Enabling Multilevel Trust in Privacy Preserving Data Mining

Yaping Li, Minghua Chen, Qiwei Li, and Wei Zhang

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 are published. Previous solutions of this approach are limited in their tacit assumption of single-level trust on data miners. In this work, we relax this assumption and expand the scope of perturbation-based PPDM to Multilevel 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 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 services. 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 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 copy 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.

Index Terms—Privacy preserving data mining, multilevel trust, random perturbation.

1 Introduction

Data perturbation, a widely employed and accepted Privacy Preserving Data Mining (PPDM) approach, tacitly assumes single-level trust on data miners. This approach introduces uncertainty about individual values before data are 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 a 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 cannot reconstruct the original data *more* accurately when it uses both copies than when it uses only the leaked internal copy.

This new dimension of *Multilevel 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 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 a *diversity attack*. It includes the colluding attack scenario where adversaries combine their copies to mount an attack; 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 in solving 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 Gaussian noise is added to the original data with *arbitrary* distribution, 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.

- *Y. Li, M. Chen, and W. Zhang are with the Department of Information Engineering, Ho Sin Hang Engineering Building, The Chinese University of Hong Kong, Shatin, Hong Kong. E-mail: {yaping, minghua, zw009}@ie.cuhk.edu.hk.*
- *Q. Li is with the Department of Statistics, Rice University, 6100 Main Street, Houston, Texas 77005. E-mail: qiwei.li@rice.edu.*

Manuscript received 4 Feb. 2010; revised 31 Oct. 2010; accepted 25 Mar. 2011; published online 26 May 2011.
Recommended for acceptance by C. Clifton.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2010-02-0067.
Digital Object Identifier no. 10.1109/TKDE.2011.124.

1.1 Contributions

We make the following contributions:

- We expand the scope of perturbation-based PPDM to multilevel trust, by relaxing the implicit assumption of single-level trust in existing 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 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 explain through a case study in Section 4.
- 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 experiments on real data.
- 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.

1.2 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 permit useful mining operations over the modified data. This approach can be further divided into three categories: 1) k-anonymity [14], [15], [16], [17], [18], [19], 2) 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 3) data perturbation (which introduces uncertainty about individual values before data are published) [1], [2], [3], [4], [5], [6], [7], [23].

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] and [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].

Our work does not reanonymize a data set after it is updated with insertions and/or deletions, which is a topic studied by the authors in [31], [32], [33], [34]. Instead, we study anonymizing the same data set at multiple trust levels. The two problems are orthogonal.

An earlier version of this paper appeared in [35] and initiated the topic of MLT-PPDM. Recently, Xiao et al. proposed an algorithm of multilevel uniform perturbation [36]. Our paper differs from [36] in three main aspects. First, the two papers address different problems and tackle the problems under different privacy measures. We propose multilevel privacy preserving for additive Gaussian noise perturbation, and use a measure based on how closely the original values can be reconstructed from the perturbed data [2], [4], [5]. While [36] presents an algorithm of multilevel uniform perturbation, and studies its performance using the $\rho_1 - \rho_2$ privacy measure [37]. As a result, neither the solution in [36] can be easily applied to the problem in this paper nor the solution in this paper can be directly applied to the problem in [36]. Second, based on Gaussian noise perturbation, the solution in this paper is more suitable for high-dimensional data, as compared to that in [36] based on uniform perturbation [38]. Third, We present several non-trivial theoretical results. We discuss reconstruction errors under independence noise, analyze the security of our scheme when collusion occurs, and study the computational complexities based on Kroneckor product. These results provide fundamental insights into the problem.

1.3 Paper Layout

The rest of the paper is organized as follows: we go over preliminaries in Section 2. We formulate the problem, and define our privacy goal in Section 3. In Section 4, 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 5, we formally present our solution, and prove that it achieves our privacy goal. Algorithms that target different scenarios are also proposed, and their complexities are studied. We carry out extensive experiments on real data in Section 6 to verify our theoretical analysis. Section 7 concludes the paper.

2 Preliminaries

2.1 Jointly Gaussian

In this paper, we focus on perturbing data by additive Gaussian noise [1], [2], [4], [5], [7], i.e., the added noises are jointly Gaussian.[1]

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.[2] 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.

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.

If multiple random variables are jointly Gaussian, then conditional on a subset of them, the remaining variables are still jointly Gaussian. Specifically, partition a jointly Gaussian vector $\mathbb{G}$ as

$$\mathbb{G} = \begin{bmatrix} \mathbb{G}_1 \\ \mathbb{G}_2 \end{bmatrix},$$

and

$$\mu_{\mathbb{G}} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, K_{\mathbb{G}} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix},$$

accordingly. Then the distribution of $\mathbb{G}_2$ given $\mathbb{G}_1 = v_1$ is also a jointly Gaussian with mean $\mu_2 + K_{21}K_{11}^{-1}(v_1 - \mu_1)$ and covariance matrix $K_{22} - K_{21}K_{11}^{-1}K_{21}^T$ [39]. This is a key property of jointly Gaussian variables. We utilize this property in Section 5.3.

2.2 Additive Perturbation

The single-level trust PPDM problem via data perturbation has been widely studied in the 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 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 jth 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 semidefinite matrix given by

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

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 semidefinite matrix given by

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

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. Huang et al. [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.

2.3 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 [39].

1. Note that we do not make any assumptions about the distribution of the data.

2. Two random variables are independent if knowing the value of one yields no knowledge about that of the other. Mathematically, two random variables G_1 and G_2 are independent if, for any values g_1 and g_2, $f_{G_1,G_2}(g_1, g_2) = f_{G_1}(g_1) f_{G_2}(g_2)$, where $f_{G_1,G_2}(g_1, g_2)$ is the joint probability density function of G_1 and G_2, and $f_{G_1}(g_1)$ and $f_{G_2}(g_2)$ are the probability density functions of G_1 and G_2, respectively. Generally, random variables G_1 through G_L are mutually independent if, for any values g_1 through g_L, $f_{G_1,\ldots,G_L}(g_1, \ldots, g_L) = f_{G_1}(g_1) \cdots f_{G_L}(g_L)$.

The LLSE estimate of X given Y, denoted by $\hat{X}(Y)$, is (see Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.124, for the deduction)

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

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

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

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.

2.4 Kronecker Product

In the MLT-PPDM problem, the covariance matrix of noises can be written as the Kronecker product [40] of two matrices. In this paper, we explore the properties of the Kronecker product for efficient computation.

The Kronecker product [40] is a binary matrix operator that maps two matrices of arbitrary dimensions into a larger matrix with a special block structure. Given an $n \times m$ matrix A and $p \times q$ matrix B, where

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix},$$

their Kronecker product, denoted as $A \otimes B$, is an $np \times mq$ matrix with the block structure

$$\begin{bmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & a_{nm}B. \end{bmatrix}.$$

We list several properties of Kronecker product that will be used later. Assume that A, B, C, and D are matrices and their dimensions are appropriate for the computation in each property, we have

1. $(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha(A \otimes B)$, where $\alpha \in \mathbb{R}$;
2. $(A \otimes B)^T = A^T \otimes B^T$;
3. $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$;
4. $(A \otimes B)(C \otimes D) = AC \otimes BD$;
5. $\mathrm{vec(ABC)} = (\mathrm{C^T} \otimes \mathrm{A})\mathrm{vec(B)}$, where $\mathrm{vec}(\cdot)$ denotes the vectorization of a matrix formed by stacking the columns of the matrix into a single column vector.

3 Problem Formulation

In this section, we present the problem settings, describe our threat model, state our privacy goal, and identify the design space. Table 1 lists the key notations used in the paper.

TABLE 1
Key Notations

Notation	Definition
X	original data
Y_i	perturbed copy of X of trust level i
Z_i	noise added to X to generate Y_i
M	number of trust levels
N	number of attributes in X
$\mathbb{Y}$	a vector of all M perturbed copies
$\mathbb{Z}$	a vector of noise 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 $\mathbb{Z}$

3.1 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 multilevel trust setting, data miners at higher trust levels can access less perturbed copies. Such less perturbed copies are not accessible by data miners at lower trust levels. In some scenarios, such as the motivating example we give at the beginning of Section 1, data miners at higher trust levels may also have 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 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 2.2. 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.

It is true that the data owner may consider to release only the mean and covariance of the original data. We remark that simply releasing the mean and covariance does not provide the same utility as the perturbed data. For many real applications, knowing only the mean and covariance may not be sufficient to apply data mining techniques, such as clustering, principal component analysis, and classification [6]. By using random perturbation to release the data set, the data owner allows the data miner to exploit more statistical information without releasing the exact values of sensitive attributes [1], [2].

Let $\mathbb{Y} = [Y_1^T, \ldots, Y_M^T]^T$ be the vector of all perturbed copies $Y_i (1 \le i \le M)$. 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 $Z_i, 1 \le i \le M$ 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 the same correlations as the attributes themselves, otherwise Z_i can be easily filtered out [4]. As such, we have

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

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.

3.2 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 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.

The reconstruction accuracy depends heavily 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 $K_{\mathbb{Z}}$. Note that the adversaries with less knowledge are weaker than the ones we study in this paper.

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 works in perturbation-based PPDM, such as [4] and [5], make this assumption of linear estimation. See [7] for a comprehensive review.

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

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

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 (6), we can compute the covariance matrix as follows:

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

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

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

3.3 Definitions

3.3.1 Distortion

To facilitate future discussion on privacy, we define the concept of perturbation $\mathcal{D}$ between two data sets as the average expected square difference between them. For example, the distortion between the original data X and the perturbed copy Y as defined in Section 2.2 is given by

$$\mathcal{D}(X, Y) = \frac{1}{N} \sum_{j=1}^{N} E[(y_j - x_j)^2] \ge 0.$$

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

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

3.3.2 Privacy under Single-Level Trust Setting

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.

A more perturbed copy of the data does not necessarily have more privacy 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 $\hat{X}(Y)$. A larger distortion hides the original values better (and thus preserves more privacy), so we refer to a perturbed data Y_2 to preserve *more privacy* than Y_1 with respect to X if and only if $\mathcal{D}(X, \hat{X}(Y_2)) > \mathcal{D}(X, \hat{X}(Y_1))$.

3.3.3 Privacy under Multilevel Trust Setting

We now define privacy for the multilevel 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 $\mathbb{Y}$. 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 $\hat{X}(\mathbb{Y})$.

3.4 Privacy Goal and Design Space

In a 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., $Y_i (1 \le i \le M)$, each for a trust level i. The assumption of M is for ease of analysis. 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 (8), we express the privacy of Y_i, i.e., $\mathcal{D}(X, \hat{X}(Y_i))$, as follows:

$$\begin{aligned}&\mathcal{D}(X,\hat{X}(Y_i))\\&=\frac{1}{N}Tr(E[(\hat{X}(Y_i)-X)(\hat{X}(Y_i)-X)^T])\\&=\frac{\sigma^2_{Z_i}}{\sigma^2_{Z_i}+1}\frac{1}{N}Tr(K_X),\end{aligned}\tag{9}$$

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

The data owner can easily control the privacy of an individual copy Y_i by setting $\sigma^2_{Z_i}$ 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 4.2.1.

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 copy.

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

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

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 least privacy 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 $Z_i, 1 \le i \le M$ must follow $N(0, \sigma^2_{Z_i}K_X)$. In the rest of the paper, we show by properly correlating noise $Z_i, 1 \le i \le M$, the desired privacy goal can be achieved.

4 Case Study

In this section, we study a basic case corresponding to the motivating example we described at the beginning of Section 1. 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 (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.

4.1 An Illustrative Case

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

$$Y_2 = X + Z_2.$$

Denote the variance of X as σ^2_X, and the Gaussian noise $Z_2 \sim N(0, \sigma^2_2\sigma^2_X)$ is independent of X.

The data owner now wishes to produce another perturbed copy Y_1. It generates Gaussian noise $Z_1 \sim N(0, \sigma^2_1\sigma^2_X)$, 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 $\sigma^2_2 > \sigma^2_1$ so that Y_1 is less perturbed than Y_2.

The privacy goal in (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 $\min(\mathcal{D}(X,\hat{X}(Y_1)),\mathcal{D}(X,\hat{X}(Y_2)))$ can be simplified to $\mathcal{D}(X,\hat{X}(Y_1))$, i.e., the less perturbed copy gives better estimate.

4.2 Two False Starts

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

4.2.1 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 of the original data, but also independent of 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 utilizing (7) shows that

$$\mathcal{D}(X,\hat{X}(Y_1,Y_2)) = \frac{\sigma^2_X}{1+1/\sigma^2_1+1/\sigma^2_2}.$$

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

$$\mathcal{D}(X,\hat{X}(Y_i)) = \frac{\sigma^2_X}{1+1/\sigma^2_i},$$

following (8). Thus, (11) is not satisfied and the desired privacy goal is not achieved.

Example. Assume that the original data set has single attribute data X with mean $\mu_X = 10$ and variance $\sigma^2_X = 1$. The data owner releases perturbed copies $Y_1 = X + Z_1$ and $Y_2 = X + Z_2$ of two (sensitive) values $X = [9, 11]^T$ to Alice and Bob with different trust levels $\sigma^2_1 = 1$ and $\sigma^2_2 = 4$, respectively.

Alice reconstructs the data values using (4), and obtains $\hat{X}(Y_1) = [9.5, 10.5]^T + 0.5Z_1$. The average estimation error is

$$\frac{1}{2}E[(\hat{X}-X)^T(\hat{X}-X)] = 0.125E\left[Z_1^TZ_1\right] + 0.25 = 0.5.$$

Bob reconstructs the data values using (4), and obtains $\hat{X}(Y_1) = [9.8, 10.2]^T + 0.2Z_2$. The average estimation error is

$$\frac{1}{2}E[(\hat{X}-X)^T(\hat{X}-X)] = 0.02E[Z_2^T Z_2] + 0.64 = 0.8.$$

Assume that Y_1 and Y_2 are generated independently. The reconstructed data after the collusion between Alice and Bob using (6) are $\hat{X}(\mathbb{Y}) = [85, 95]^T/9 + 4Z_1/9 + Z_2/9$. The average estimation error is

$$\frac{1}{2}E[(\hat{X}-X)^T(\hat{X}-X)] = \frac{8}{81}Z_1^T Z_1 + \frac{1}{162}Z_2^T Z_2 + \frac{16}{81} = \frac{4}{9}.$$

Thus, the collusion results in a smaller error.

Intuitively, this is because the two copies of the data are generated independently, each containing some innovative information of the original data that are 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.

4.2.2 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 $Z_1 \sim N(0, \sigma_1^2\sigma_X^2)$. However, $Y_1 = X + Z_1$ again fails to achieve the privacy goal.

To see this, notice that the adversaries who have 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 (11) is not satisfied.

4.3 Proposed Solution

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

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

$$Y_2 = Y_1 + (Z_2 - Z_1). \tag{12}$$

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 (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 $Z_1 \sim N(0, \sigma_1^2\sigma_X^2)$, $Z_2 \sim N(0, \sigma_2^2\sigma_X^2)$, 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*

$$\begin{bmatrix} \sigma_1^2\sigma_X^2 & \sigma_1^2\sigma_X^2 \\ \sigma_1^2\sigma_X^2 & \sigma_2^2\sigma_X^2 \end{bmatrix}. \tag{13}$$

Proof. Refer to Appendix B, available in the online supplemental material. □

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

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

Proof. Refer to Appendix C, available in the online supplemental material. □

Example. We now revisit the example in Section 4.2.1 to show that collusion does not improve estimation accuracy in our scheme. Assume that Y_1 and Y_2 are generated following the proposed solution, i.e., Z_1 and Z_2 are jointly Gaussian and their covariance matrix is $[\begin{smallmatrix}1 & 1\\ 1 & 4\end{smallmatrix}]$. The reconstructed data after the collusion between Alice and Bob using (6) are $\hat{X}(Y_1) = [9.5, 10.5]^T + 0.5Z_1$. The average estimation error is

$$\frac{1}{2}E[(\hat{X}-X)^T(\hat{X}-X)] = 0.125E[Z_1^T Z_1] + 0.25 = 0.5.$$

This error of joint estimation is the same as the error of estimation using only the least perturbed copy. Thus, the collusion does not result in a smaller error in our scheme.

Remark. Intuitively, since Y_2 is a perturbed observation of Y_1 as shown in (12), Y_2 cannot provide extra innovative information to improve the estimation accuracy achieved by utilizing only Y_1, and (11) is satisfied.

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 5.

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 (13). 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, (11) is satisfied.

5 Solution to General Cases

We now show that the solutions to the general cases of arbitrarily fine trust levels follow naturally from that to the two trust level case studied in Section 4.

5.1 Shaping the Noise

5.1.1 Independent Noise Revisited

In Section 4, we show that adding independent noise to generate two differently perturbed copies, although convenient, fails to achieve our privacy goal. 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 noise $Z_i, 1 \leq i \leq M$ 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 increases, the estimation errors decrease, so does the distortion $\mathcal{D}(X, \hat{X}(\mathbb{Y}))$.

Proof. Refer to Appendix D, available in the online supplemental material. □

Remark. The theorem says that when adding a new copy that is perturbed by independent noise, the estimation error decreases. It agrees with the intuition that a new independently-perturbed copy adds extra innovative information to improve the estimation accuracy.

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

5.1.2 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 (13) into a *corner-wave* property for a multidimensional noise covariance matrix. This property becomes the cornerstone of Theorem 4 which is a generalization of Theorems 1 and 2.

Corner-wave Property. Theorem 4 states that for M perturbed copies, the privacy goal in (10) is achieved if the noise covariance matrix $K_{\mathbb{Z}}$ has the corner-wave pattern as shown in (15). 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, and
- 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

$$\sigma_{Z_i}^2 < \sigma_{Z_{i+1}}^2, \forall i = 1, \ldots, M-1. \tag{14}$$

Then, the following equation holds:

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

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

$$K_{\mathbb{Z}} = \begin{bmatrix} \sigma_{Z_1}^2 K_X & \sigma_{Z_1}^2 K_X & \cdots & \sigma_{Z_1}^2 K_X \\ \sigma_{Z_1}^2 K_X & \sigma_{Z_2}^2 K_X & \cdots & \sigma_{Z_2}^2 K_X \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{Z_1}^2 K_X & \sigma_{Z_2}^2 K_X & \cdots & \sigma_{Z_M}^2 K_X \end{bmatrix}. \tag{15}$$

Proof. Refer to Appendix C, available in the online supplemental material. □

Remark. The corner-wave property of $K_{\mathbb{Z}}$ given in (15) guarantees that (10) holds. Therefore, the diversity attack does not help to improve the estimation accuracy.

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 Corollary 1.

Corollary 1. *If the privacy goal in (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 (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 (15). 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.

5.2 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 Z_1 to Z_M in parallel while Algorithm 2 sequentially.

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 (15)
4: Generate $\mathbb{Z}$ with $K_{\mathbb{Z}}$, according to (16)
5: Generate $\mathbb{Y} = HX + \mathbb{Z}$
6: Output $\mathbb{Y}$

Algorithm 2. Sequential Generation
1: // Input: X, K_X, and $\sigma^2_{Z_1}$ to $\sigma^2_{Z_M}$
2: // Output: Y_1 to Y_M
3: Construct $Z_1 \sim N(0, \sigma^2_{Z_1} K_X)$
4: Generate $Y_1 = X + Z_1$
5: Output Y_1
6: **for** i from 2 to M **do**
7: Construct noise $\xi \sim N(0, (\sigma^2_{Z_i} - \sigma^2_{Z_{i-1}})K_X)$
8: Generate $Y_i = Y_{i-1} + \xi$
9: Output Y_i
10: **end for**

5.2.1 Algorithm 1: Parallel Generation

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

$$f_{\mathbb{Z}}(v) = \frac{1}{\sqrt{(2\pi)^M \det(K_{\mathbb{Z}})}} e^{-\frac{1}{2} v^T K_{\mathbb{Z}}^{-1} v}, \tag{16}$$

where $K_{\mathbb{Z}}$ is given by (15).

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

Algorithm 1 serves as a baseline algorithm for the next two algorithms.

5.2.2 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. The validity of the alternative procedure is based on the insight in the following theorem.

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

$$\sigma^2_{Z_i} < \sigma^2_{Z_{i+1}}, \forall\, i = 1, \ldots, M-1.$$

Then, $\mathbb{Z}$ *is a jointly Gaussian vector and* $K_{\mathbb{Z}}$ *has the form in (15), if and only if* Z_1, *and* $(Z_i - Z_{i-1}), i = 2, ..., M$ *are mutually independent.*

Proof. Refer to Appendix F, available in the online supplemental material. □

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, Algorithm 2 generates the perturbed copies Y_1 to Y_M by adding the corresponding noise. We refer to Algorithm 2 as *sequential generation*.

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 (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, (10) is satisfied. This intuition is similar to the explanation for the case study in Section 4.

5.2.3 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.

5.3 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 section $\sigma^2_{Z_1}$ to $\sigma^2_{Z_M}$ 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 (15) (after properly ordering Z_1 to Z_M according to $\sigma^2_{Z_1}$ to $\sigma^2_{Z_M}$).

According to Section 2.1, 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 $\mathbb{Z}''$.

TABLE 2
Comparison of Applicabilities, Space Complexity, and Time Complexity of Three Proposed Algorithms

	Batch Generation	On-demand Generation	Space Complexity	Time Complexity
Algorithm 1	✓		$O(M+N^2)$	$O(N^3+MN^2)$
Algorithm 2	✓		$O(N^2)$	$O(N^3+MN^2)$
Algorithm 3	✓	✓	$O(M^2+N^2)$	$O(M^3+N^3)$

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.

Algorithm 3. On Demand Generation
1: // Input: X, K_X, $\sigma^2_{Z_1}$ to $\sigma^2_{Z_M}$, and values of $\mathbb{Z}'$: v_1
2: // Output: New copies $\mathbb{Z}''$
3: Construct $K_{\mathbb{Z}}$ with K_X and $\sigma^2_{Z_1}$ to $\sigma^2_{Z_M}$, according to (15)
4: Extract $K_{\mathbb{Z}'}$, $K_{\mathbb{Z}''\mathbb{Z}'}$, and $K_{\mathbb{Z}''}$ from $K_{\mathbb{Z}}$
5: Generate $\mathbb{Z}''$ as a Gaussian with mean and variance in (17) and (18), respectively
6: **for** i from $L+1$ to M **do**
7: Generate $Y_i = X + Z_i$
8: Output Y_i
9: **end for**

5.4 Time and Space Complexity

In this section, we study the time and space complexity of the three algorithms. One may notice that all the covariance matrices of noise in the three algorithms, such as (15) and (18), can be written as the Kronecker product of two matrices. For such covariance matrices, we have the following observation:

Lemma 1. *Assume that μ and K are the mean and covariance matrix of the jointly Gaussian random vector $\mathbb{G}$. If $K_{\mathbb{G}} = \Sigma_{\mathbb{G}} \otimes K_0$, where $\Sigma_{\mathbb{G}}$ and K_0 are $P \times P$ and $Q \times Q$, respectively, and K_0 is also a covariance matrix, then the time complexity of generating $\mathbb{G}$ is $O(P^3 + Q^3)$.*

Proof. Refer to Appendix G.1, available in the online supplemental material. □

Remark. Directly generating $\mathbb{G}$ using $K_{\mathbb{G}}$, the complexity is $O(P^3Q^3)$. Viewing $K_{\mathbb{G}}$ as a Kronecker product of two matrices of smaller dimensions, we can utilize the properties of Kronecker product to reduce the complexity to $O(P^3 + Q^3)$.

The proof suggests an efficient implementation of the proposed three algorithms. Note that for each algorithm, the time complexity may be further reduced.

Utilizing Lemma 1, we give the following theorems on the time and space complexity of the proposed three algorithms.

Theorem 6. *Given an N-dimensional data vector X, the time complexity of generating M perturbed copies using Algorithm 1 is $O(N^3 + MN^2)$, and the space complexity is $O(M + N^2)$.*

Proof. Refer to Appendix G.2, available in the online supplemental material. □

TABLE 3
Statistics of the Original Data *Age* and *Income*

	Mean μ_X	Variance σ^2_X
Age	50.06	303.03
Income	16.57	219.92

Theorem 7. *Given an N-dimensional data vector X, the time complexity of generating M perturbed copies using Algorithm 2 is $O(N^3 + MN^2)$, and the space complexity is $O(N^2)$.*

Proof. Refer to Appendix G.3, available in the online supplemental material. □

Remark. Using a similar set of arguments, we can show the time complexity of the independent noise scheme described in Section 5.1.1 is the same as Algorithm 2.

Theorem 8. *Given an N-dimensional data vector X and L $(1 \le L \le M-1)$ perturbed copies of X, the time complexity of generating $(M-L)$ perturbed copies using Algorithm 3 is $O(M^3 + N^3)$, and the space complexity is $O(M^2 + N^2)$.*

Proof. Refer to Appendix G.4, available in the online supplemental material. □

Table 2 compares the applicabilities and complexity of the three proposed algorithms. In summary, Algorithms 1 and 2 have less space and time complexity than Algorithm 3. Algorithm 3 offers data owners maximum flexibility by generating perturbed copies in an on-demand fashion.

6 Experiments

6.1 Methodology and Settings

We design two experiments, performance test (Experiment 1) and scalability test (Experiment 2). Experiment 1 explores answers to the following questions numerically:

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

Experiment 2 demonstrates the runtime of our proposed Algorithm 3.

We run our experiments on a real data set CENSUS [41], which is commonly used in the literature of privacy preservation such as [42], for carrying out the experiments and evaluating their performance in a fully controlled manner. This data set contains one million tuples with four attributes: *Age*, *Education*, *Occupation*, and *Income*. We take the first 10^5 tuples and conduct the experiments on the *Age* and *Income* attributes. The statistics and distribution of the data are shown in Table 3 and Fig. 1, respectively.

Given data X (*Age* and *Income*), 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 Z_i to X. The constant $\sigma^2_{Z_i}$ represents the perturbation magnitude determined by the data owner according to the trust level i. The

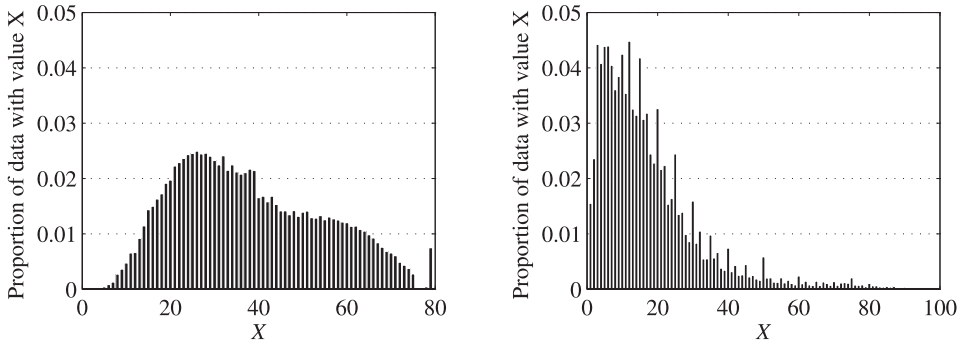


Fig. 1. Distribution of sensitive values *Age* and *Income*.

noise for different trust levels are generated either independently, or in a properly correlated manner following our proposed solution in Section 5.

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 has 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 has 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 of μ_X and K_X.

In Experiment 1, we use two performance metrics, average normalized estimation error and distribution of estimation error. For LLSE estimate of X based on $\mathbb{Y}$, i.e., $\hat{X}(\mathbb{Y})$, we define its normalized estimation error as

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

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. When showing the distribution of the estimation error, we use

$$\sqrt{\mathcal{D}(X,\hat{X}(\mathbb{Y}))}$$

directly, and one may see how large the distortion is, compared to the values of the original data shown in Fig. 1, as we do not normalize it. The distribution is represented by a histogram as well as a cumulative histogram. The curve of cumulative histogram starts from 0 and increases to 1. The faster the curve approaches 1, i.e., the bigger proportion of accurate estimates, the better the LLSE-based diversity attack performs. We conduct experiments on data with two attributes (i.e., $N=2$); however, for ease of illustration, we show the performance on different attributes separately.

6.2 Experiment 1: Performance Test

In this section, we show the superiority of our scheme over the scheme that simply adds independent noise, and how data miner's knowledge affects the power of LLSE-based diversity attacks. Algorithm 3 is used for the experiment due to its maximum flexibility among the three proposed algorithms.

M perturbed copies Y_i, $1 \le i \le M$, are generated one by one upon requests, adding independent noise to the original data or using our proposed Algorithm 3. Each request is at a different trust level with corresponding $\sigma^2_{Z_i}$ randomly generated in $[0.25, 1]$. Fig. 2 shows $\sigma^2_{Z_i}$ as a function of perturbed copy number i.

We assume that data miners can access all the M perturbed copies. This setting represents the most severe attack scenario where data miners jointly estimate X using all the available M perturbed copies. Since the perturbed copies are released one by one, the number of the available perturbed copies also increases one by one.

We also 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 and $N=2$ in our experiments.

Figs. 2a and 2b show the normalized estimation errors of both schemes as a function of the number of perturbed copies, on attributes *Age* and *Income*, respectively.

The results of the experiments clearly show that the diversity gain in joint estimation reduces the normalized estimation error dramatically. While for our algorithm, we find that the estimation error drops only when a perturbed copy with minimum perturbation magnitude so far becomes available. Using our algorithm, the curve of attacks utilizing the least perturbed copy overlaps with the curve of attacks utilizing all the available M copies. The above observations imply 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.

We also find that when data miners have perfect knowledge, the normalized estimation error decreases monotonically as M increases for copies perturbed by

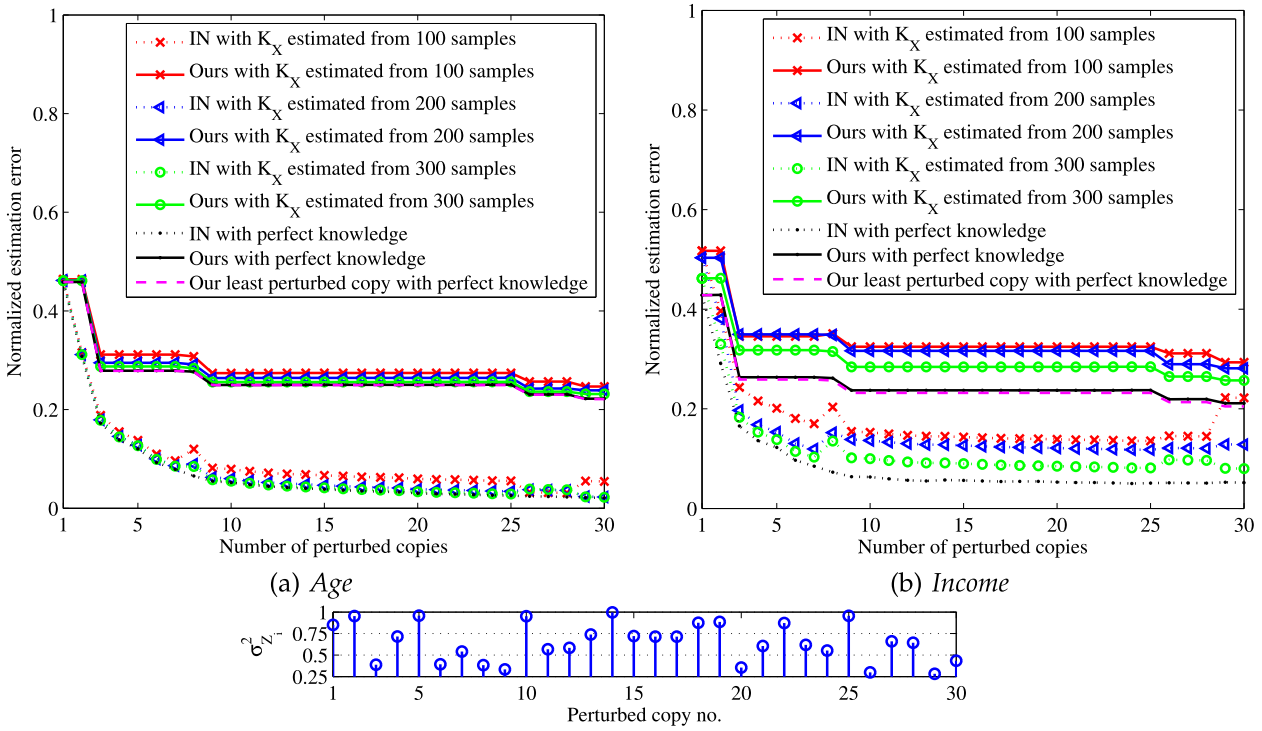


Fig. 2. Comparisons of average normalized estimation error of the independent noise scheme (denoted as *IN*) and our scheme (denoted as *Ours*) on the data (a)*Age* and (b)*Income*, respectively. The average normalized estimation error of each setting is shown as a function of the number of generated perturbed copies. Note that using our algorithm, the curve of attacks utilizing the least perturbed copy overlaps with the curve of attacks utilizing all the available M copies. Perturbation magnitude $\sigma^2_{Z_i}$ is shown as a function of perturbed copy number i at the bottom.

independent noise. This trend indicates a perfect reconstruction of X when M goes to infinity. It also confirms Theorem 3 empirically.

On the other hand, if the adversaries have to estimate μ_X and K_X from samples, i.e., the attackers have partial knowledge, the curve flattens and even slightly increases as M becomes large. 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. The estimation based on inaccurately estimated m_X and K_X is not optimal.

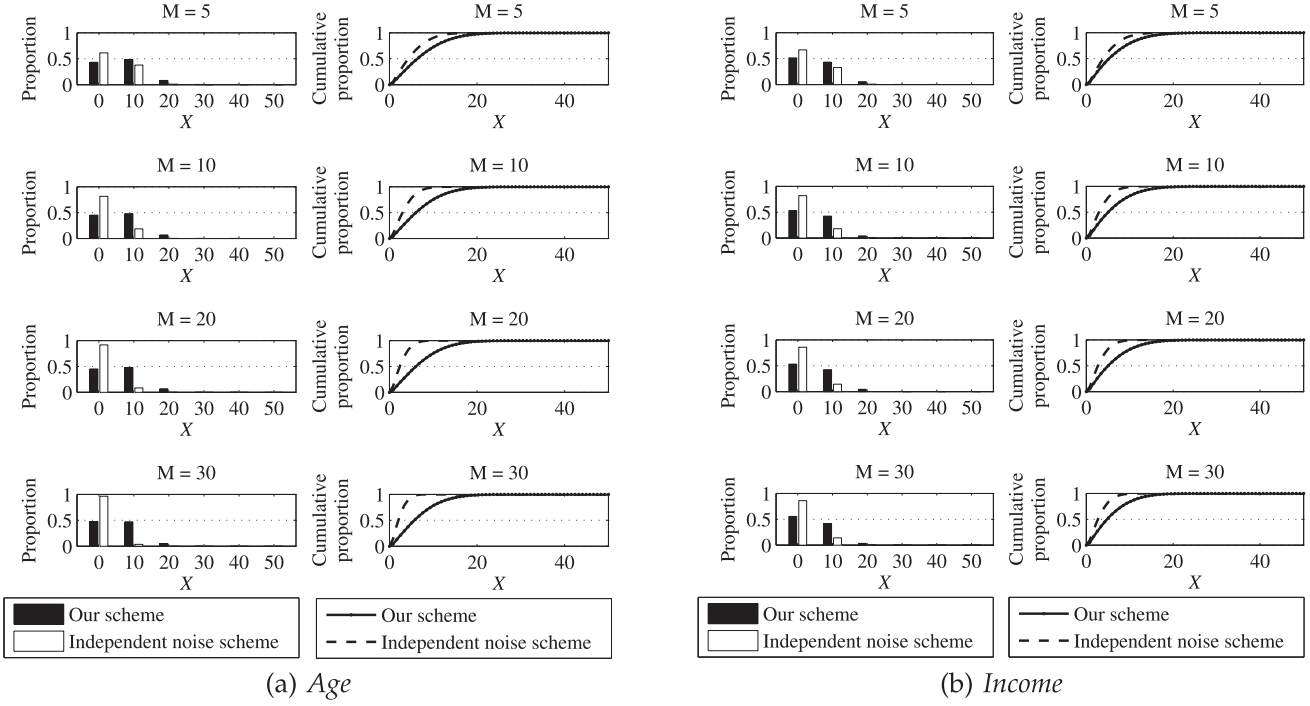
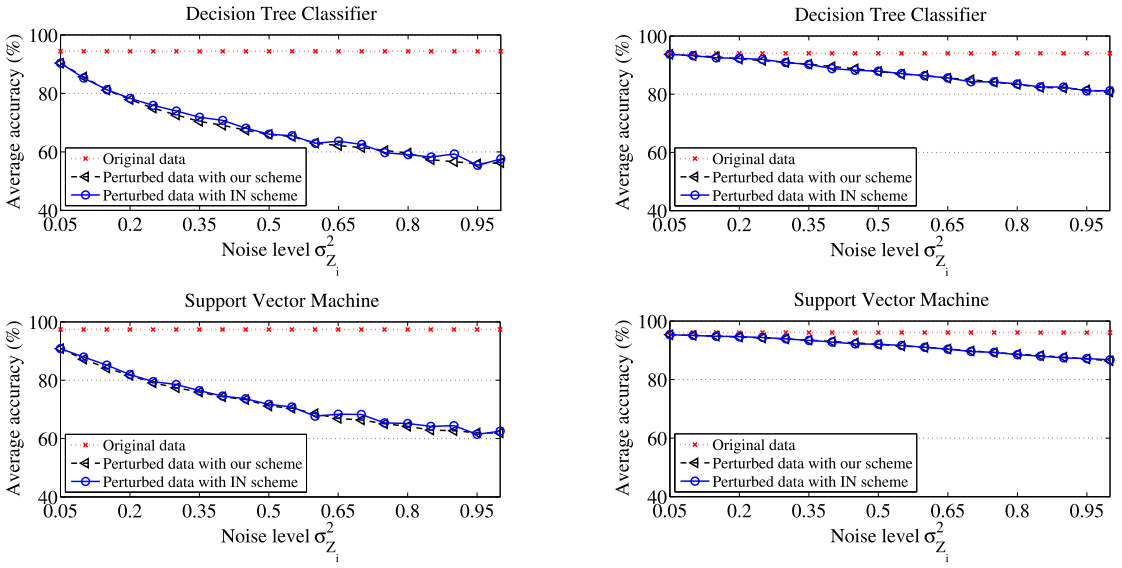


Fig. 3. The corresponding histogram and cumulative histogram of the estimation error when $M = 5$, 10, 20 and 30, respectively, using the two different schemes.



(a) Iris Plant database (b) Wisconsin Diagnostic Breast Cancer Database

Fig. 4. Comparison of utilities of perturbed copies by different noise addition techniques. We show the classification accuracy on the perturbed data at $M = 20$ different noise levels. The Iris Plant database has 150 tuples with four numerical attributes, and contains three classes of 50 tuples each. The Wisconsin Diagnostic Breast Cancer database has 699 tuples with nine numerical attributes, and contains two classes.

Consequently, the estimation accuracy does not always improve as M increases. Fig. 2 also shows that adversaries having more samples perform better in estimating μ_X and K_X, resulting in improved overall accuracy.

Figs. 3a and 3b show the corresponding histograms and cumulative histograms of the estimation errors for $M = 5$, 10, 20, and 30, using our proposed scheme and the independent noise scheme. The cumulative histograms of our scheme approaches 1 much slower than those of the independent noise scheme. This indicates that the adversaries obtain less accurate estimations from copies generated by our scheme than from those generated by the independent noise scheme. We also observe that as M increases, the cumulative histograms of our scheme are almost identical as expected; while those by the independent noise scheme approaches the vertical axis, implying estimation errors decrease as adversaries obtain more independently perturbed copies.

In summary, the privacy goal in Section 3.4 is achieved in this most severe attacking scenario.

We further verify that the perturbed copy by our scheme has the same utility as that by the independent noise scheme, if their trust levels are the same. We use the Iris Plant and Wisconsin Diagnostic Breast Cancer databases from the UCI Machine Learning Repository for the experiment. We measure the utilities with a decision tree classifier and a SVM classifier with radial basis kernel. The average accuracies over 10-fold cross validation are reported in Fig. 4. As seen from Fig. 4, at all noise levels, the accuracies by the same classifier on the data perturbed by adding independent noise and by properly adding correlated noise following our scheme are identical. Therefore, the perturbed copies at the same trust level by different noise addition techniques have the same utilities.

6.3 Experiment 2: Scalability Test

The scalability test is conducted in Matlab v7.6 on a PC with 2.5 GHz CPU and 2 GB memory. The attribute *Income* is used as the original data. We only test Algorithm 3 as it offers the maximum flexibility in generating perturbed copies and it has the highest time complexity among our three proposed algorithms. We use the independent noise scheme with the same settings as a baseline algorithm. Note that this scheme, although with less runtime, is not resistent to diversity attacks.

Theorem 8 states that to generate one tuple, the time complexity is $O(M^3 + N^3)$. To generate T tuples together, some of the computation can be shared, e.g., generating the covariance matrix of $\mathbb{Z}''$. As a result, the total time complexity to generate T perturbed tuples is $O(M^3 + N^3 + T(M^2N + MN^2))$, and the average time complexity for one tuple is $O(M^2N + MN^2)$ for large T.

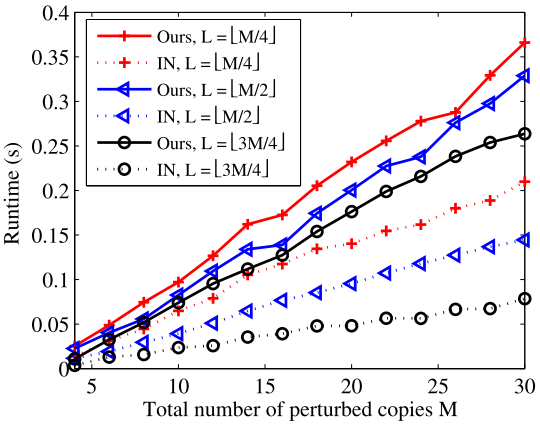


Fig. 5. The runtime as a function of the total number of perturbed copies M, when the data owner generates $M - L$ perturbed copies each of 10^5 tuples. The runtime is averaged on 100 repeated tests.

Fig. 5 shows the runtime of Algorithm 3 as a function of the total number of perturbed copies M. For each value of M, the data owner generates $M - L$ perturbed copies each of 10^5 tuples. We set $L = \lfloor M/4 \rfloor$, $\lfloor M/2 \rfloor$, and $\lfloor 3M/4 \rfloor$, respectively. Our observations are three-folded. First, our algorithm is fast. For example, generating 23 perturbed copies ($M = 30$, $L = \lfloor M/4 \rfloor = 7$) only takes 0.37 seconds. Second, the actual runtime of Algorithm 3 we observe only increases approximately linearly in M. This observed complexity is much smaller than the theoretical upper bound $O(M^3 + N^3 + M^2N + MN^2)$ we estimated in Section 5.4. Third, the runtime difference between Algorithm 3 and the independent noise scheme is considerably small. The time complexity of Algorithm 3 is the same as that of generating jointly Gaussian noise given the mean and covariance. One of the reasons why the independent noise scheme is marginally faster is that it uses an all-zero mean vector and diagonal covariance matrix.

7 Conclusion and Future Work

In this work, we expand the scope of additive perturbation based PPDM to multilevel trust (MLT), by relaxing an implicit assumption of 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 multilevel 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 multilevel trust. It is also of great interest to extend our approach to handle evolving data streams.

As with most existing work on perturbation-based PPDM, our work is limited in the sense that it considers only linear attacks. More powerful adversaries may apply nonlinear techniques to derive original data and recover more information. Studying the MLT-PPDM problem under this adversarial model is an interesting future direction.

Acknowledgments

The authors would like to thank Dr. Xiaokui Xiao for discussions related to the time complexity analysis.

References

[1] D. Agrawal and C.C. Aggarwal, "On the Design and Quantification of Privacy Preserving Data Mining Algorithms," *Proc. 20th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS '01)*, pp. 247-255, May 2001.

[2] R. Agrawal and R. Srikant, "Privacy Preserving Data Mining," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '00)*, 2000.

[3] K. Chen and L. Liu, "Privacy Preserving Data Classification with Rotation Perturbation," *Proc. IEEE Fifth Int'l Conf. Data Mining*, 2005.

[4] Z. Huang, W. Du, and B. Chen, "Deriving Private Information From Randomized Data," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD)*, 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," *Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE)*, 2007.

[6] K. Liu, H. Kargupta, and J. Ryan, "Random Projection-Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 1, pp. 92-106, Jan. 2006.

[7] S. Papadimitriou, F. Li, G. Kollios, and P.S. Yu, "Time Series Compressibility and Privacy," *Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB '07)*, 2007.

[8] Y. Lindell and B. Pinkas, "Privacy Preserving Data Mining," *Proc. Int'l Cryptology Conf. (CRYPTO)*, 2000.

[9] J. Vaidya and C.W. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, 2002.

[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," *Proc. ACM SIGKDD Int'l Conf. 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," *Proc. IEEE Fifth Int'l Conf. Data Mining*, 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," *Proc. Second Int'l Workshop Data Management on New Hardware (DaMoN '06)*, 2006.

[14] C.C. Aggarwal and P.S. Yu, "A Condensation Approach to Privacy Preserving Data Mining," *Proc. Int'l Conf. Extending Database Technology (EDBT)*, 2004.

[15] E. Bertino, B.C. Ooi, Y. Yang, and R.H. Deng, "Privacy and Ownership Preserving of Outsourced Medical Data," *Proc. 21st Int'l Conf. Data Eng. (ICDE)*, 2005.

[16] D. Kifer and J.E. Gehrke, "Injecting Utility Into Anonymized Datasets," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 2006.

[17] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-Diversity: Privacy Beyond K-Anonymity," *Proc. Int'l Conf. Data Eng.*, 2006.

[18] L. Sweeney, "K-Anonymity: A Model for Protecting Privacy," *Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS)*, vol. 10, pp. 557-570, 2002.

[19] X. Xiao and Y. Tao, "Personalized Privacy Preservation," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 2006.

[20] R. Agrawal, R. Srikant, and D. Thomas, "Privacy Preserving OLAP," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 2005.

[21] W. Du and Z. Zhan, "Using Randomized Response Techniques for Privacy-Preserving Data Mining," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, 2003.

[22] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy Preserving Mining of Association Rules," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, 2002.

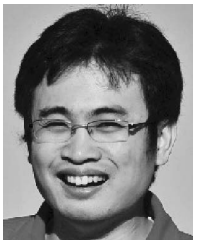
[23] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the Privacy Preserving Properties of Random Data Perturbation Techniques," *Proc. IEEE Third Int'l Conf. Data Mining*, 2003.

[24] R. Agrawal, A. Evfimievski, and R. Srikant, "Information Sharing across Private Databases," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 2003.

[25] R. Agrawal, D. Asonov, M. Kantarcioglu, and Y. Li, "Sovereign Joins," *Proc. 22nd Int'l Conf. Data Eng. (ICDE '06)*, 2006.

[26] C. Clifton, M. Kantarcioglu, X. Lin, J. Vaidya, and M. Zhu, "Tools for Privacy Preserving Distributed Data Mining," *ACM SIGKDD Explorations,* vol. 4, no. 2, pp. 28-34, 2003.
[27] B.A. Huberman, M. Franklin, and T. Hogg, "Enhancing Privacy and Trust in Electronic Communities," *Proc. First ACM Conf. Electronic Commerce,* pp. 78-86, Nov. 1999.
[28] M. Freedman, K. Nissim, and B. Pinkas, "Efficient Private Matching and Set Intersection," *Advances in Cryptology—EUROCRYPT,* vol. 3027, pp. 1-19, 2004.
[29] L. Kissner and D. Song, "Privacy-Preserving Set Operations," *Proc. Int'l Cryptology Conf. (CRYPTO),* 2005.
[30] A. Iliev and S. Smith, "More Efficient Secure Function Evaluation Using Tiny Trusted Third Parties," Technical Report TR2005-551, Dept. of Computer Science, Dartmouth Univ., 2005.
[31] J. Byun, Y. Sohn, E. Bertino, and N. Li, "Secure Anonymization for Incremental Datasets," *Proc. Third VLDB Workshop Secure Data Management,* 2006.
[32] X. Xiao and Y. Tao, "M-Invariance: Towards Privacy Preserving Re-Publication of Dynamic Datasets," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* 2007.
[33] B. Fung, K. Wang, A. Fu, and J. Pei, "Anonymity for Continuous Data Publishing," *Proc. Int'l Conf. Extending Database Technology (EDBT),* 2008.
[34] G. Wang, Z. Zhu, W. Du, and Z. Teng, "Inference Analysis in Privacy-Preserving Data Re-Publishing," *Proc. Int'l Conf. Data Mining,* 2008.
[35] Y. Li and M. Chen, "Enabling Multi-Level Trust in Privacy Preserving Data Mining," Technical Report UCB/EECS-2008-156, EECS Dept., Univ. of California, Berkeley, http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-156.html, Dec. 2008.
[36] X. Xiao, Y. Tao, and M. Chen, "Optimal Random Perturbation at Multiple Privacy Levels," *Proc. Int'l Conf. Very Large Data Bases,* 2009.
[37] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting Privacy Breaches in Privacy Preserving Data Mining," *Proc. ACM Symp. Principles of Database Systems,* 2003.
[38] C. Aggarwal, "Privacy and the Dimensionality Curse," *Privacy-Preserving Data Mining,* pp. 433-460, 2008.
[39] K. Shanmugan and A. Breipohl, *Random Signals: Detection, Estimation, and Data Analysis.* John Wiley & Sons Inc, 1988.
[40] J. Brewer, "Kronecker Products and Matrix Calculus in System Theory," *IEEE Trans. Circuits and Systems,* vol. 25, no. 9, pp. 772-781, Sept. 1978.
[41] "MPC Data Projects,"http://www.ipums.org, 2012.
[42] X. Xiao and Y. Tao, "Output Perturbation with Query Relaxation," *Proc. Int'l Conf. Very Large Data Bases,* 2008.
[43] G. Golub and C. Van Loan, *Matrix Computations.* The Johns Hopkins Univ. Press, 1996.
[44] "Multivariate Normal Distribution,"http://en.wikipedia.org/wiki/Multivariate_normal_distribution, 2012.
[45] D. Knuth, *The Art of Computer Programming: Seminumerical Algorithms,* vol. 2, ch. 3, Addison-Wesley, 1981.

Yaping Li received the BS degree from the Department of Computer Science, State University of New York, Stony Brook and the PhD degree from the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. She is currently a postdoctoral researcher in the Department of Information Engineering at the Chinese University of Hong Kong. Her research interests include database privacy, secure network coding, and applications for secure coprocessors.



Minghua Chen received the BEng and MS degrees from the Department of Electronics Engineering, Tsinghua University in 1999 and 2001, respectively. He received the PhD degree from the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley (UC Berkley) in 2006. He spent one year visiting Microsoft Research Redmond, Washington, as a postdoctoral researcher. He joined the Department of Information Engineering, the Chinese University of Hong Kong, in 2007, where he is currently an assistant professor. He is coauthor of two books on wireless and IPv6 networking written in 2000 and 2008, respectively. His research interests include complex systems and networked systems, distributed and stochastic network optimization and control, multimedia networking, p2p networking, wireless networking, multilevel trust data privacy, network coding, and secure network communications. He received the Eli Jury Award from UC Berkeley in 2007 (presented to a graduate student or recent alumnus for outstanding achievement in the area of Systems, Communications, Control, or Signal Processing), the ICME Best Paper Award in 2009, and the IEEE Transactions on Multimedia Prize Paper Award in 2009.

Qiwei Li received the BEng degree in electronic engineering from Tsinghua University in 2008, and the MPhil degree in information engineering from The Chinese University of Hong Kong in 2010. Currently, he is a PhD student in the Department of Statistics, Rice University. His research interests focus on bioinformatics, including repeats detection, motif discovery, microarray data analysis, microarray data analysis, molecular dynamics, etc.

Wei Zhang received the BEng degree in electronic engineering from Tsinghua University, Beijing, in 2007, and the MPhil degree in information engineering from the Chinese University of Hong Kong in 2009. He is currently working toward the PhD degree in the Department of Information Engineering at the Chinese University of Hong Kong. His research interests include machine learning and its applications to computer vision, image processing, and data mining.