DPPro: Differentially Private High-Dimensional Data Release via Random Projection

Chugui Xu, *Student Member, IEEE*, Ju Ren, *Member, IEEE*, Yaoxue Zhang, Zhan Qin, *Student Member, IEEE*, and Kui Ren, *Fellow, IEEE*

***Abstract*— Releasing representative data sets without compromising the data privacy has attracted increasing attention from the database community in recent years. Differential privacy is an influential privacy framework for data mining and data release without revealing sensitive information. However, existing solutions using differential privacy cannot effectively handle the release of high-dimensional data due to the increasing perturbation errors and computation complexity. To address the deficiency of existing solutions, we propose DPPro, a differentially private algorithm for high-dimensional data release via random projection to maximize utility while guaranteeing privacy. We theoretically prove that DPPro can generate synthetic data set with the similar squared Euclidean distance between high-dimensional vectors while achieving (ϵ, δ)-differential privacy. Based on the theoretical analysis, we observed that the utility guarantees of released data depend on the projection dimension and the variance of the noise. Extensive experimental results demonstrate that DPPro substantially outperforms several state-of-the-art solutions in terms of perturbation error and privacy budget on high-dimensional data sets.**

***Index Terms*— Differential privacy, high-dimensional data, privacy guarantees, random projection, utility guarantees.**

I. Introduction

PRIVACY preserving data publishing (PPDP) [1] has received considerable attention in recent years as a promising approach for sharing data while preserving data privacy. A classic case of PPDP is when a database can be modeled as a table, where each row may contain information about an individual (e.g. health data, financial or employment information). The aim of PPDP is to release some manipulated sensitive data such that the released data can still be used for the intended purposes, but the privacy of the individuals in the database is preserved.

Differential privacy [2] is an influential framework to quantify to what extent individual privacy in a statistical database is preserved while releasing useful aggregate information about the database. It provides strong privacy guarantees by requiring the indistinguishably of the involvement of an individual in the dataset based on the released information [3], [4]. However, putting differential privacy into practice remains a challenging problem. Since its proposal, there have been many efforts to develop data release mechanisms for different kinds of input databases, and for different objectives of intended uses. Nevertheless, existing techniques using differential privacy cannot handle the release of high-dimensional data effectively and efficiently. In particular, when the input dataset contains high dimensions and large attribute domains, existing solutions require injecting a prohibitive amount of noise, which renders the published data to be nearly useless [5]–[8]. In fact, all these solutions suffer from the curse of dimensionality, that is, they cannot achieve either reasonable scalability or desirable utility. For example, a database has 10M tuples, 20 attributes (dimensions), and 10 values per attribute. The full tuple distribution has $10^{20} \approx 10T$ cells, and most of them have non-zero counts after noise injection. Thus, the average information in each cell can be calculated as $\frac{10M}{10T} = 10^{-6}$. If the average noise is $1/\epsilon = 10$ (for $\epsilon = 0.1$). Obviously, the signal-to-noise ratio is extremely low. Therefore, special solution is in high demand to overcome the challenges incurred by high dimensionality.

Generally, to address the challenges, a promising way is to decompose high-dimensional data into a set of low-dimensional marginal tables, along with an inference mechanism that can infer the joint data distribution from these tables. PrivBayes [8] is a representative solution that can learn a set of low-dimensional conditional probabilities via a Bayesian network and approximate the joint distribution by the chain rule for Bayesian networks. However, with the increasing number of attributes, PrivBayes has to face a limitation that the privacy budget used for determining each attribute's parent set decreases quickly, making the learnt conditional probabilities unreliable. Recently, an improved method based on PrivBayes is proposed. Chen *et al.* [5] develop a robust sampling-based

Manuscript received February 3, 2017; revised July 2, 2017 and July 21, 2017; accepted July 21, 2017. Date of publication August 9, 2017; date of current version August 29, 2017. This work was supported in part by the Natural Science Foundation Project of Fujian Province of China under Grant 2015J01271, in part by the Young, Middle-aged Scientific Research Project in the Department of Education of Fujian Province of China under Grant JAT160469, in part by the Innovation-Driven Project of Central South University under Grant 2016CXS013, in part by the International Science & Technology Cooperation Program of China under Grant 2013DFB10070, in part by the China Hunan Provincial Science & Technology Program under Grant 2012GK4106, and in part by the Innovation Foundation for Postgraduate of Central South University under Grant 2016zzts057. The work of K. Ren was supported by the U.S. National Science Foundation under Grant CNS-1262277. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Stefan Katzenbeisser. *(Corresponding author: Ju Ren.)*

C. Xu is with the School of Information Science and Engineering, Central South University, Changsha 410083, China, and also with Sanming University, Sanming 365004, China (e-mail: chuguixu@csu.edu.cn).

J. Ren and Y. Zhang are with the School of Information Science and Engineering, Central South University, Changsha 410083, China (e-mail: renju@csu.edu.cn; zyx@csu.edu.cn).

Z. Qin and K. Ren are with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14228 USA (e-mail: zhanqin@buffalo.edu; kuiren@buffalo.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2017.2737966

framework to systematically explore the dependencies among all attributes and subsequently build a dependency graph. They apply the junction tree algorithm to establish an inference mechanism for inferring the joint data distribution, which may cause some errors in learning the pairwise correlations of all attributes. It makes this solution still not able to adequately capture the characteristics of the underlying data to maximize data utility. Admittedly, existing solutions can help tackling many research barriers by incorporating the inference mechanism of joint data distribution to the differential privacy mechanism. However, there are still some main challenges in releasing high-dimensional data with differential privacy.

(i) The underlying distribution of the data may be unknown in many cases or different from the assumed distribution, especially for data with high dimensions, rendering the released data to be nearly useless.
(ii) The high dimensions and large attribute domains result in the synthetic data that may have skewed distributions, leading to significant perturbation or estimation errors in releasing high-dimensional data.
(iii) How to tune the tradeoff between the reduced dimensionality and added noise when we design a mechanism for releasing high-dimensional data with differential privacy?

In this paper, we investigate the problem of differentially private high-dimensional data release, and present a data release algorithm via random projection to tackle this problem, namely DPPro. The main idea of DPPro is to project a dataset from a high-dimensional space to a randomly chosen lower-dimensional subspace in order to preserve pairwise L_2 distances and thus users segmentation based on these distances. Such that, it can minimize the amount of added noise to maximize utility while guaranteeing the privacy of released data. Summarily, our contributions of this paper are as follows.

(i) We proposed an algorithm to project a d-dimensional vector representation of a user's feature attributes into a lower k-dimensional space by first applying a random projection, and then adding Gaussian noise to each resulting vector in order to maximize utility while guaranteeing privacy.
(ii) We theoretically prove that DPPro satisfies (ϵ, δ)-differential privacy with regard to a change in an individual attribute. It indicates that an adversary who knows all but one attribute of a user cannot recover the value of that attribute from the released data with high confidence, regardless of the adversary's background knowledge.
(iii) We theoretically analyze the utility of the released data by DPPro, and prove that the squared Euclidean distance between each pair of users can be preserved in expectation. Furthermore, we found that the utility guarantees of the released data by DPPro depend on projection dimension and the variance of the noise with high probability.
(iv) We comprehensively evaluate the performance of DPPro by extensive experiments on a large number of real datasets, the experimental results demonstrate that DPPro can generate highly accurate synthetic data and significantly outperform several state-of-the-art solutions.

The remainder of this paper is organized as follows. We present the related work in Section II, and provide the preliminaries in section III. Section IV introduces the details of DPPro and section V conducts the theoretical analysis on the privacy and utility guarantees of the DPPro. Section VI shows experimental results, followed by a conclusion in section VII.

II. Related Work

Various approaches have been proposed recently for releasing differentially private data. We briefly review here the most relevant work to our paper and discuss how our work differs from existing work.

A few research efforts have been devoted to differentially private multi-dimensional data release by applying the Laplace mechanism. Barak *et al.* [9] show how to construct a synthetic database to preserve all low-dimensional marginals by adding noise to the Fourier domain. The problem in [9] is equivalent to publishing OLAP (online analytical processing) cubes, which is studied by Ding *et al.* [10]. They firstly compute a subset of cuboids and then generate the remaining cuboids from this subset. The main limitation of these two approaches is their exponential complexity in the dimensionality of the domain. Mohammed *et al.* [11] introduce probabilistic generalization to overcome the curse of dimensionality. However, with the increasing dimensionality, the benefit of generalization diminishes rapidly. DPCube [12] is based on KD-Tree partitioning. It first uses Dworks method to generate a DP cell histogram and then applies partitioning on the noisy cell histogram to create the final DP histogram. However, for high-dimensional data with large attribute domain, either the level of partitioning will be high resulting in high perturbation errors or the distribution of each partition will be skewed leading to high estimation error. Acs *et al.* [13] study two sanitization algorithms for generating differential privacy histograms. The technique improves the Fourier perturbation scheme through tighter utility analysis, while there are limitations for high dimension data. However, when the number of bins in original histograms is extremely large, the accuracy of each partitioning step would have large perturbation error and the computation complexity would be proportional to the quadratic number of bins in the worst case. Cormode *et al.* [14] consider the scalability aspect of the problem. They design a statistical process to compute a private summary without materializing the entire contingency table.

Moreover, some researchers investigate other transformations of the original data aimed at reducing their sensitivity [15], [16]. Soria-Comas *et al.* [15] present an approach that combines k-anonymity and ϵ-differential privacy in order to reduces the information loss of standard differential privacy, while preserving its privacy guarantee. SÃ¡nchez *et al.* [16] adopt individual ranking microaggregation in order to reduce the amount of noise needed to satisfy differential privacy.

Some works have been presented to release private data by incorporating the Johnson-Lindenstrauss transform to differential privacy mechanism [17]–[23]. Blocki *et al.* [17] apply the

Johnson-Lindenstrauss transform to the task of approximating cut-queries, and the technique outperforms existing algorithms for answering cut-queries in a differentially private manner. Upadhyay [23] improves the run time of the technique proposed by Blocki *et al.* without using the graph sparsification trick, and proves that a general class of random projection matrices that satisfies the Johnson-Lindenstrauss lemma can preserve differential privacy. Kenthapadi *et al.* [18] introduce and motivate the problem of releasing data to enable third parties to perform distance computations and clustering on users. Sheffet [20] gives three differentially private techniques for approximating the 2nd-moment matrix of the data that output a positive-definite matrix, and analyzes their utility for corresponding to existing techniques in linear regression theoretically and empirically. Linear regression is one of the most prevalent techniques in data analysis. To achieve similar guarantees on data under differentially private estimators, Sheffet [19] analyzes the result of using the JL transform for projecting the least squares problem and estimating confidence intervals over the projected data, and provides an analysis of a differentially private technique for Ordinary Least Squares (OLS)' statistical inference. Generally, low-rank factorization is used in many areas of computer science where one performs spectral analysis on large sensitive data stored in the form of matrices. The problem of computing a low-rank factorization of a $m \times n$ matrix in the general turnstile update model including both the private and non-private setting is studies in [22]. Compared to prior works, their result can achieve time and space efficiency, optimal additive error and applicability. Upadhyay [21] studies two computationally efficient and sub-linear space algorithms for computing a differentially private low-rank factorization. He also shows that both these privacy levels are stronger than those studied in some existing algorithms. However, their works cannot give the optimal settings to tune the tradeoff between perturbation errors and privacy.

Several recent works have been proposed to address the problem of differentially private high-dimensional data releasing. Qardaji *et al.* [7] study how to generate accurate k-way marginals for a binary dataset. They propose PriView that uses covering design to select a set of low-dimensional marginals called views and then generates *k*-way marginals based on maximum entropy optimization. The work closest to ours is PrivBayes [8], which iteratively learns the parent sets of the attributes in a Bayesian network by applying the exponential mechanism with a surrogate function for mutual information. But the performance of PrivBayes is sensitive to the randomly selected initial attribute, and limits the size of each attribute's parent set to be identical. Based on PrivBayes, Su *et al.* [24] present DP-SUBN, which develops a non-overlapping covering design (NOCD) method for generating all 2-way marginals of a given set of attributes to improve the fitness of the Bayesian network and reduce the communication cost. Compared to PrivBayes, Chen *et al.* [5] feature a systematic exploration of attribute correlations and approximate the joint distribution based on the solid inference foundation of the junction tree algorithm while minimizing the resultant error, which together achieve substantially better performance.

TABLE I
FREQUENTLY USED SYMBOLS

Symbol	Description
ϵ, δ	Differential privacy parameters
$D, \hat{D}$	Input database and output database
D_1, D_2	Any two neighboring datasets
n	The number of vectors in a dataset
x, y	Any two vectors in datasets
R	Random projection matrix
$\theta_2(R)$	The L_2-sensitivity of a $d \times k$ projection matrix R
ϕ	Random noise matrix
σ^2	The variance of a Gaussian distribution
k	Random projection dimensionality
d	The original dimensionality
$\|\cdot\|_2$	The norm of L_2
k_{opt}	The optimal value of projection dimension
erf( )	Error function
χ^2_k	The chi-squared distribution with k degrees of freedom
$u(x, y)$	The Euclidean distance of any pair vectors x, y
$U^2(x, y)$	The squared distance between two users in original space
$\tilde{U}^2(x, y)$	The squared distance between two users in projection space

But there is maximum final error in using the sampling-based inference mechanism. The connection between probabilistic inference and differential privacy is also studied in [25]. Li *et al.* [6] present DPCopula, a differentially private data synthesization method for high dimensional and large domain data using copula functions. Copula functions are used to describe the dependence between multivariate random vectors and allow us to build the multivariate joint distribution using one-dimensional marginal distributions. Day and Li [26] present DPSense to publish statistical information (i.e., column counts) of input datasets under differential privacy via sensitivity control.

However, different from these solutions, our work focuses on handling the release of high-dimensional data due to the increasing perturbation errors and computation complexity. To relieve the curse of dimensionality, DPPro is designed to reserve pairwise L_2-distances between users by random projection, and provide privacy and utility guarantees of released data. Whenever answers to users queries can be formalized as L_2-distances of the product between the given database and a query-vector, utility bounds are straight-forward. Specially, DPPro allows us to publish a sanitized covariance matrix that preserves differential privacy with regard to bounded changes (each row in the matrix can change by at most a L_2 vector) while adding noise of magnitude dependent on the optimal projection dimension, and independent of the size of the matrix.

III. PRELIMINARIES

This section reviews two concepts closely related to our work, namely, differential privacy and random projection. The mathematical notations frequently used in this paper are summarized in Table I.

A. Differential Privacy

Let D be a sensitive dataset to be published. Differential privacy requires that, prior to D's release, it should be modified

using a randomized algorithm A, such that the output of A does not reveal much information about any particular tuple in D. The formal definition of differential privacy is detailed as follow.

Definition 1 ((ϵ, δ)-Differential Privacy [27]): A (randomized) algorithm A satisfies (ϵ, δ)-differential privacy. For all inputs D_1 and D_2 differing in at most one user's one attribute value, and for all sets of possible outputs. $O \subseteq Range(A)$,

$$Pr\left[A\left(D_1\right) \in O\right] \leq \exp(\epsilon) \cdot Pr\left[A\left(D_2\right) \in O\right] + \delta, \quad (1)$$

where $Pr\left[\cdot\right]$ denotes the probability of an event.

Intuitively, it can be derived that when $\delta = 0$, (ϵ, δ)-differential privacy is equivalent to ϵ-differential privacy [2]. Since δ is non-negative, any mechanism that satisfies ϵ-differential privacy also satisfies (ϵ, δ)-differential privacy for any value of δ. When $\delta > 0$, (ϵ, δ)-differential privacy relaxes ϵ-differential privacy by ignoring outputs of A with very small probability (controlled by parameter δ). In other words, an (ϵ, δ)-differentially private mechanism satisfies ϵ-differential privacy with a probability controlled by δ.

Generally, the maximal impact of a tuple to the output of a function $\theta(Q)$ is called its sensitivity. A basic mechanism for enforcing (ϵ, δ)-differential privacy is the Gaussian mechanism, which involves the concept of L_2-sensitivity [27].

Definition 2 (L_2-SENSITIVITY [27]): For any two neighboring databases D_1 and D_2, the L_2-sensitivity $\theta(Q)$ of a query set Q is defined as:

$$\theta(Q) = \max_{D_1, D_2} \parallel Q(D_1), Q(D_2) \parallel_2 . \quad (2)$$

L_2-sensitivity $\theta(Q)$ depends on the data domain and the query set Q, rather than the actual data.

Many mechanisms are adopted to achieve (ϵ, δ)-differential privacy. In this paper, since we aim to preserve pairwise distances with the goal of performing users segmentation and nearest neighbor computations, an algorithm is that one is able to guarantee (ϵ, δ)-differential privacy by adding noise from the Gaussian distribution, with the variance of the noise depending on the L_2 sensitivity of the chosen projection matrix R, which is defined as follows.

Definition 3 (L_2-Sensitivity of R): Define the L_2-sensitivity of a $d \times k$ projection matrix $R = \left\{R_{ij}\right\}_{d \times k}$, denoted by $\theta_2(R)$, as the maximum L_2-norm of any row in R, i.e., $\theta_2(R) = \max_{1 \leq i \leq d} \sqrt{\sum_{j=1}^{k} \mid R_{ij} \mid^2}$. Equivalently, $\theta_2(R)$ can be defined as $\max_{e_i} \|e_i R\|_2$, where $\{e_i\}_{i=1}^{d}$ are standard basis unit vectors.

Privacy composition will be useful to understand how privacy parameters for each steps of an algorithm compose into privacy guarantees for the entire algorithm. The following useful theorem is a special case of a theorem proven by Dwork *et al.* [28].

Theorem 1 (Privacy Composition [28]): Let $\epsilon > 0, \delta < 1$, and let $A_i, 0 \leq i \leq T$ be a non-interactive privacy mechanism which satisfies ϵ_i-differential privacy.

$$\epsilon_i \leq \frac{\epsilon}{\sqrt{8T \log\left(\frac{1}{\delta}\right)}}.$$

Then the output of mechanism $A(D) = (A_1(D), ..., A_i(D))$ over the database D is (ϵ, δ)-differential privacy.

Definition 4 (Query Matrix [29]): A query matrix is a collection of linear queries, arranged by rows to form an $p \times m$ matrix.

Given a $p \times m$ query matrix Q, the query answer for Q is a length-p column vector of query results, which can be computed as the matrix product Q_x. For example, an $m \times m$ identity query matrix I_m will result in a length-m column vector consisting of all the cell counts in the original data vector x.

A data release algorithm, consisting of a sequence of designed queries using the differential privacy interface, can be represented as a query matrix. We will use this query matrix representation in the analysis of our algorithms.

We analyze the utility of the released data by the notion of (ϵ, δ)-usefulness [30].

Definition 5 ((ϵ, δ)-Usefulness [30]): A database mechanism A is (ϵ, δ)-useful for queries in class C if with probability $1 - \delta$, for every $Q \in C$, and every database D, $A(D) = \hat{D}$, $\mid Q(\hat{D}) - Q(D) \mid \leq \epsilon$.

B. Random Projection

Random projection refers to the technique of projecting a set of data points from a high-dimensional space to a randomly chosen lower-dimensional subspace in order to deal with the curse of dimensionality. The key idea of random projection arises from the Johnson-Lindenstrauss Lemma [31].

Lemma 1 (JOHNSON-LINDENSTRAUSS LEMMA [31]): For any $0 < \lambda < 1$ and any integer s, let k be a positive integer such that $k \geq \frac{4 \ln s}{\lambda^2/2 - \lambda^3/3}$. Then, for any set S of $s = \mid S \mid$ data points in $\mathbb{R}^m$, there is a map $f: \mathbb{R}^d \to \mathbb{R}^k$ such that, for all $x, y \in S$,

$$(1 - \lambda) \parallel x - y \parallel^2 \leq \parallel f(x) - f(y) \parallel^2 \leq (1 + \lambda) \parallel x - y \parallel^2, \quad (3)$$

where $\| \cdot \|$ denotes the vector 2-norm. The proof of this result may be found in [32].

Lemma 1 shows that any set of s points in m-dimensional Euclidean space can be embedded into an $O(\frac{\log s}{\lambda^2})$-dimensional space such that the pairwise L_2 distances of any two points are maintained within an arbitrarily small factor. This beautiful property implies that it is possible to change the data's original form by reducing its dimensionality but still maintains its statistical characteristics.

Lemma 2 [33]: Let $R = (r_{ij})$ be a random $n \times r$ matrix, such that each entry r_{ij} is chosen independently according to $N(0, 1)$. For any vector fixed $X \in \mathbb{R}^n$, and any $\lambda > 0$, let $X' = \frac{1}{\sqrt{k}}(R^T X)$. Then,

$$E\left(\left\|X'\right\|^2\right) = \|X\|^2, \quad (4)$$

$$Pr\left[\left\|X'\right\|^2 > (1 + \lambda)\|X\|^2\right] \leq \left((1 + \lambda) e^{-\lambda}\right)^k \leq e^{-\left(\lambda^2 - \lambda^3\right)\frac{k}{4}}, \quad (5)$$

$$Pr\left[\left\|X'\right\|^2 < (1 - \lambda)\|X\|^2\right] \leq \left((1 - \lambda) e^{-\lambda}\right)^k \leq e^{-\left(\lambda^2 - \lambda^3\right)\frac{k}{4}}. \quad (6)$$

Intuitively, Lemma 2 indicates vectors in a high-dimensional space after random projection with random directions are almost orthogonal. The proof of Lemma 2 can be found in [33].

IV. The Proposed DPPro Algorithm

This section presents an detailed introduction of DPPro for releasing a high-dimensional dataset in an (ϵ, δ)-differentially private manner. The framework of DPPro is shown in Fig. 1.

A. The Overview of DPPro

The main idea of DPPro is to project a $n \times d$ dataset of user data into a lower-dimensional $n \times k$ dataset that can be publicly shared without compromising the privacy of any individual involved and can simultaneously preserve distance characteristics of the original dataset. Using a random $k \times d$ matrix R, where $X_{n\times d}$ is the original set of n d-dimensional observations, $X^{RP}_{k\times n}$ is the projection of the data into a lower k-dimensional subspace, which can be denoted as

$$X^{RP}_{k\times n} = R_{k\times d} X_{d\times n}. \tag{7}$$

The key idea of random projection arises from Johnson-Lindenstrauss Lemma, i.e., Lemma 1. It claims that the distances between points can be approximately preserved, if the points in a vector space are projected onto a randomly selected subspace of suitably high dimension. We denote the Euclidean distance between two data vectors x and y in the original large-dimensional space as $\| x - y \|$. After the random projection, the distance is approximated by the scaled Euclidean distance $u(x, y)$ of these vectors in the reduced space:

$$u(x, y) = \sqrt{d/k} \parallel Rx - Ry \parallel, \tag{8}$$

where d and k are the original and the reduced dimensionality of the dataset respectively. The scaling term $\sqrt{d/k}$ takes into account the decrease in the dimensionality of the data. According to Lemma 1, the expected norm of a unit vector after random projection is $\sqrt{d/k}$ [31].

The detailed procedures of DPPro are presented in Algorithm 1. First, the data to be released is projected into a much lower dimension ($k \ll d$) subspace to obtain a reduced representation that preserves pairwise distances (steps 1-3), similar to many dimensionality reduction techniques. Then, the resulting data is slightly perturbed by adding noise ϕ to guarantee the privacy of each user (steps 4-7). The benefit of random projection is to achieve (ϵ, δ)-differential privacy with less noise addition.

In the following of this section, we discuss the key procedures of DPPro, namely, (i) choosing the projection matrix R (and its sensitivity); (ii) choosing the desired privacy guarantees (ϵ, δ), which determines the distribution of the noise; (iii) choosing the optimal projection dimension. It is important to note that the projection matrix as well the noise matrix do not depend on dataset X, but only require the values of the number of users n, the original dimension d and the desired privacy parameters (ϵ, δ).

Algorithm 1 Differentially Private High-Dimensional Data Release via Random Projection Algorithm (DPPro)

Input: $n \times d$ matrix X whose rows correspond to people and columns correspond to attributes learned about the users; Privacy parameters ϵ, δ; Projected dimension k.

Output: $d \times k$ projection matrix R; differential privacy $n \times k$ matrix P, both of which can be released.

1: Sample each entry of the random projection matrix R drawn independently from a Gaussian distribution $N(0, 1/k)$;
2: Construct random $d \times k$ projection matrix R;
3: Compute $Y := XR$;
4: Select noise parameter σ;
5: Construct random $n \times k$ noise matrix ϕ, based on privacy parameters ϵ, δ and projection matrix R;
6: Compute differential privacy $n \times k$ matrix $P := Y + \phi$;
7: **return** (R, P).

B. Choosing of Random Projection Matrix

There are many ways to choose a projection matrix for dimensionality reduction, depending on the properties of the data that need to be preserved. Our choice of projection matrix R is guided by two considerations: (i) we would like to preserve pairwise L_2 distances and thus users segmentation based on these distances, (ii) we would like to minimize the amount of noise to be added in order to maximize utility while guaranteeing privacy.

To preserve L_2 distances between vectors, the candidate projection matrices are the random projection matrices satisfying Johnson-Lindenstrauss Lemma 1, where we suppose each entry of the matrix drawn independently from a Gaussian distribution $N(\mu, \sigma^2)$ with $\mu = 0$ and $\sigma^2 = 1/k$. Since noise ϕ follows a Gaussian distribution, the amount of added noise needed to preserve differential privacy depends on the L_2-sensitivity of the chosen projection matrix R. Therefore, it is desirable to adopt a projection matrix with low L_2-sensitivity, in order to minimize the amount of added noise, that can maximize utility of the released data while preserve privacy. The expected L_2-sensitivity of all of the random projection matrices described above is tightly concentrated around 1 (using the alternative definition of $\max_{e_i} \|e_i R\|_2$, where $\{e_i\}_{i=1}^d$ are standard basis unit vectors, and by applying the proofs of low distortion for these matrices), all of them are suitable for random projection that aim to preserve the maximum utility.

We analyze that the specific measure of the sensitivity of matrix R, namely L_2-sensitivity, which is determined by the amount of added noise to achieve different privacy, rather than by choosing of L_1 norm seeking to preserve pairwise distance under random projection.

C. Choosing the Random Noise Matrix

The choice of the desired privacy guarantees and projection matrix R determines the noise matrix ϕ. Each entry in ϕ is

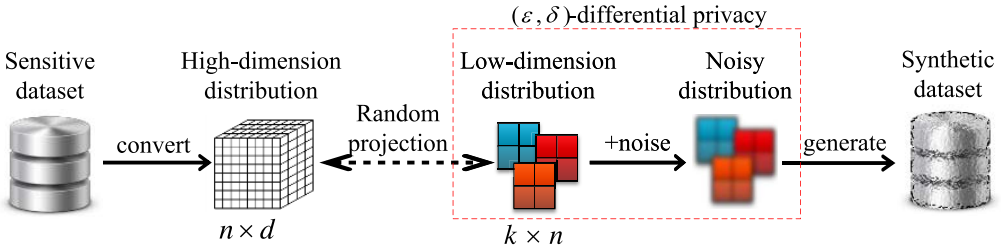

Fig. 1. The framework of DPPro.

drawn randomly and independently from Gaussian distribution $N(0, \sigma^2)$, where the variance of the noise depends on L_2-sensitivity of the projection matrix R and the privacy parameters σ and δ. By carefully choosing σ to satisfy the algorithm, DPPro can guarantee (ϵ, δ)-differential privacy.

A classic result in differential privacy shows that any function can be computed with (ϵ, δ)-differential privacy, as long as the Gaussian noise calibrated according to the L_2-sensitivity of that function is added to the true function value prior to its release [27]. Thus, a natural alternative approach that can guarantee (ϵ, δ)-differential privacy and preserve pairwise L_2 distances between vectors, is to add properly calibrated noise to the true distances.

D. Choosing the Optimal Projection Dimension

Intuitively, for a fixed level of privacy budget, there are two factors that affect the utility of the released data as we vary the projected dimension k. On the one hand, as k gets smaller, dimensionality reduction plays a greater role in the distortion of distances. On the other hand, as k gets larger, noise added plays a greater role in the distortion of distances. However, finding the optimal projected dimension k is a theoretically challenging problem, as it depends on the underlying data distributions and the specific distance we expect to preserve.

To find the tradeoff between the reduced dimensionality and added noise, we should adopt an approach for finding the optimal projection dimension k_{opt} for a fixed σ. The optimal projection dimension k_{opt} can be denoted as follow,

$$k_{opt} = \frac{\|y - x\|_2^2}{2\sigma^2}, \tag{9}$$

which will be theoretically proved in Lemma 5.

The approach implies that k_{opt} depends on the expected distance between vectors and the setting of noise σ. This approach is applied in our solution by using different projection matrices with different k for a particular range of distances.

V. Analysis on Privacy and Utility Guarantees

In the section, we theoretically analyze and prove the privacy and utility guarantees of DPPro.

A. Privacy Guarantees of DPPro

To prove that DPPro can satisfy with (ϵ, δ)-differential privacy, we first analyze which stages in DPPro will consume the privacy budget. We can prove a more general geometric statement, which will be used to prove the privacy guarantees of our algorithm. Lemma 3 extends the result of Dwork *et al.* [27] to high dimensions.

Lemma 3: Let X and X' be any two vectors, and $X \subset \mathbb{R}^d$, $X' \subset \mathbb{R}^k$, $\|X - X'\|_2 \leq \theta_2(R)$. Then for any $O \subset \mathbb{R}^d$, and any noise ϕ draw from $N^d(0, \sigma^2)$, where $\sigma \geq \theta_2(R)\frac{\sqrt{2(\ln \frac{1}{2\delta} + \epsilon)}}{\epsilon}$ and $\delta < \frac{1}{2}$, the following inequality holds: $Pr\left[A\left(X' + \phi\right) \in O\right] \leq \exp(\epsilon) \cdot Pr\left[A\left(X + \phi\right) \in O\right] + \delta$.

Proof: The detailed proof of Lemma 3 is shown as APPENDIX A. □

A significant feature of the DPPro is that the amount of added noise in order to satisfy privacy guarantees depends on the sensitivity $\theta_2(R)$ on R's dimension, rather than the dimensions of the projection matrix R , which is crucial for privacy guarantees of DPPro.

Theorem 2: DPPro satisfies (ϵ, δ)-differential privacy with regard to a change in an individual attribute, if $\delta < \frac{1}{2}$ and the entries of the noise matrix are sampled from $N(0, \sigma^2)$ with $\sigma \geq \theta_2(R)\frac{\sqrt{2(\ln(1/2\delta) + \epsilon)}}{\epsilon}$.

Proof: In order to prove that DPPro satisfies (ϵ, δ)-differential privacy, we need to prove that one element difference in matrices M and M' will affect only one row of the projection. It means that, for any two input matrices M and M' differing in one element m_{aj} (corresponding to user a having a binary value for attribute j), and for any O, where O is a set of possible outputs of DPPro, namely a set of $n \times k$ matrices, we should prove that the following inequality holds, i.e.,

$$Pr\left[MR + \phi \in O\right] \leq Pr\left[M'R + \phi \in O\right] + \delta,$$

where ϕ is a $n \times k$ noise matrix with each element drawn independently and randomly from $N(0, \sigma^2)$.

Given M and M', and define V and V' as flattened vectors with length $n * k$, we can determine

$$\begin{aligned} \left\|V - V'\right\|_2 &= \left\|MR - M'R\right\|_2 \\ &= \left\|\left(M - M'\right)R\right\|_2 \\ &\leq \max_{1 \leq i \leq d} \sqrt{\sum_{j-1}^{k} R_{ij}^2} = \theta_2(R), \end{aligned}$$

if M and M' are binary and $\left\|M - M'\right\|_2 = 1$. Applying the result of Lemma 3 to V and V', we can obtain the desired privacy guarantees.

Therefore, if $\sigma \geq \theta_2(R)\frac{\sqrt{2(\ln(1/2\delta) + \epsilon)}}{\epsilon}$, DPPro can satisfy (ϵ, δ)-differential privacy with regard to a change in an individual attribute. □

B. Utility Guarantees of DPPro

In this subsection, we discuss the utility guarantees provided by DPPro. DPPro can preserve the squared Euclidean distance between two vectors in expectation by privacy transformations. Moreover, it can further provide utility guarantees on how far the distance after random projection can deviate from the original distance. Specifically, from a data requester perspective, these guarantees should meet the following requirements:

(i) The vectors which are close in the original space are likely to remain close in the projection space.
(ii) Similarly, the vectors which are far apart are likely to remain far apart after random projection.

To estimate the squared distance between two vectors, we present Algorithm 2 that can compute the squared L_2 distance between the transformed representations in the k dimensional space, and the discount $2k\sigma^2$ representing the expected distortion in the squared distance due to Gaussian noise addition.

Algorithm 2 Estimating the Squared Distance Between Two Vectors After Random Projection

Input: Differential privacy $n \times k$ matrix P; Privacy parameters ϵ, δ; Projected dimension k; Noise parameter σ.
Output: Estimating the squared distance between two vectors x_a and y_b in original space.
1: **for** i, j=1 to n **do**
2: Let $\tilde{x}$ and $\tilde{y}$ be the ith and jth rows in P, respectively;
3: Select privacy parameters ϵ, δ, projection dimension k, noise parameter σ;
4: Compute $U^2(x_i, y_j) = \|\tilde{x} - \tilde{y}\|_2^2 - 2k\sigma^2$.
5: **end for**
6: **return** $U^2(x_i, y_j)$

Actually, the utility guarantees of DPPro depend on the type of the projection matrix R and privacy parameters ϵ, δ. With regard to the possible choices for projection matrices R, we analyze the guarantees afforded by the use of the Gaussian projection matrix according to the approach proposed by Indyk and Motwani [34]. We also can prove that the resulting estimate of the squared Euclidean distance is unbiased, and can compute its variance and a tail probability bound.

According to Theorem 2, DPPro satisfies (ϵ, δ)-differential privacy and σ can be determined by a function of ϵ, δ, and R. The following Lemma bounds σ under a given setting of ϵ, δ and k.

Lemma 4: Let projection matrix R be a $d \times k$ matrix whose entries are independent and identically distributed random variables drawn from $N(0, 1/k)$. By using a noise matrix whose entries are sampled from $N(0, \sigma^2)$, DPPro can satisfy (ϵ, δ)-differential privacy if

$$\begin{cases} \sigma \geq \dfrac{4}{\epsilon}\sqrt{\ln(1/\delta)}, \\ k > 2(\ln d + \ln(2/\delta)), \\ \epsilon < \ln(1/\delta). \end{cases}$$

Proof: The detailed proof of Lemma 4 is shown as APPENDIX B. □

Lemma 4 implies that the value of σ can be chosen independently from R. This property, which repeatedly uses the independence of the matrix R and the noise ϕ depending on the scale of σ, is important for proving the following theorem.

Theorem 3: If the entires of R are independently sampled from $N(0, 1/k)$, DPPro can satisfy the following utility guarantees:

(i) *U_{RP}^2 is an unbiased estimator of $\|x - y\|_2^2$,*

$$E[U_{RP}^2(x, y)] = \|x - y\|_2^2. \tag{10}$$

(ii) *Variance of U_{RP}^2 is given by the following expression,*

$$D\left[U_{RP}^2(x, y)\right] = \frac{2}{k}\|x - y\|_2^4 + 8\sigma^2\|x - y\|_2^2 + 8\sigma^4 k \tag{11}$$

Proof: First, we note that the random noise ϕ follows a k-dimensional Gaussian distribution, i.e., $N^k(0, \sigma^2)$. Let $\phi = \phi_1 - \phi_2$, then we have

$$\begin{aligned} U_{RP}^2(x, y) &= \|\tilde{x} - \tilde{y}\|_2^2 - 2k\sigma^2 \\ &= \|xR + \phi_1 - yR - \phi_2\|_2^2 - 2k\sigma^2 \\ &= \|(x - y)R + \phi\|_2^2 - 2k\sigma^2 \\ &= \|(x - y)R\|_2^2 + 2\langle (x - y)R, \phi\rangle + \|\phi\|_2^2 - 2k\sigma^2 \\ &= \|(x - y)R\|_2^2 + 2\langle (x - y)R, \phi\rangle + \|\phi\|_2^2 - 2k\sigma^2. \end{aligned}$$

For a fixed user vectors x and y, let $z = x - y = (z_1, ..., z_d)$, and $\rho = \|x - y\|_2$. Since the entries of R are independently drawn from $N(0, 1/k)$, the projection $(x - y)R$ should follow $N^k(0, \rho^2/k)$. Then, the i-th entry of $(x - y)R$ follows a distribution as

$$\sum_{i=1}^{d} z_i N(1, 1/k) \sim \sum_{i=1}^{d} N(0, z_i/k) \sim N(0, \rho^2/k).$$

Supposed that $\|(x - y)R\|_2^2 = Q_1$, $2\langle (x - y)R, \phi\rangle = Q_2$, and $\|\phi\|_2^2 - 2k\sigma^2 = Q_3$. Based on the above expression, we can denote the variables Q_1, Q_2, Q_3 as

$$\begin{aligned} Q_1 &\sim \left\| N^k(0, \|x - y\|_2^2/k) \right\|_2^2 = \rho^2 \cdot \chi_k^2, \\ Q_2 &\sim N(0, 8\sigma^2 Q_1), \\ Q_3 &\sim 2\sigma^2\chi_k^2 - 2k\sigma^2, \end{aligned}$$

where χ_k^2 is the chi-squared distribution with k degrees of freedom. Then, we can calculate the expectation of $U_{RP}^2(x, y)$ as

$$\begin{aligned} &E[U_{RP}^2(x, y)] \\ &= E[\|(x - y)R\|_2^2 + 2\langle (x - y)R, \phi\rangle + \|\phi\|_2^2 - 2k\sigma^2] \\ &= E[\|(x - y)R\|_2^2] + E[2\langle (x - y)R, \phi\rangle] + E[\|\phi\|_2^2 - 2k\sigma^2] \\ &= E[\rho^2 \cdot \chi_k^2/k] + 0 + 0 \\ &= \rho^2. \end{aligned}$$

It proves that U_{RP}^2 is an unbiased estimator of $\|x - y\|_2^2$, $E[U_{RP}^2(x, y)] = \|x - y\|_2^2$.

Then, we can compute the variance of $U_{RP}^2(x, y)$ as

$$\begin{aligned} D\left[U_{RP}^2(x,y)\right] &= D(Q_1+Q_2+Q_3) \\ &= E\left[(Q_1+Q_2+Q_3)^2\right] - (E[Q_1+Q_2+Q_3])^2 \\ &= E\left[Q_1^2+Q_2^2+Q_3^2+2Q_1Q_2+2Q_2Q_3+2Q_1Q_3\right] \\ &\quad - (E[Q_1]+E[Q_2]+E[Q_3])^2. \end{aligned}$$

Since σ is chosen independently from R, $(x-y)R$ and ϕ are independent. We have,

$$\begin{aligned} E[Q_1Q_2] &= E[\|(x-y)R\|_2^2 \cdot 2\langle (x-y)R, \phi\rangle] \\ &= E[\langle \|(x-y)R\|_2^2 \cdot 2(x-y)R, \phi\rangle] = 0, \\ E[Q_2Q_3] &= E[2\langle (x-y)R, \phi\rangle \cdot \|\phi\|_2^2 - 2k\sigma^2] \\ &= E[2\langle (x-y)R, \phi \cdot \|\phi\|_2^2 - 2k\sigma^2\rangle] = 0, \\ E[Q_1Q_3] &= E[Q_1]E[Q_3] = 0. \end{aligned}$$

Since $D(\chi_k^2) = 2k$ and $Q_2 = 2\langle (x-y)R, \phi\rangle$, where $(x-y)R \sim N^k(0, \rho^2/k)$ and $\phi \sim N^k(0, 2\sigma^2)$, we have

$$\begin{aligned} E[Q_1^2] - E[Q_1]^2 &= D(Q_1) = D(\rho^2 \cdot \chi_k^2) = \frac{\rho^4}{k^2} D(\chi_k^2) = \frac{2\rho^4}{k}, \\ E[Q_3^2] - E[Q_3]^2 &= D(Q_3) = D(2\sigma^2\chi_k^2 - 2k\sigma^2) \\ &= 4\sigma^2 D(\chi_k^2) = 8k\sigma^4, \\ E[Q_2^2] - E[Q_2]^2 &= D(Q_2) = D(2\sum_{i=1}^{k} N(0, \rho^2/k), N(0, 2\sigma^2)) \\ &= kD(N(0, \rho^2/k), N(0, 2\sigma^2)) = 8\rho^2\sigma^2. \end{aligned}$$

Putting the above expressions together, we obtain

$$\begin{aligned} D[U_{RP}^2(x,y)] &= \frac{2\rho^4}{k} + 8\rho^2\sigma^2 + 8k\sigma^4. \\ &= \frac{2}{k}\|x-y\|_2^4 + 8\sigma^2\|x-y\|_2^2 + 8\sigma^4 k. \end{aligned}$$

which completes the proof of theorem 3. □

C. The Proof of Lemma 5

Lemma 5: To minimize the variance of the squared distance estimate returned by DPPro, the optimal projection dimension k_{opt} can be denoted,

$$k_{opt} = \frac{\|y-x\|_2^2}{2\sigma^2}.$$

Proof: According to Theorem 3, the variance of U_{RP}^2 is given by the following expression,

$$D\left[U_{RP}^2(x,y)\right] = \frac{2}{k}\|x-y\|_2^4 + 8\sigma^2\|x-y\|_2^2 + 8\sigma^4 k$$

Supposed $\rho = \|x-y\|_2$, we have

$$D[U_{RP}^2(x,y)] = \frac{2\rho^4}{k} + 8\rho^2\sigma^2 + 8k\sigma^4.$$

To minimize the variance of the squared distance estimate $\min(D[U_{RP}^2(x,y)])$, and $k \neq 0$, we have

$$\frac{\partial D[U_{RP}^2(x,y)]}{\partial \rho^2} = \frac{2}{k} \cdot 2\rho^2 + 8\sigma^2 = 0$$

when $\rho^2 = \|y-x\|_2^2 = 2k\sigma^2$, we can attain the minimum of $D[U_{RP}^2(x,y)]$. Therefore, the optimal projection dimension k_{opt} can be denoted,

$$k_{opt} = \frac{\|y-x\|_2^2}{2\sigma^2},$$

which completes the proof of lemma 5. □

VI. Experimental Evaluation

In the section, we evaluate the performance of DPPro by comparing with three existing solutions, namely JTree [5], PrivBayes [8] and PriView [7]. Moreover, we compare DPPro to PrivateSVM [35] in terms of the performance of SVM classification.

A. Datasets and Configuration

Our experiments are based on six standard real including binary and non-binary datasets. For binary datasets, we deliberately choose the *AOL*[1] and *Retail*[2] with larger domain sizes to evaluate the performance of DPPro. *AOL* is a search log dataset that includes users'search keywords and is preprocessed to contain 45 binary attributes. *Retail* is a *Retail* market basket dataset, where each record consists of the distinct items purchased in a shopping visit. We preprocess *Retail* to include 50 binary attributes (for the reason of reproducibility, we choose the top 50 most frequent items as the binary attributes). For non-binary datasets, we use the same datasets including *BR2000*[3] and *Adult*[4] used as [8]. *BR2000* contains the demographics information collected from Brazil in 2000. The *Adult* dataset from the UCI Machine Learning repository originally has 48,842 records and 14 attributes. After deleting the missing records, we eventually have 30,162 records. To evaluate the performance of SVM classification by DPPro, both *TPC-E*[5] and *NLTCS*[6] are adopted. *TPC-E* contains the information of "Trade", "Security", "Security status" and "Trade type" tables in the *TPC-E* benchmark, while *NLTCS* contains records of 21,574 individuals participated in the National Long Term Care Survey. We summarize the statistics of the datasets in Table II.

B. Evaluation Methodology

To evaluate to the performance of DPPro, for each dataset, we generate a query set Q with 10,000 random linear queries, and report the average total variation distance between the original datasets and the noisy datasets. The utility of released data was measured by average L_2 Error.

$$L_2\ Error = \frac{\|\ U^2(x,y) - \tilde{U}^2(x,y)\ \|_2^2}{\|\ U^2(x,y)\ \|_2^2}, \tag{12}$$

where $U^2(x, y)$ is the squared distance between two users in original space, while $\tilde{U}^2(x, y)$ is the squared distance between

[1] AOL search log dataset, http://www.gregsadetsky.com/aol-data/.
[2] Retail, http://fimi.ua.ac.be/data/.
[3] BR2000, https://international.ipums.org.
[4] K. Bache, M. Lichman, https://archive.ics.uci.edu/ml/datasets.html/.
[5] Transaction processing performance council, http://www.tpc.org.
[6] Statlib, http://lib.stat.cmu.edu/.

TABLE II
DATASET CHARACTERISTICS

Datasets	Cardinality	Dimensionality	Domain Size
AOL	619,418	45	2^{45}
Retail	88,162	50	2^{50}
BR2000	38,000	14	$\approx 2^{32}$
Adult	45,222	15	$\approx 2^{52}$
TPC-E	40,000	24	$\approx 2^{77}$
NLTCS	21,514	16	$\approx 2^{16}$

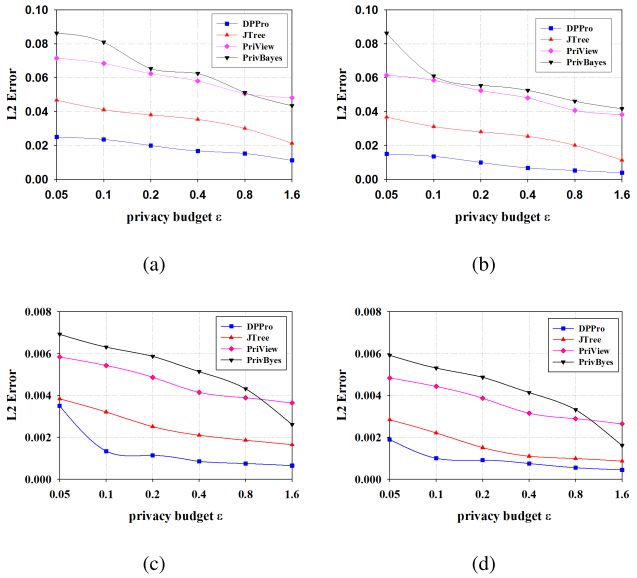

Fig. 2. L_2 Error of DPPro on binary datasets. (a) *Retail:* $\sigma = 0.1$. (b) *Retail:* $\sigma = 0.2$. (c) *AOL:* $\sigma = 0.1$. (d) *AOL:* $\sigma = 0.2$.

two users in projection space. A lower L_2 Error implies a better utility. In addition, we evaluate the classification results with SVM classifiers, as in PrivBayes, we also employ the misclassification rate as the performance metric. All experimental results we report below are the average of 50 times.

C. The Performance of DPPro on Binary Datasets

In the first set of experiments, we compare the performance of the four solutions (e.g., DPPro, JTree, PrivBayes and PriView) on the binary datasets under different privacy budgets, as presented in Fig. 2. The experiments were conducted on two datasets *Retail* and *AOL* with the privacy budget varied from 0.05 to 1.6. It can be seen that the accuracy of DPPro is substantially better than that of JTree and PrivBayes in most cases. Compared with PriView, DPPro also achieves comparable accuracy. The superiority of DPPro is more significant when ϵ is small. It indicates that, as a generic method to publish synthetic datasets, DPPro can achieve an acceptable trade-off between data privacy and utility on binary datasets.

D. The Impact of Projection Dimensionality for DPPro

Since projection dimension k is an important parameter in DPPro, we here evaluate the impact of the projection dimension k on the performance of DPPro, under different σ. Fig. 3 shows the L_2 Errors of DPPro on the binary datasets with

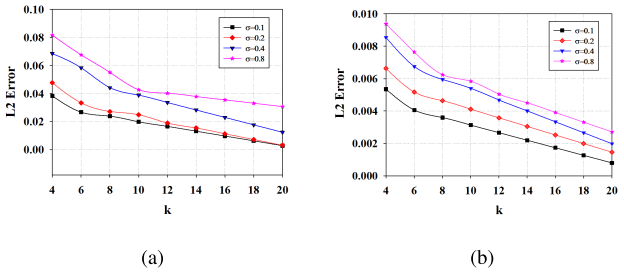

Fig. 3. The impact of projection dimensionality for DPPro. (a) *Retail*. (b) *AOL*.

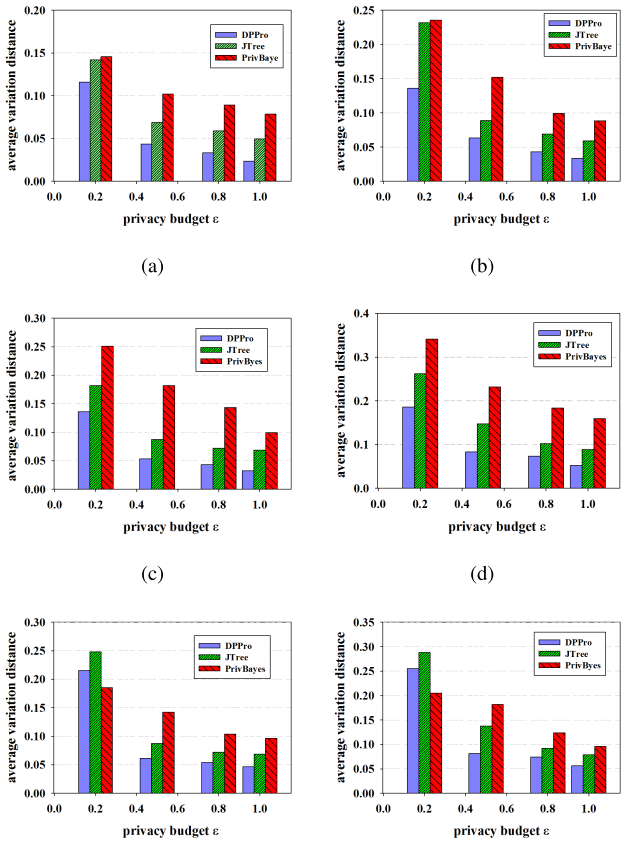

Fig. 4. Average total variation distance between pairs of users on non-binary datasets. (a) *BR2000:* $\sigma = 0.1$. (b) *BR2000:* $\sigma = 0.2$. (c) *Adult:* $\sigma = 0.1$. (d) *Adult:* $\sigma = 0.2$. (e) *TPC-E:* $\sigma = 0.1$. (f) *TPC-E:* $\sigma = 0.2$.

respect to varying projection dimension k under different σ. We set k ranging from 4 to 20. It can be seen that the L_2 Error of DPPro decrease gradually as the number of dimensions increases. This is because, for a fixed σ, the L_2 Error of the squared distance between any pair of users in DPPro tends to be smaller when k gets larger. This result is consistent with our analysis in subsection IV-D and subsection V-C, where the optimal value for target dimension of DPPro depends on the expected distance between vectors measured by this approach.

E. The Performance of DPPro on Non-Binary Datasets

In this subsection, to evaluate the performance of DPPro on non-binary datasets, we compare the average total variation distances of DPPro, PrivBayes and JTree under a varying

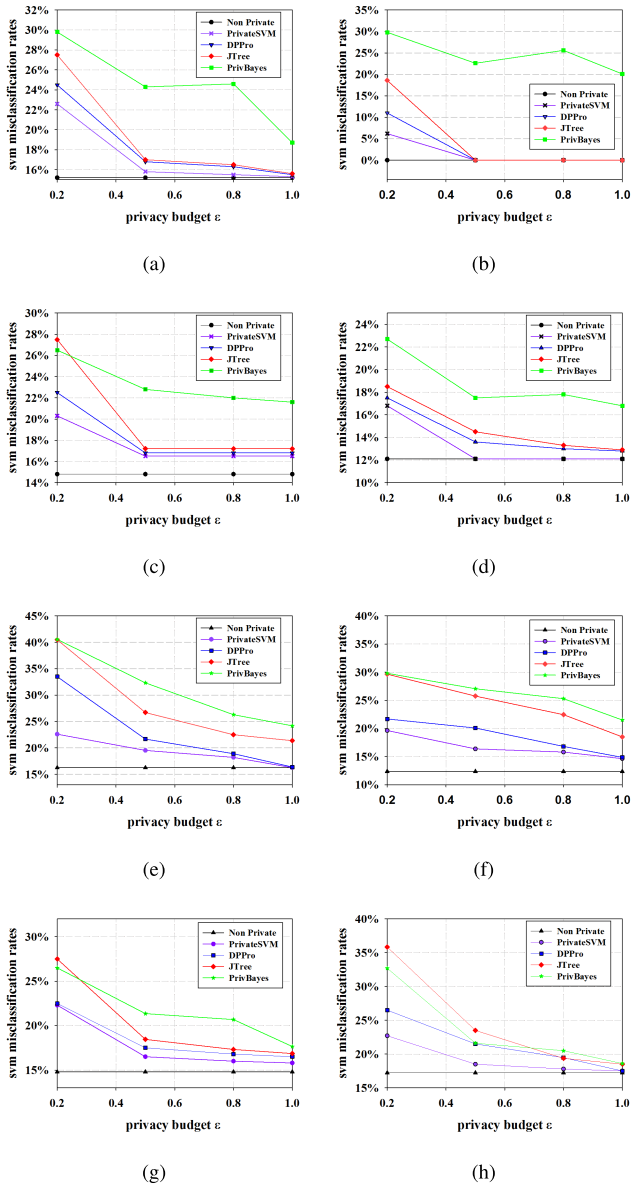

Fig. 5. SVM misclassification rates on non-binary datasets. (a) *Adult*, Y = gender. (b) *Adult*, Y = education. (c) *Adult*, Y = salary. (d) *Adult*, Y = marital. (e) *NLTCS*, Y = outside. (f) *NLTCS*, Y = money. (g) *NLTCS*, Y = bathing. (h) *NLTCS*, Y = traveling.

privacy budget ϵ from 0.2 to 1.0, in Fig. 4. The experiments are performed on *BR2000*, *Adult* and *TPC-E*. Since PriView cannot process non-binary datasets, we did not compare its performance here. As shown in Fig. 4, DPPro substantially outperforms PrivBayes in almost all cases. The only case that PrivBayes performs better than DPPro and JTree is when $\epsilon = 0.2$ on *TPC-E*, but even in such case, the performance of DPPro still outperforms that of JTree. This indicates that the the performance of DPPro is not good on larger domain size datasets.

F. The Performance of DPPro for SVM Classification

To evaluate the performance of DPPro for SVM classification, we also compare DPPro, JTree, PrivBayes, PrivateSVM and Non-Private. Fig. 5 shows the misclassification rates of each solution under different privacy budgets. Here, PrivateSVM denotes the benchmark of SVM classification, while Non-Private denotes the SVM classification scheme without differential privacy. We can observe that DPPro consistently outperforms PrivBayes and JTree on both datasets of *Adult* and *NLTCS*. Moreover, it can be seen from the results on *Adult* that the misclassification rate decreases faster when ϵ increases from 0.2 to 0.5, than when ϵ increases from 0.5 to 1. It indicates that a higher privacy level ($\epsilon = 0.2$) leads to a lower utility. In addition, DPPro can achieve comparable performance when compared to PrivateSVM. It demonstrates that DPPro is capable of retaining the utility of the released data while satisfying a suitable privacy requirement.

VII. Conclusion

Releasing high-dimensional data with differential privacy guarantees is one of the most fundamental and challenging problems for privacy preservation in big data. In this paper, we have proposed DPPro to address the challenges in differentially private high-dimensional data release. It can simultaneously guarantee the utility and privacy of the released data, by preserving the pairwise distances between users with random projection and minimizing the amount of added noise. Furthermore, we have found that the utility guarantees of the released data by DPPro depend on the projection dimension and the variance of the noise with high probability. Extensive experimental results on a variety of real datasets have validated our theoretical analysis and demonstrated the effectiveness and superior performance of DPPro, particularly on high-dimensional datasets.

For our future work, we plan to extend our research to differentially private large domain data release with the consideration of computation efficiency. Since the large domain space incurs a high computation complexity both in time and space, and it is infeasible to read all data with large attribute domains into memory simultaneously due to memory constraints.

Appendix A
Proof of Lemma 3

Proof: According to spherical symmetry properties of Gaussian noise, we suppose that X and X' differ in exactly one dimension. Then, we divide O into two sets of vectors

$$O_1 = \left\{\hat{O} \in O : \langle X' - X, \hat{O} - X' \rangle \leq K\theta_2(R)\right\}, \quad (13)$$

$$O_2 = \left\{\hat{O} \in O : \langle X' - X, \hat{O} - X' \rangle > K\theta_2(R)\right\}, \quad (14)$$

where K is an introduced parameter to assist the proof of this lemma. Such that, we can complete the proof by choosing σ to make R satisfy both constraints (13) and (14), summarizing the resulting inequalities, and observing that $Pr\left[A\left(X+\phi\right) \in O_1\right] \leq Pr\left[A\left(X'+\phi\right) \in O\right]$. First, we should prove that

$$Pr\left[X' + \phi \in O_1\right] \leq e^{\epsilon} \cdot Pr\left[X + \phi \in O_1\right], \quad if \ K \geq \frac{2\epsilon\sigma^2 - [\theta_2(R)]^2}{2\theta_2(R)}. \quad (15)$$

and then prove that

$$Pr\left[X'+\phi\in O_2\right]\le\delta,\ if\ K\ge\sigma\sqrt{2\ln(\frac{1}{2\delta})} \tag{16}$$

(i) Proof of inequality (15). If $K\ge\frac{2\epsilon\sigma^2-[\theta_2(R)]^2}{2\theta_2(R)}$, according to the definition of the Gaussian noise, we have

$$Pr\left[X'+\phi\in O_1\right]=\frac{1}{(\sqrt{2\pi}\sigma)^l}\int_{O_1}\exp\left(-\frac{1}{2\sigma^2}\left\|X'-z\right\|_2^2\right)dz$$

The density of function restricted to O_1 satisfies

$$\begin{aligned}
&\exp\left(-\frac{1}{2\sigma^2}\left\|X'-z\right\|_2^2\right)\\
&=\exp\left(-\frac{1}{2\sigma^2}\left(\|X-z\|_2^2-\left\|X-X'\right\|_2^2-2\langle X-X',z-X'\rangle\right)\right)\\
&=\exp\left(-\frac{1}{2\sigma^2}\|X-z\|_2^2\right)\\
&\quad\cdot\exp\left(\frac{1}{2\sigma^2}\left(\left\|X-X'\right\|_2^2+2\langle X-X',z-X'\rangle\right)\right)\\
&\le\exp\left(-\frac{1}{2\sigma^2}\|X-z\|_2^2\right)\cdot\exp\left(\frac{[\theta_2(R)]^2+2K\theta_2(R)}{2\sigma^2}\right)\\
&\le\exp\left(-\frac{1}{2\sigma^2}\|X-z\|_2^2\right)\cdot\exp(\epsilon)
\end{aligned}$$

It implies that

$$\begin{aligned}
&Pr\left[X'+\phi\in O_1\right]\\
&\quad=\frac{1}{(\sqrt{2\pi}\sigma)^l}\int_{O_1}\exp\left(\frac{1}{2\sigma^2}\left\|X'-z\right\|_2^2\right)dz\\
&\quad\le\frac{1}{(\sqrt{2\pi}\sigma)^l}\int_{O_1}\exp\left(-\frac{1}{2\sigma^2}\|X-z\|_2^2\right)\cdot\exp(\epsilon)\,dz\\
&\quad\le\exp(\epsilon)\cdot Pr\left[X+\phi\in O_1\right].
\end{aligned}$$

(ii). Proof of inequality (16). Supposed that

$$K\ge\sigma\sqrt{2\ln(\frac{1}{2\delta})},$$

and we define a coordinate system that $X=\{x_1,...,x_l\}$ and $X'=\left\{x_1',...,x_l'\right\}$ differ only in the first coordinate and $x_1'<x_1$. Then, we have

$$\begin{aligned}
O_2&=\left\{\hat{O}\in O:\langle X'-X,\hat{O}-X'\rangle>K\theta_2(R)\right\}\\
&\subseteq\left\{z\in\mathbb{R}^l:\left(X_1'-X_1\right)\left(z_1-X_1'\right)>K\theta_2(R)\right\}
\end{aligned}$$

It implies that the following bound on the probability of $X'+\phi$ falls inside O_2, which means

$$\begin{aligned}
&Pr\left[X'+\phi\in O_2\right]\\
&=\frac{1}{(\sqrt{2\pi}\sigma)^l}\int_{O_2}\exp\left(-\frac{1}{2\sigma^2}\left\|X'-z\right\|_2^2\right)dz\\
&\le\frac{1}{(\sqrt{2\pi}\sigma)^l}\int_{(X_1'-X_1)(z_1-X_1')>K\theta_2(R)}\int_{-\infty}^{+\infty}...\\
&\quad\int_{-\infty}^{+\infty}\exp\left(-\frac{1}{2\sigma^2}\left\|X'-z\right\|_2^2\right)dz_1...dz_l\\
&=\frac{1}{(\sqrt{2\pi}\sigma)^l}\int_{(X_1'-X_1)(z_1-X_1')>K\theta_2(R)}\prod_{i=1}^{l}\exp\left(-\frac{1}{2\sigma^2}\left(x_i'-z_i\right)^2\right)dz_1...dz_l\\
&=\frac{1}{(\sqrt{2\pi}\sigma)}\int_{(X_1'-X_1)(z_1-X_1')>K\theta_2(R)}\exp\left(-\frac{1}{2\sigma^2}\left(x_1'-z_1\right)^2\right)\\
&\quad\times dz_1\\
&=\frac{1}{(\sqrt{2\pi}\sigma)}\int_{-\infty}^{\frac{K\theta_2(R)}{x_1'-x_1}+x_1'}\exp\left(-\frac{1}{2\sigma^2}\left(x_1'-z_1\right)^2\right)dz_1
\end{aligned}$$

If $K\ge\sigma\sqrt{2\ln(\frac{1}{2\delta})}$, the bound follows from $1-erf(\tau)=\exp(-\tau^2)$ and $\tau>0$ [36], then, we have

$$\begin{aligned}
&Pr\left[X'+\phi\in O_2\right]\\
&\quad=\frac{1}{2}\left(1+erf\left(\frac{\frac{K\theta_2(R)}{x_1'-x_1}+x_1'-x_1'}{\sqrt{2}\sigma}\right)\right)\\
&\quad=\frac{1}{2}\left(1-erf\left(\frac{K\theta_2(R)}{\sqrt{2}\sigma\left(x_1-x_1'\right)}\right)\right)\\
&\quad\le\frac{1}{2}\left(\exp\left(\frac{K\theta_2(R)}{\sqrt{2}\sigma\left(x_1-x_1'\right)}\right)^2\right)\\
&\quad\le\frac{1}{2}\exp\left(-\frac{K^2[\theta_2(R)]^2}{2\sigma^2[\theta_2(R)]^2}\right)=\frac{1}{2}\exp\left(-\frac{K^2}{2\sigma^2}\right)\le\delta.
\end{aligned}$$

Therefore, to make inequalities (15) and (16) to hold simultaneously, i.e,

$$Pr\left[A\left(X+\phi\right)\in O\right]\le\exp(\epsilon)\cdot Pr\left[A\left(X'+\phi\right)\in O\right]+\delta.$$

we have

$$0<\sigma\sqrt{2\ln(\frac{1}{2\delta})}\le K\le\frac{2\epsilon\sigma^2-[\theta_2(R)]^2}{2\theta_2(R)}$$

which completes the proof of lemma 3. □

APPENDIX B
PROOF OF LEMMA 4

Proof: According to Theorem 2, $(\epsilon,\delta/2)$-differential privacy is satisfied if $\sigma\ge\theta_2(R)\frac{\sqrt{2(\ln(1/\delta)+\epsilon)}}{\epsilon}$, where $\theta_2(R)$ is the L_2-sensitivity of R. Since the entries of R are Gaussian distribution, its sensitivity $\theta_2(R)$ has the following distribution:

$$\theta_2(R)\sim\sqrt{\max_{1\le i\le d}\mid N\left(0,1/k\right)\mid^2}\sim\sqrt{\max_{1\le i\le d}\frac{Z_i}{k}},$$

where Z_i are independent and identically distributed χ_k^2 variables. Let $\omega=\ln d+\ln(2/\delta)$, then ω should be bounded based on the tail probability of the chi-squared distribution [37]. Thus, we have

$$Pr\left[\theta_2\left(R\right)>1+\sqrt{\frac{2\omega}{k}}\right]<\delta/2.$$

If $k>2\left(\ln d+\ln\left(2/\delta\right)\right)$, then

$$Pr[\theta_2\left(R\right)>2]<\delta/2.$$

According to Theorem 2, we find that DPPro satisfies (ϵ, δ)-differential privacy for $\epsilon < \ln(1/\delta)$ if

$$\sigma \geq \frac{4}{\epsilon}\sqrt{\ln(1/\delta)} > \frac{2}{\epsilon}\sqrt{2(\ln(1/\delta)+\epsilon)}$$

and

$$k > 2\left(\ln d + \ln(2/\delta)\right).$$

which completes the proof. □

REFERENCES

[1] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surv.*, vol. 42, no. 4, pp. 14:1–14:53, Jun. 2010.

[2] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. 3rd Conf. Theory Cryptograph.*, 2006, pp. 265–284.

[3] C. Dwork, "Differential privacy: A survey of results," in *Proc. 5th Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.

[4] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. 48th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2007, pp. 94–103.

[5] R. Chen, Q. Xiao, Y. Zhang, and J. Xu, "Differentially private high-dimensional data publication via sampling-based inference," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 129–138.

[6] H. Li, L. Xiong, and X. Jiang, "Differentially private synthesization of multi-dimensional data using copula functions," in *Proc. Int. Conf. Extending Database Technol.*, 2014, pp. 475–486.

[7] W. Qardaji, W. Yang, and N. Li, "PriView: Practical differentially private release of marginal contingency tables," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 1435–1446.

[8] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "PrivBayes: Private data release via Bayesian networks," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 1423–1434.

[9] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar, "Privacy, accuracy, and consistency too: A holistic solution to contingency table release," in *Proc. 26th ACM SIGMOD-SIGACT-SIGART Symp. Principles Database Syst.*, 2007, pp. 273–282.

[10] B. Ding, M. Winslett, J. Han, and Z. Li, "Differentially private data cubes: Optimizing noise sources and consistency," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2011, pp. 217–228.

[11] N. Mohammed, R. Chen, B. Fung, and P. S. Yu, "Differentially private data release for data mining," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 493–501.

[12] Y. Xiao, J. Gardner, and L. Xiong, "DPCube: Releasing differentially private data cubes for health information," in *Proc. IEEE 28th Int. Conf. Data Eng. (ICDE)*, Apr. 2012, pp. 1305–1308.

[13] G. Acs, C. Castelluccia, and R. Chen, "Differentially private histogram publishing through lossy compression," in *Proc. IEEE 12th Int. Conf. Data Mining (ICDM)*, Dec. 2012, pp. 1–10.

[14] G. Cormode, C. Procopiuc, D. Srivastava, and T. T. L. Tran, "Differentially private summaries for sparse data," in *Proc. 15th Int. Conf. Database Theory*, 2012, pp. 299–311.

[15] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and S. Martínez, "Enhancing data utility in differential privacy via microaggregation-based k-anonymity," *VLDB J.*, vol. 23, no. 5, pp. 771–794, 2014.

[16] D. Sánchez, J. Domingo-Ferrer, S. Martínez, and J. Soria-Comas, "Utility-preserving differentially private data releases via individual ranking microaggregation," *Inf. Fusion*, vol. 30, pp. 1–14, Jul. 2016.

[17] J. Blocki, A. Blum, A. Datta, and O. Sheffet, "The Johnson–Lindenstrauss transform itself preserves differential privacy," in *Proc. IEEE 53rd Annu. Symp. Found. Comput. Sci.*, Oct. 2012, pp. 410–419.

[18] K. Kenthapadi, A. Korolova, I. Mironov, and N. Mishra, "Privacy via the Johnson–Lindenstrauss transform," *J. Privacy Confidentiality*, vol. 5, no. 1, pp. 39–71, 2013.

[19] O. Sheffet, "Differentially private ordinary least squares: T-values, confidence intervals and rejecting null-hypotheses," in *Proc. TPDP*, 2016, pp. 1–14.

[20] O. Sheffet, "Private approximations of the 2nd-moment matrix using existing techniques in linear regression," in *Proc. NIPS Workshop Learn. Privacy*, 2015, pp. 1–9.

[21] J. Upadhyay. (2016). "On low-space differentially private low-rank factorization in the spectral norm." [Online]. Available: https://arxiv.org/abs/1611.08954

[22] J. Upadhyay. (2016). "Fast and space-optimal low-rank factorization in the streaming model with application in differential privacy." [Online]. Available: https://arxiv.org/abs/1604.01429

[23] J. Upadhyay. (2014). "Randomness efficient fast-Johnson–Lindenstrauss transform with applications in differential privacy and compressed sensing." [Online]. Available: https://arxiv.org/abs/1410.2470

[24] S. Su, P. Tang, X. Cheng, R. Chen, and Z. Wu, "Differentially private multi-party high-dimensional data publishing," in *Proc. IEEE 32nd Int. Conf. Data Eng.*, May 2016, pp. 205–216.

[25] O. Williams and F. Mcsherry, "Probabilistic inference and differential privacy," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23. 2010, pp. 2451–2459.

[26] W.-Y. Day and N. Li, "Differentially private publishing of high-dimensional data using sensitivity control," in *Proc. 10th ACM Symp. Inf., Comput. Commun. Secur.*, 2015, pp. 451–462.

[27] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2006, pp. 486–503.

[28] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *Proc. 51st Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2010, pp. 51–60.

[29] C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor, "Optimizing linear counting queries under differential privacy," in *Proc. 29th ACM SIGMOD-SIGACT-SIGART Symp. Principles Database Syst.*, 2010, pp. 123–134.

[30] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to non-interactive database privacy," in *Proc. 14th Annu. ACM Symp. Theory Comput.*, 2008, pp. 609–618.

[31] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipshitz mapping into Hilbert space," *Contemp. Math.*, vol. 26, no. 1, pp. 189–206, 1984.

[32] S. Dasgupta and A. Gupta, "An elementary proof of a theorem of Johnson and Lindenstrauss," *Random Struct. Algorithms*, vol. 22, no. 1, pp. 60–65, 2003.

[33] R. I. Arriaga and S. Vempala, "An algorithmic theory of learning: Robust concepts and random projection," in *Proc. 40th Annu. Symp. Found. Comput. Sci.*, Oct. 1999, pp. 616–623.

[34] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. 13th Annu. ACM Symp. Theory Comput. (STOC)*, May 1998, pp. 604–613.

[35] B. I. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, "Learning in a large function space: Privacy-preserving mechanisms for SVM learning," *J. Privacy Confidentiality*, vol. 4, no. 1, pp. 65–100, 2012.

[36] M. Chiani, D. Dardari, and M. K. Simon, "New exponential bounds and approximations for the computation of error probability in fading channels," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 840–845, Jul. 2003.

[37] B. Laurent and P. Massart, "Adaptive estimation of a quadratic functional by model selection," *Ann. Statist.*, vol. 28, no. 5, pp. 1302–1338, 2000.

Chugui Xu (S'16) received the B.S. degree in physics from Hunan Normal University, China, in 2005, and the M.S. degree in computer science from the Hunan University of Technology, China, in 2010. He is currently pursuing the Ph.D. degree with Central South University, China. His research interests include Internet of Things, transparent computing, data privacy, and security.

Ju Ren (S'13–M'16) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Central South University, China, in 2009, 2012, and 2016, respectively. From 2013 to 2015, he was a Visiting Ph.D. Student with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently a Professor with the School of Information Science and Engineering, Central South University. He has published over 30 peer-reviewed papers in some prestigious international journals and conferences, including the IEEE TIFS, TWC, TII, TVT, and TETC. His research interests include wireless sensor network, mobile sensing/computing, transparent computing, and cloud computing. He is a member of ACM and CCF. He is serving on the Editorial Board of *Peer-to-Peer Networking and Applications*, and severed as a leading Guest Editor of the IEEE NETWORK, the Track Chair of IEEE VTC 2017 fall, and a TPC Member of many IEEE conferences, including IEEE INFOCOM 18, Globecom 16–17, and WCNC 17.

Yaoxue Zhang received the B.Sc. degree from the Northwest Institute of Telecommunication Engineering, China, in 1982, and the Ph.D. degree in computer networking from Tohoku University, Japan, in 1989. He is currently a Professor with the Department of Computer Science, Central South University, China, and also a Professor with the Department of Computer Science and Technology, Tsinghua University, China. He has published over 200 technical papers in international journals and conferences, and nine monographs and textbooks. His research interests include computer networking, operating systems, ubiquitous/pervasive computing, transparent computing, and big data. He is a fellow of the Chinese Academy of Engineering. He is currently serving as the Editor-in-Chief of the *Chinese Journal of Electronics*.

Zhan Qin is currently pursuing the Ph.D. degree with the Director of the Ubiquitous Security and Privacy Research Laboratory, Computer Science and Engineering Department, State University of New York at Buffalo, NY, USA. His research interests focus on data privacy, crowdsourcing security, and smart grid.

Kui Ren (F'16) received the Ph.D. degree from the Worcester Polytechnic Institute. He is currently a Professor of computer science and engineering and the Director of the UbiSeC Lab, State University of New York at Buffalo (UB). He has published 200 papers in peer-reviewed journals and conferences. His current research interest spans cloud and outsourcing security, wireless and wearable systems security, and mobile sensing and crowdsourcing. He is a Distinguished Lecturer of the IEEE, a member of ACM, and a Past Board Member of Internet Privacy Task Force, State of Illinois. He received several best paper awards, including IEEE ICDCS 2017, IWQoS 2017, and ICNP 2011. He received the IEEE CISTC Technical Recognition Award in 2017, the UB Exceptional Scholar Award for Sustained Achievement in 2016, the UB SEAS Senior Researcher of the Year Award in 2015, the Sigma Xi/IIT Research Excellence Award in 2012, and the NSF CAREER Award in 2011. He currently serves on the editorial boards of the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON SERVICE COMPUTING, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE WIRELESS COMMUNICATIONS, the IEEE INTERNET OF THINGS JOURNAL, and *SpingerBriefs on Cyber Security Systems and Networks*.