1 and $Z_2 - Z_1$ are independent, then their covariance matrix, denoted by K_C, must be given by

$$K_C = \begin{bmatrix} \sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_2^2 \end{bmatrix}.$$

Based on Y_1, the LLSE estimation of X has an estimation error of

$$\sigma_X^2 - \frac{\sigma_X^4}{\sigma_X^2 + \sigma_1^2} = \frac{\sigma_X^2}{1 + \sigma_X^2/\sigma_1^2}, \quad (20)$$

which can be computed using Equation 8.

Similarly, based on both Y_1 and Y_2, the LLSE estimation of X has an estimation error of

$$\left[\frac{1}{\sigma_X^2} + \begin{bmatrix} 1 & 1 \end{bmatrix} K_C^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right]^{-1}.$$

After simplification, the above estimation error is exactly the one shown in Equation 20. Thus, Equation 11 holds. ■

C. Proof of Theorem 3

Proof: If $Z_i, 1 \leq i \leq M$ are independent to each other, then $K_{\mathbb{Z}}$ is given by

$$K_{\mathbb{Z}} = \begin{bmatrix} \sigma_{Z_1}^2 K_X & 0 & \cdots & 0 \\ 0 & \sigma_{Z_2}^2 K_X & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{Z_M}^2 K_X \end{bmatrix}.$$

By Equation 7, the estimation errors are the diagonal terms of the following matrix

$$\left[K_X^{-1} + H^T K_{\mathbb{Z}}^{-1} H\right]^{-1} = \left(1 + \sum_{i=1}^{M} \frac{1}{\sigma_{Z_i}^2}\right)^{-1} K_X.$$

■

D. Proof of Theorem 4

Proof: By the definition of distortion and the result shown in Equation 7, we have

$$\mathcal{D}(X, \hat{X}(\mathbb{Y})) = \frac{1}{N} Tr\left(\left[K_X^{-1} + H^T K_{\mathbb{Z}}^{-1} H\right]^{-1}\right),$$

and for $i = 1, \ldots, M$,

$$\mathcal{D}(X, \hat{X}(Y_i)) = \frac{\sigma_{Z_i}^2}{1 + \sigma_{Z_i}^2} \frac{1}{N} Tr\left(K_X\right).$$

Two observations can be made for the above two equations. First, we must have $\mathcal{D}(X, \hat{X}(Y_i)) < \mathcal{D}(X, \hat{X}(Y_{i+1}))$ due to the assumption on σ_{Z_i} in Equation 16, and

$$\min_{i=1,\ldots,M} \mathcal{D}(X, \hat{X}(Y_i)) = \mathcal{D}(X, \hat{X}(Y_1)) = \frac{\sigma_{Z_1}^2}{\sigma_{Z_1}^2 + 1} \frac{Tr(K_X)}{N}.$$

Second, the proof is complete if we can show that

$$H^T K_{\mathbb{Z}}^{-1} H = K_{Z_1}^{-1}. \tag{21}$$

This obviously holds for the case of $M = 1$.

Rewrite $K_{\mathbb{Z}}$ as the following form

$$K_{\mathbb{Z}} = \begin{bmatrix} K_{Z_1} & K_{Z_1} & \cdots & K_{Z_1} \\ K_{Z_1} & \frac{\sigma_{Z_2}^2}{\sigma_{Z_1}^2} K_{Z_1} & \cdots & \frac{\sigma_{Z_2}^2}{\sigma_{Z_1}^2} K_{Z_1} \\ \vdots & \vdots & \ddots & \vdots \\ K_{Z_1} & \frac{\sigma_{Z_2}^2}{\sigma_{Z_1}^2} K_{Z_1} & \cdots & \frac{\sigma_{Z_M}^2}{\sigma_{Z_1}^2} K_{Z_1} \end{bmatrix}.$$

We find its inverse following a standard process. We perform row operation to the matrix $[K_{\mathbb{Z}} \mid I]$ until it has the form $[I \mid A]$. Then matrix A is $K_{\mathbb{Z}}^{-1}$. Note the structure of $K_{\mathbb{Z}}$ makes this process pretty straightforward and easy.

Following above process, we find the expression of $K_{\mathbb{Z}}^{-1}$ for the case of $M \geq 2$ as follows:

$$\begin{bmatrix} \frac{c_1 \sigma_{Z_2}^2}{\sigma_{Z_1}^2} K_{Z_1}^{-1} & -c_1 K_{Z_1}^{-1} & 0 & \cdots & 0 \\ -c_1 K_{Z_1}^{-1} & (c_1 + c_2) K_{Z_1}^{-1} & -c_2 K_{Z_1}^{-1} & \cdots & 0 \\ 0 & -c_2 K_{Z_1}^{-1} & (c_3 + c_2) K_{Z_1}^{-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & c_{M-1} K_{Z_1}^{-1} \end{bmatrix},$$

where

$$c_i = \frac{1}{\sigma_{Z_{i+1}}^2 / \sigma_{Z_1}^2 - \sigma_{Z_i}^2 / \sigma_{Z_1}^2}, \quad 1 \leq i \leq M-1.$$

It is straightforward to verify the product of $K_{\mathbb{Z}}$ and the above matrix is an identity matrix.

Noticing that $K_{\mathbb{Z}}^{-1}$ only have non-zero entries in the main diagonal and two adjacent diagonals, and that its column and row sums are zero except the first row and column, we have

$$H^T K_{\mathbb{Z}}^{-1} H = \begin{bmatrix} K_{Z_1}^{-1} & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} I_N \\ \vdots \\ I_N \end{bmatrix} = K_{Z_1}^{-1},$$

and the proof is complete. ■

E. Proof of Theorem 5

Proof: We first prove the if part of the theorem. Since Z_1 to Z_M are jointly Gaussian variables, Z_1, and $(Z_i - Z_{i-1})$ for are also jointly Gaussian variables. This is because any linear combination of them is simply another linear combination of Z_1 to Z_M, and is thus a Gaussian. For jointly Gaussian variables, they are mutually independent if their covariance matrix is a diagonal matrix. This can be easily verified by evaluating their joint distribution.

From the covariance matrix of $\mathbb{Z}$, we know that for $j > i$, $E[Z_i Z_j^T] = K_{Z_i}$. For $2 \leq i < j \leq M$, we have

$$\begin{aligned} & E[(Z_i - Z_{i-1})(Z_j - Z_{j-1})^T] \\ = \; & E[Z_i Z_j^T] - E[Z_i Z_{j-1}^T] - E[Z_{i-1} Z_j^T] + E[Z_{i-1} Z_{j-1}^T] \\ = \; & K_{Z_i} - K_{Z_i} - K_{Z_{i-1}} + K_{Z_{i-1}} = 0. \end{aligned}$$

We also have for $2 \leq i \leq M$,

$$\begin{aligned} E[Z_1 (Z_i - Z_{i-1})^T] &= E[Z_1 Z_i^T] - E[Z_1 Z_{i-1}]^T \\ &= K_{Z_1} - K_{Z_1} = 0. \end{aligned}$$

As such, we must have the covariance matrix of Z_1, and $(Z_i - Z_{i-1})$ for to be diagonal, and they are mutually independent.

We now prove the only if part of the theorem. Since Z_1, and $(Z_i - Z_{i-1})$ for i from 2 to M are mutually independent Gaussian variables, we must have Z_1 to Z_M to be jointly Gaussian. This is because each of them is simply a linear combination of independent Gaussian variables.

We also have for $j > i$,

$$\begin{aligned} E[Z_i Z_j^T] &= E\left[Z_i \left(Z_i + \sum_{l=i+1}^{j} (Z_l - Z_{l-1})\right)^T\right] \\ &= E[Z_i Z_i^T] + \sum_{l=i+1}^{j} E[Z_i (Z_l - Z_{l-1})^T] \\ &= K_{Z_i}. \end{aligned}$$

It follows that $K_{\mathbb{Z}}$ must have the form as in Equation 14. ■