Differentially Private Principal Component Analysis Over Horizontally Partitioned Data

Sen Wang
Department of Electrical Engineering
University of South Florida
Tampa, Florida 33620
Email: senwang@mail.usf.edu

J. Morris Chang
Department of Electrical Engineering
University of South Florida
Tampa, Florida 33620
Email: chang5@usf.edu

***Abstract*—Principal Component Analysis (PCA) is widely adopted in various data mining and machine learning applications, it computes a low dimension subspace that captures the most variances of the underlying data. The area of distributed computing provides a promising domain for PCA, where it has been studied in many fields. In big data era, large volume and high dimensional data are generated at all times. For instance, mobile devices become the important producer and carrier for personal information, which can provide a considerable social utility. However, the current distributed PCA protocol cannot provide the efficiency and scalability with respect to such large amounts of data. Furthermore, the privacy issue arises when data contains sensitive information. The data owner would not prefer to sharing the data in cleartext, and the inference from PCA should also be prevented. Motivated to resolve these challenges, in this paper, we design and implement a highly efficient and largely scalable privacy preserving distributed PCA protocol, in which the (ϵ, δ)-Differential Privacy is guaranteed. In the experiments, we evaluate the protocol in terms of efficiency and utility, and shows that it maintains a high data utility while preserving the privacy.**

I. Introduction

Principal Component Analysis (PCA) [1] is a statistical procedure which computes a low dimension subspace that captures most variances of the underlying data, it generates a new set of variables that are linear combination of the original ones. It has a widely usage in various data mining and machine learning applications, such as network intrusion detection [2], recommendation system [3], text and image data processing [4] [5]. The area of distributed computing brings a promising domain for machine learning methods, and PCA over distributed data has been studied in many fileds, like distributed database [6], distributed sensor networks [7] [8] [9], distributed social networks [10]. However, data explosion never stops. In the big data era, large volume and high dimensional data is generated at all times. For instance, mobile devices become the important producer and carrier for personal information, such as images, video, text messages and activity recording, and such large amounts of "Mobile Databases" can provide a considerable social utility. For example, in Boston Marathon bombing [11], through face recognition, on-site images could help authorities to quickly identify suspects. In such cases, the distributed PCA protocol should be highly efficient and largely scalable.

Furthermore, privacy concerns arise with personal data. Individuals would not prefer to share the data in cleartext, which exposes the content directly. The AOL search engine log [12] and Netflix prize contest [13] privacy attacks point out the severity of privacy leaking and make people hesitate to share the data. Differential Privacy [14] [15] [16] [17] is one of the most popular schemes for privacy protection in the last decade, which prevents the inference about individuals from participation of computation. It defines a mechanism that the computation of the data is robust to any change of any individual sample by introducing uncertainty into the algorithm. To overcome the above issues, in this paper, we design and implement a highly efficient and largely scalable distributed PCA protocol, in which the protocol satisfies the (ϵ, δ)-differential privacy.

To the best of our knowledge, the only privacy preserving distributed PCA protocol is proposed by Imtiaz et al. [18]. The author approximates the global PCA by aggregating the local PCA from each data owner, in which the data owner holds a horizontally partitioned data. However, their work suffers from an excessive running time and a utility degradation while the local principal components fail to provide a good representation of the data. More specifically, their solution requires all data owner be on line and transferring the local PCA one by one. The serialized computation fashion makes their protocol running time depends on the number of data owners, which cannot provide the efficiency and scalability in scenarios like emergency response. For the utility of the principal components, the local principal components cannot provide a good representation when the amount of data is much less than its number of feature. For instance, the preference regarding a list of the music and movies, the personal activity data, the medical records. Given such type of data, the approximate PCA fails to maintain a good utility.

In the paper, we assume the data is horizontally partitioned, in which all data share the same features. The magnitude of data owners would be more than hundreds. We assume an untrustworthy data user would like to learn the principal components over the distributed data. A honest-but-curious intermediary party, named proxy [19] [20], works between

978-1-5386-5790-4/18/$31.00 ©2018 IEEE

Authorized licensed use limited to: University of South Florida. Downloaded on November 13,2022 at 22:50:58 UTC from IEEE Xplore. Restrictions apply.

the data user and data owners. Data owners simultaneously encrypt their own data share and send to the proxy. The proxy runs a differentially private aggregation algorithm over the encrypted data, then sends the output to the data user. Data user computes the principal components from the output without learning the content of the underlying data. In the experiments, we study the running time of our protocol and compare with previous work [18]. We also investigate the utility and privacy trade-off in terms of the captured variance and the number of principal components. The result confirms that our protocol could maintain a high utility while preserving a strong privacy at the same time.

The contributions of our work are listed below:

- We design,analyze and implement a highly efficient and largely scalable privacy preserving distributed PCA protocol, which satisfies the (ϵ, δ)-differential privacy.
- We evaluate the proposed protocol over large dimensional and high volume real datasets with respect to the utility and privacy. In terms of SVM classification, the experiment result shows that our protocol could maintain a high utility while preserving privacy.
- We compare the proposed protocol to previous work regarding the efficiency and utility, and it shows that our protocol outperforms previous work on the running time, which achieves higher utility as well.

The rest of paper is organized as follow. The related work is presented in Section II. The preliminary background is in Section III. We describe the protocol design is in Section IV. The experiment and evaluation is in Section V. The conclusion is in Section VI.

II. Related Work

Differentially private PCA has been investigated in [21] [22] [23] [24] [25]. Blum et al. [21] present a Sub-Linear Query(SuLQ) input perturbation framework. The author defines a private database access mechanism and proves that a small amount of noise is needed, with an assumption that the number of queries is sub-linear to the number of database entries. To get the private principal components, the incidence matrix associated with the data is firstly computed, then a noise matrix is generated and added into the matrix, finally the Singular Value Decomposition is applied on the noisy matrix. Chaudhuri et al. [22] propose an iterative differential privacy mechanism using the exponential method. Given a square matrix A, a function is defined as $H() = v^T Av$, in which v is the first eigenvector of A, and H is served as the score function in the exponential mechanism. The author provides utility proof for the top k eigenvectors. Dwork et al. proposed a differentially private mechanism [24] to approximate the covariance matrix of the data by adding the symmetric noise matrix sampled from Gaussian distribution, which could be furthered used to approximate PCA, and it is proved that the noise with standard deviation $O(1)$ is sufficient to satisfy a (ϵ, δ)-diffierential privacy when the l_2 norm of each row is bounded by one. Sheffet [25] proposed another (ϵ, δ)-differentially private mechanism for the approximation of covariance matrix, and it is proved to output a positive-definite matrix. Hardt and Roth [23] propose a private power iteration method to compute the eigenvector of data matrices. It outputs the vector after a fixed number of iterations, since the vector cannot converge with arbitrary accuracy due to the Gaussian noise added in each round. The author only proves a utility guarantee for the first eigenvector, and there is no direct guarantee for the rest of them. All these work consider the data has already been collected, which is different from our scenario that the data are horizontally partitioned, where the new scenario creates challenge for PCA computation without disclosing the distributed data.

Pathak and Raj [26] propose a method to compute the eigenvector over the horizontally partitioned data using secure multi-party computation scheme. An arbitrator wants to learn the eigenvector of the distributed data held by multiple data owners. In their solution, each data owner computes and encrypts a share of his own data, then sends to the arbitrator, the arbitrator aggregates the encrypted data shares and sends the result back to each data owner. Data owners and arbitrator repeat the process till the vector converged. At the end, the arbitrator learns nothing about the data except the eigenvector. However, only one eigenvector is computed, and the author does not address the computation for other eigenvectors. In contrast, our protocol could compute a full set of principal components. Qu et al. [27] propose a distributed PCA protocol, they consider the scenario that the data are horizontally partitioned and placed in multiple places, such as data clusters. Each data cluster computes a local PCA based on its own share, then transmits the local principal components and descriptive statistics to the centralized cluster. After collecting local statistics from all data clusters, the global PCA is computed in the centralized data cluster. However, this work does not preserve the data privacy, in contrast, our proposed protocol could maintain the high utility while preserves the privacy.

III. Preliminary

A. Principal Component Analysis

Given a square matrix A, an eigenvector $\boldsymbol{v}$ of A is a non-zero vector that does not change direction when A is applied to it, such that:

$$A\boldsymbol{v} = \lambda \boldsymbol{v} \tag{1}$$

where λ is a real number scalar, referred as the eigenvalue. Suppose $A \in \mathbb{R}^{n \times n}$, then it has at most n eigenvectors, each eigenvector associates with a distinct eigenvalue.

Consider a data set with N samples $\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_N}$, where each sample has M features ($\boldsymbol{x_i} \in \mathbb{R}^M$). A center-adjusted scatter matrix $\overline{S} \in \mathbb{R}^{M \times M}$ is computed as below:

$$\overline{S} = \sum_{i=1}^{N} (\boldsymbol{x_i} - \boldsymbol{\mu})(\boldsymbol{x_i} - \boldsymbol{\mu})^T = U \Lambda U^T \tag{2}$$

where $\boldsymbol{\mu}$ is the mean vector, $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x_i}$. By using Eigenvalue Decomposition (EVD) on $\overline{S}$, we have Λ and

Authorized licensed use limited to: University of South Florida. Downloaded on November 13,2022 at 22:50:58 UTC from IEEE Xplore. Restrictions apply.

U, where $\Lambda = diag(\lambda_1, \lambda_2, .., \lambda_M)$ is a diagonal matrix of eigenvalues, and it could be arranged to a non-increasing order in absolute value, i.e., $||\lambda_1|| \geq ||\lambda_2|| \geq .. \geq ||\lambda_M||$, $U = [\boldsymbol{u_1 u_2 ... u_M}]$ is an $M \times M$ matrix where $\boldsymbol{u_j}$ denotes the j^{th} eigenvector of $\overline{S}$. In PCA, each eigenvector represents a principal component.

B. Homomorphic Encryption

An important building block in our work is homomorphic encryption. Homomorphic encryption allows computations to be performed over the encrypted data, in which the decryption of the generated result matches the result of operations performed on the plaintext. In this paper, we choose the Paillier cryptosystem to implement our protocol. Let the function $\mathcal{E}_{pk}[\cdot]$ be the encryption scheme with public key pk, function $\mathcal{D}_{sk}[\cdot]$ be the decryption scheme with private key sk, the additive homomorphic encryption is defined as:

$$a + b = \mathcal{D}_{sk}[\mathcal{E}_{pk}[a] \otimes \mathcal{E}_{pk}[b]] \tag{3}$$

where $\otimes$ denotes the modulo multiplication operator in the encrypted domain, a and b are the plaintext messages. The multiplicative homomorphic encryption is defined as:

$$a \cdot b = \mathcal{D}_{sk}[\mathcal{E}_{pk}[a]^b] \tag{4}$$

Since the cryptosystem only accepts integers as input, real numbers should be discretized. In this paper, we adopt the following equation [28],

$$Discretize_{e,F}(x) = \left\lfloor \frac{(2^e - 1) \cdot (x - min_F)}{max_F - min_F} \right\rfloor \tag{5}$$

where e is the number of bits and min_F, max_F are the minimal and maximal value of feature F. x is the real number that to be discretized, and $Discretize_{e,F}(x)$ takes value in $[0, 2^e - 1]$.

C. Differential Privacy

Differential privacy is one of the most popular privacy protection scheme. It defines a mechanism that the computation of a dataset is robust to any change of any single sample of the dataset. Given a data matrix $A \in \mathbb{R}^{n \times d}$, $A, A' \in \mathcal{A}$ are called *neighbors* if they differs on at most one row, where A, A' have a fixed size n. The formal definition of the (ϵ, δ)-differential private mechanism over A is defined below:

Definition 1 (ϵ, δ)-differential privacy [24] [29] [30]: A randomized mechanism $\mathcal{F}$ is (ϵ, δ)-differentially private if for every two neighboring matrices $A, A' \in \mathcal{A}$ and for all events $\mathcal{O} \subseteq Range(\mathcal{F})$,.

$$Pr[\mathcal{F}(A) \in \mathcal{O}] \leq e^{\epsilon} Pr[\mathcal{F}(A') \in \mathcal{O}] + \delta \tag{6}$$

The smaller ϵ, δ are, the closer $Pr[\mathcal{F}(A) \in \mathcal{O}]$ and $Pr[\mathcal{F}(A') \in \mathcal{O}]$ are, and the stronger privacy protection gains. When $\delta = 0$, the mechanism $\mathcal{F}$ is ϵ-differentially private, which is a stronger privacy guarantee than (ϵ, δ)-differential privacy with $\delta > 0$.

Given such a matrix $A \in \mathbb{R}^{n \times d}$, Dwork et al. [24] propose the mechanism for the approximation of the covariance matrix by adding noise sampled from Gaussian distribution, which satisfies a (ϵ, δ)-diffierential privacy.

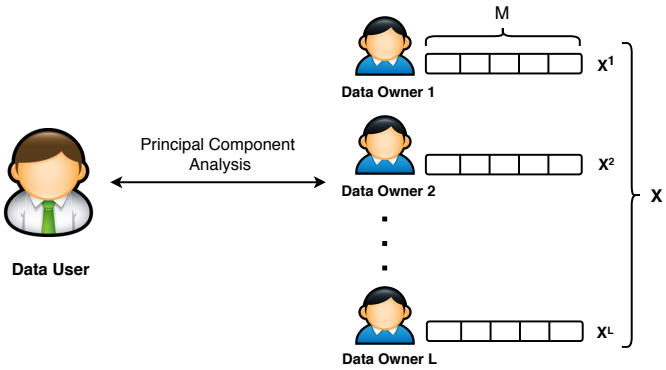


Fig. 1: Problem Description

Theorem 2 (Gaussian Mechanism) [24]. Let $f : f(A) = A^T A$, Let $\tau = \Delta f \sqrt{2ln(1.25/\delta)}/\epsilon$. The Gaussian mechanism which adds independently drawn random noise distributed as $\mathcal{N}(0, \tau^2)$ to each output of $f(A)$ ensures (ϵ, δ)-differential privacy.

where $A^T A$ is referred as the covariance matrix of A, and $||A^T A||_2$ is defined as the spectral norm of $A^T A$. $\Delta f = \max_{A, A'} ||f(A) - f(A')||_2$ is the l_2 sensitivity of f, which is the maximum over all pairs A, A' of neighboring datasets of $||f(A) - f(A')||_2$. Moreover, assuming each row of A has a unit l_2 norm, Δf is at most one. And the utility of the approximation of covariance matrix is claimed by the following theorem.

Theorem 3 (Worst case utility guarantee) [24]. Let V_k be the principal rank-k right singular subspace of A and let $\widehat{V_k}$ be the principal rank-k subspace of the approximation of the covariance matrix $\widehat{C}$, then with high probability,

$$||A\widehat{V_k}||_F^2 \geq ||AV_k||_F^2 - O(k\sqrt{d}\tau) \tag{7}$$

For the proof of *Theorem 2* and *Theorem 3*, we refer to [24].

IV. Protocol Design

A. Problem Description and Security Assumption

The problem is depicted in Fig. 1. Suppose there are L data owners, each data owner l has a data set $X^l \in \mathbb{R}^{N^l \times M}$, where M is the number of dimensions, and N^l is the number of samples held by l. The horizontal aggregation of $X^i, i = 1, 2, .., l$ generates a data matrix $X \in \mathbb{R}^{N \times M}$, where $N = \sum_1^l N^l$. There is a data user that wants to perform PCA on X. To protect the privacy of the original data, data owners would not share the data with the data user in cleartext. Moreover, any inference from the PCA should also be prevented. To resolve these privacy issues, we proposed a differentially private distributed PCA protocol, which allows the data user to learn nothing except the principal components.

In the problem, data owners are assumed to be honest and do not collude with each other. The data user is assumed to be untrustworthy, who wants to learn more information other than

Authorized licensed use limited to: University of South Florida. Downloaded on November 13,2022 at 22:50:58 UTC from IEEE Xplore. Restrictions apply.

the principal components. The proxy works as a honest-but-curious intermediary party, who does not collude with either the data user or data owners.

To learn the principal components of X, the scatter matrix of X needs to be computed. In our protocol, each data owner l computes a data share of X^l. To prevent the proxy learning the data, each data share is encrypted before sending to the proxy. Once receiving the encrypted data share from each data owner, the proxy runs the differentially private aggregation algorithm and sends the aggregated result to the data user. Then the data user constructs the scatter matrix from the result and computes the principal components.

B. Distributed Scatter Matrix Computation

We explain a solid solution block — the distributed scatter matrix computation [31] here. Suppose there are L data owners, and each data owner l has a data set, $X^l, X^l \in \mathbb{R}^{N^l \times M}$, where M is the number of dimensions, and N^l is the number of samples held by l. Each data owner locally computes a data share that contains:

$$R^l = \sum_{i=1}^{N^l} \boldsymbol{x_i} \boldsymbol{x_i}^T, \boldsymbol{v}^l = \sum_{i=1}^{N^l} \boldsymbol{x_i} \tag{8}$$

$\boldsymbol{x_i} = [x_{i1} x_{i2} ... x_{iM}]^T$. The scatter matrix $\overline{S}$ could be computed by summing the data share from each data owner:

$$\begin{aligned} \overline{S} &= \sum_{i=1}^{N} (\boldsymbol{x_i} - \boldsymbol{\mu})(\boldsymbol{x_i} - \boldsymbol{\mu})^T \\ &= \sum_{i=1}^{N} \boldsymbol{x_i} \boldsymbol{x_i^T} - N \boldsymbol{\mu} \boldsymbol{\mu}^T \\ &= \sum_{l=1}^{L} R^l - \frac{1}{N} \boldsymbol{v} \boldsymbol{v}^T \\ &= R - \frac{1}{N} \boldsymbol{v} \boldsymbol{v}^T \end{aligned} \tag{9}$$

where

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x_i}, R = \sum_{l=1}^{L} R^l, \boldsymbol{v} = \sum_{l=1}^{L} \boldsymbol{v^l}, N = \sum_{l=1}^{L} N^l \tag{10}$$

The distributed scatter matrix computation allows each data owner to compute a partial result simultaneously. Comparing with previous work [18], our method reduces the dependence between data owners and allows them to send the data share simultaneously.

C. Protocol Design

To prevent the proxy learning the data, the data share is encrypted by data owners. Then the proxy aggregates the received encrypted share from each data owner. To prevent the inference from PCA, a noise matrix is added into the aggregated result by the proxy, which makes the approximation of the scatter matrix satisfying the (ϵ, δ)-differential privacy, then the aggregated result is sent to the data user. The data user decrypts the result and constructs an approximation of the scatter matrix, then proceeds to PCA. The whole protocol is depicted in Fig. 2, and is described in detail as below:

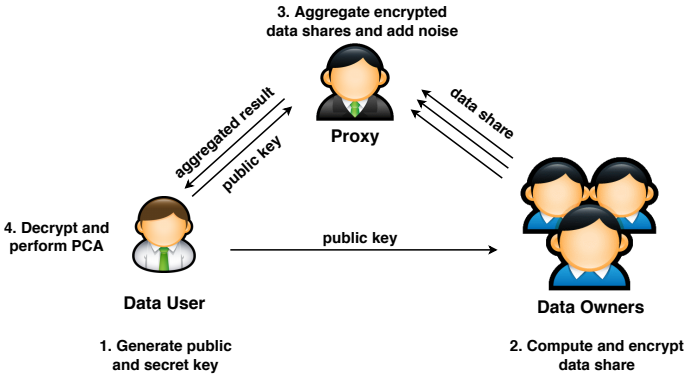


Fig. 2: Protocol Design

1) The data user generates the public key pk and sk for Paillier's cryptosystem, and sends pk to the proxy and data owners. The secure distribution of keys is well studied and out the scope of this paper, which is not discussed here.
2) Data owners compute the share $R^l, \boldsymbol{v}^l, l = 1, 2, ..., L$, and sends $\mathcal{E}_{pk}[R^l], \mathcal{E}_{pk}[\boldsymbol{v}^l], \mathcal{E}_{pk}[N^l]$ to the proxy.
3) After receiving the encrypted data share from each data owner, the proxy aggregates shares and applies the symmetric matrix noise to satisfy the differential privacy. The complete algorithm is given in Alg. 1, and the procedure is explained lines by lines below:
   a) Lines 2-4) Aggregates the data shares from each data owner.

$$\mathcal{E}_{pk}[R] = \otimes_{l=1}^{L} \mathcal{E}_{pk}[R^l] \tag{11}$$
$$\mathcal{E}_{pk}[\boldsymbol{v}] = \otimes_{l=1}^{L} \mathcal{E}_{pk}[\boldsymbol{v}^l] \tag{12}$$
$$\mathcal{E}_{pk}[N] = \otimes_{l=1}^{L} \mathcal{E}_{pk}[N^l] \tag{13}$$

   b) Lines 5-7) Constructs the noisy $\mathcal{E}_{pk}[\boldsymbol{v'}]$. To prevent the data user learning information from $\boldsymbol{v}$, the proxy generates a noisy $\mathcal{E}_{pk}[\boldsymbol{v'}]$ by summing a random vector $\mathcal{E}_{pk}[\boldsymbol{b}]$ with $\mathcal{E}_{pk}[\boldsymbol{v}], \mathcal{E}_{pk}[\boldsymbol{v'}] = \mathcal{E}_{pk}[\boldsymbol{v}] \otimes \mathcal{E}_{pk}[\boldsymbol{b}]$. It can be shown that the element v'_{ij} of $\boldsymbol{v'} \boldsymbol{v'}^T$ is:

$$\begin{aligned} v'_{ij} &= (v_i + b_i)(v_j + b_j) \\ &= v_i v_j + v_i b_j + b_i v_j + b_i b_j \end{aligned} \tag{14}$$

   Both sides of equation are divided by N yields,

$$\frac{v'_{ij}}{N} = \frac{v_i v_j}{N} + \frac{v_i b_j + b_i v_j + b_i b_j}{N} \tag{15}$$

   thus we have

$$\frac{\boldsymbol{v'} \boldsymbol{v'}^T}{N} = \frac{\boldsymbol{v} \boldsymbol{v}^T}{N} + G \tag{16}$$
$$G_{ij} = \frac{v_i b_j + b_i v_j + b_i b_j}{N} \tag{17}$$

Authorized licensed use limited to: University of South Florida. Downloaded on November 13,2022 at 22:50:58 UTC from IEEE Xplore. Restrictions apply.

Recall that in Paillier cryptosystem, the multiplicative homomorphic property is defined as:

$$\mathcal{E}_{pk}[a \cdot b] = \mathcal{E}_{pk}[a]^b \tag{18}$$

Then $\mathcal{E}_{pk}[G_{ij}]$ is:

$$\mathcal{E}_{pk}[G_{ij}] = \mathcal{E}_{pk}[v_i]^{\frac{b_j}{N}} \otimes \mathcal{E}_{pk}[v_j]^{\frac{b_i}{N}} \otimes \mathcal{E}_{pk}[\frac{b_i b_j}{N}] \tag{19}$$

It is easily to make the exponent be integer by multiplying $\boldsymbol{b}$ with N. It should be noted that during the encrytpion, the proxy has to learn N. To achieve this, the proxy sends $\mathcal{E}_{pk}[N]$ to the data user and the data user returns N in cleartext once decryption.

c) Lines 8-10) Applies symmetric matrix to satisfy the (ε, δ)-differential privacy. The proxy generates $G', G' \in \mathbb{R}^{M \times M}$, based on the differential privacy parameter (ε, δ), and gets $\mathcal{E}_{pk}[R'], \mathcal{E}_{pk}[\boldsymbol{v'}]$, where,

$$\mathcal{E}_{pk}[R'] = \mathcal{E}_{pk}[R] \otimes \mathcal{E}_{pk}[G] \otimes \mathcal{E}_{pk}[G'] \tag{20}$$

$$\mathcal{E}_{pk}[\boldsymbol{v'}] = \mathcal{E}_{pk}[\boldsymbol{v}] \otimes \mathcal{E}_{pk}[\boldsymbol{b}] \tag{21}$$

Then $\mathcal{E}_{pk}[R'], \mathcal{E}_{pk}[\boldsymbol{v'}]$ are sent to the data user.

4) After receiving the aggregated result from the proxy, $\mathcal{E}_{pk}[N], \mathcal{E}_{pk}[R'], \mathcal{E}_{pk}[\boldsymbol{v'}]$, the data user decrypts each and computes the $\overline{S'}$.

$$\overline{S'} = R' - \frac{1}{N} \boldsymbol{v'} \boldsymbol{v'}^T \tag{22}$$

With $\overline{S'}$, the data user could proceeds to compute the eigenvector and gets the principal components.

D. Security and Privacy Analysis

In our setting, the data user is assumed to be untrustworthy and the proxy is assumed to be honest-but-curious. Furthermore, we assume that the proxy is not colluded with the data user or data owners. To protect the data against the proxy, $R^l, \boldsymbol{v}^l$ and N^l are encrypted by the data owner. During the protocol execution, the proxy only learns N in plaintext and it would not disclose the privacy of a single data owner. Without colluding with the data user, the proxy cannot learn the value of R^l, $\boldsymbol{v}^l$ and N^l. On the other side, to prevent the data user gains information other than the principal components, the proxy mixes the $R, \boldsymbol{v}$ with random noise.

For the data received from the proxy, the data user decrypts $\mathcal{E}_{pk}[N], \mathcal{E}_{pk}[R'], \mathcal{E}_{pk}[\boldsymbol{v'}]$ and then proceeds to construct the approximation of the scatter matrix, $\overline{S'} = \overline{S} + G'$, in which G' is the Gaussian symmetric matrix and carried by R'. The (ϵ, δ)-differential privacy is closed for the post-processing algorithm of $\overline{S'}$, as the *Theorem 1* claims. $\boldsymbol{v'}$ is the mix of the horizontal aggregation of all data records $\boldsymbol{v}$ and a random vector $\boldsymbol{b}$. Since proxy is not colluding with the data user, the data user cannot learn the value of R and $\boldsymbol{v}$. Therefore, the data user learns nothing but the computed principal components.

As a flexible design, our protocol is capable of cooperating with different symmetric noise matrices to satisfy (ϵ, δ)-differential privacy. To demonstrate the protocol, we implement the Gaussian Mechanism as introduced in Section III.C, the algorithm is shown in Alg. 2.

Algorithm 1 DPAggregate

1: **Input**: $(\{\mathcal{E}_{pk}[R^1], \mathcal{E}_{pk}[R^2], ..., \mathcal{E}_{pk}[R^L]\}$, $\{\mathcal{E}_{pk}[\boldsymbol{v}^1], \mathcal{E}_{pk}[\boldsymbol{v}^2], ..., \mathcal{E}_{pk}[\boldsymbol{v}^L]\}$, $\{\mathcal{E}_{pk}[N^1], \mathcal{E}_{pk}[N^2], ..., \mathcal{E}_{pk}[N^L]\})$
2: $\mathcal{E}_{pk}[R] = \otimes_{l=1}^{L} \mathcal{E}_{pk}[R^l]$
3: $\mathcal{E}_{pk}[\boldsymbol{v}] = \otimes_{l=1}^{L} \mathcal{E}_{pk}[\boldsymbol{v}^l]$
4: $\mathcal{E}_{pk}[N] = \otimes_{l=1}^{L} \mathcal{E}_{pk}[N^l]$
5: $N \leftarrow \mathcal{E}_{pk}[N]$
6: Generate random vector $\boldsymbol{b} = [b_1, b_2, ..., b_M]$
7: Generate matrix $G \in \mathbb{R}^{M \times M}$, where $\mathcal{E}_{pk}[g_{ij}] = \mathcal{E}_{pk}[v_i]^{\frac{b_j}{N}} \otimes \mathcal{E}_{pk}[v_j]^{\frac{b_i}{N}} \otimes \mathcal{E}_{pk}[\frac{b_i b_j}{N}], v_i \in \boldsymbol{v}, b_j \in \boldsymbol{b}, i, j = 1, 2, ..., M$
8: $G' \leftarrow$ *Gaussian Mechanism*()
9: $\mathcal{E}_{pk}[R'] = \mathcal{E}_{pk}[R] \otimes \mathcal{E}_{pk}[G] \otimes \mathcal{E}_{pk}[G']$
10: $\mathcal{E}_{pk}[\boldsymbol{v'}] = \mathcal{E}_{pk}[\boldsymbol{v}] \otimes \mathcal{E}_{pk}[\boldsymbol{b}]$
11: **return** $\mathcal{E}_{pk}[R'], \mathcal{E}_{pk}[\boldsymbol{v'}]$

Algorithm 2 Gaussian Mechanism [24]

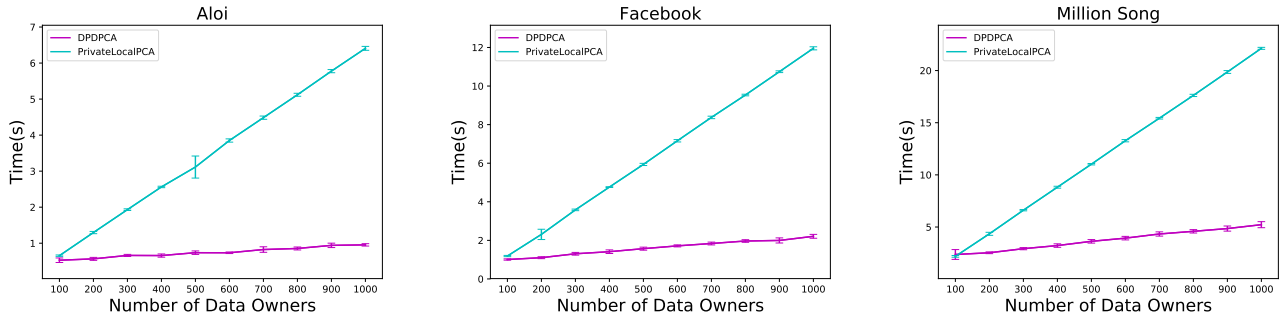
1: **Input**: $\epsilon > 0, \delta > 0$
2: Set $\tau = \sqrt{2ln(1.25/\delta)}/\epsilon$
3: Let $E \in \mathcal{R}^{n \times n}$ be the symmetric matrix with the upper triangle (including the diagonal) entry is i.i.d samples from $\mathcal{N} \sim (0, \tau^2)$, and set $E_{ji} = E_{ij}, \forall i < j$.
4: **return** E

Once the data user learns the private principal components from the protocol, he could release the principal components to public for further use [31], where the proxy is able to access the components. In that case, to the best of our knowledge, there are not enough information to recover the covariance matrix from a full set of principal components only, which implies that the proxy can't recover the approximation of covariance matrix with the released private principal components. Moreover, the data user may release a subset (top K) principal components, rather than the full set of components, which is even harder for the proxy to recover the covariance matrix. Without knowing the approximation of the covariance matrix, the proxy could not infer the plain data by removing the added noise.

V. Experiment and Evaluation

We evaluate the proposed Differentially Private Distributed PCA (DPDPCA) protocol, regarding the efficiency, utility and privacy. For the efficiency, we measure the protocol running time of DPDPCA and the previous work [18], it shows that DPDPCA outperforms the previous work. We also compare the utility of both protocols when the principal components cannot provide a good representation of the data. The experiment is developed using Python 2.7.10 and with the python Paillier homomorphic cryptosystem library published in Github [32].

Authorized licensed use limited to: University of South Florida. Downloaded on November 13,2022 at 22:50:58 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c)

Fig. 3: Running time comparison between DPDPCA and PrivateLocalPCA, $\epsilon = 0.3$

A. Dataset and Evaluation Methodology

There are 6 datasets selected for experiments, as shown in Table. I. The Aloi dataset is a collection of color images of small objects, the Facebook comment volume dataset contains features extracted from Facebook posts and the Million Song dataset consists of audio features. The cardinality of Aloi, Facebook and Million Song are more than 100,000 and the dimensionality of each is less than 100. The CNAE dataset is a text dataset, which is extracted from business documents and attributes are the term frequency. The GISETTE dataset contains grayscale images of highly confusible digits '4' and '9' and is used in NIPS 2003 feature selection challenge. The ISOLET is a dataset of spoken letters, which records the 26 English letters from 150 subjects, and it has a combination of features like spectral coefficients and contour features. All datasets excluding Aloi are from UCI machine learning repository [33], where Aloi is from LibSVM datasets repository [34]. We evaluate the performance of DPDPCA in terms of SVM classification over the CNAE, GISTTE and ISOLET dataset. For the classification result, we measure the Precision, Recall, F1 Score due to the reason that the datasets are unbalanced. All experiments have been run 10 times, the mean and standard deviation of the result are drawn in figures.

$$\text{Precision} = \frac{\text{TruePositive(TP)}}{\text{TruePositive(TP) + FalsePositive(FP)}} \tag{23}$$

$$\text{Recall} = \frac{\text{TruePositive(TP)}}{\text{TruePositive(TP) + FalseNegative(FN)}} \tag{24}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision + Recall}} \tag{25}$$

	Feature	Cardinality
Aloi	29	108000
Facebook	54	199030
Million Song	90	515345
CNAE	857	1080
ISOLET	617	7797
GISETTE	5000	13500

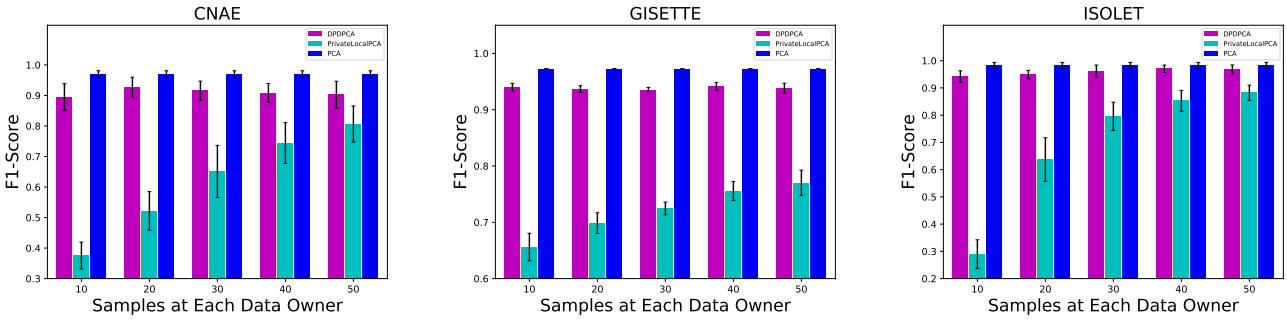
TABLE I: Experimental Datasets

B. DPDPCA vs. PrivateLocalPCA

As we mentioned in the Introduction, the previous work [18] suffers from two main issues, the excessive protocol running time and the utility degradation when the local principal components fail to provide a good data representation. In this section, we compare both protocols in these two aspects. For simplicity, we refer our protocol as "DPDPCA" and the work in [18] as "PrivateLocalPCA".

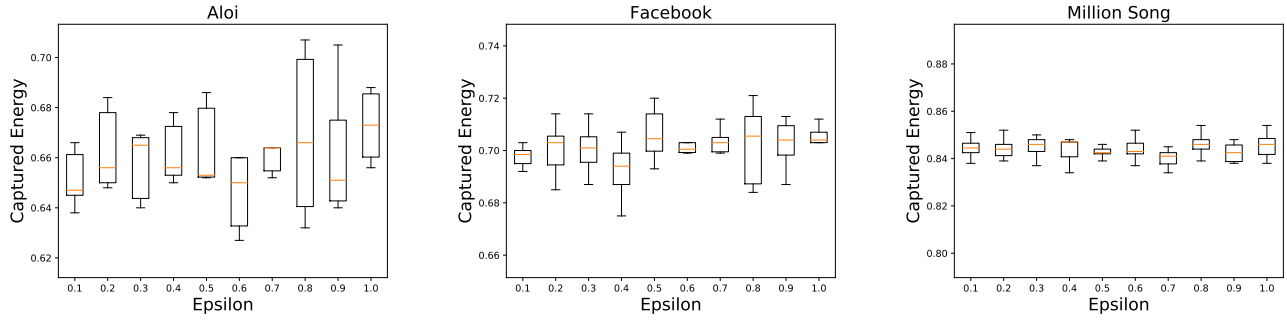
1) Efficiency: We compare the running time of DPDPCA and PrivateLocalPCA. The total running time of DPDPCA includes the average local computation time of the data owner, the time of private aggregation algorithm in the proxy and the time of performing PCA in the data user as well as the data transmission time among parties. For PrivateLocalPCA, the running time starts from the first data owner, and ends at the last data owner, including the local PCA computation and transmission time. To make the communication consistent and stable, we simulate the data transmission using the I/O operations, rather than local network. We measure the protocol running time regarding different number of data owners, and all samples distributed evenly to each data owner. The experiment ran on a desktop machine (i7-5820k, 64GB memory) and the result is shown in Fig. 3, the horizontal axis specifies the number of data owners, and the vertical axis specifies the running time in seconds. It can be seen that PrivateLocalPCA runs almost linearly upon the number of data owners, the reason is that PrivateLocalPCA required that the local principal components are transmitted through data owners one by one, the next data owner has to wait for the result from the previous one, thus it has a time complexity of $O(n)$, where n is the number of data owners. In contrast, DPDPCA costs much less time than PrivateLocalPCA given the same number of data owners, the reason is, firstly, the distributed scatter matrix computation allows each data owner to compute the local share simultaneously; secondly, the proxy could implement the aggregation of the local shares in parallel, which runs log-linearly upon the number of data owners. Overall, DPDPCA has a better scalability than PrivateLocalPCA regarding the number of data owners.

Authorized licensed use limited to: University of South Florida. Downloaded on November 13,2022 at 22:50:58 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c)

Fig. 4: Principal components utility comparison between DPDPCA and PrivateLocalPCA, $\epsilon = 0.5$.



(a) (b) (c)

Fig. 5: Captured variance, $\delta = 1/N$. (a) Aloi: $\delta = 9.26 \times 10^{-6}$. (b) Facebook: $\delta = 5.02 \times 10^{-6}$. (c) Million Song: $\delta = 1.94 \times 10^{-6}$.

2) Utility: In this subsection, we show the utility degradation of PrivateLocalPCA when the amount of data is much less than the number of feature. As stated in the Introduction, we consider the scenario that each data owner holds a dataset that the cardinality may be much smaller than the number of features, such as the images, the rating regarding music and movies, the personal activity data. To simulate this scenario, we distribute different size of samples to each data owner in the experiment. For PrivateLocalPCA, the variance is not fully preserved due to the reason that only the first few principal components are used to represent the data. In contrast, DPDPCA is not affected by the number of samples that each data owner holds, the local descriptive statistics are aggregated to build the scatter matrix, thus the total variance is not lost. In the experiment, we measure the F1 Score of the transformed data regarding different number of private principal components, in which the number of principal components is determined by the rank of the data held by each data owner. With components from each protocol, both training data and testing data are projected to a lower dimensional space. Then we use the transformed training data to train a SVM classifier with rbf kernel, and test the classifier with unseen data. To provide a ground truth, the noiseless PCA is performed over the training data as well. And the same symmetric matrix noise mechanism [18] is applied to DPDPCA to make a fair comparison. Fig. 4 shows the experiment result. The horizontal axis specifies the number of samples held by each data owner, and the vertical axis shows the F1 Score. In can be seen that the F1 Score of DPDPCA is invariant to the number of samples at each data owner, and the result is compatible to the noiseless PCA, which implies a high utility is maintained. In the mean while, the F1 Score of PrivateLocalPCA is heavily affected by the number of samples at each data owner, and it cannot maintain the utility with only few samples. Overall, for the CNAE and GISETTE dataset, the F1 Score of DPDPCA outperforms PrivateLocalPCA under all settings.

3) Captured Variance: We study the trade-off between the utility and the privacy of DPDPCA by meansuring the captured variance of the private principal components regarding the Gaussian mechanism, where the standard deviation of the additive noise is inversely proportionally to ϵ. The smaller ϵ is, the more noise adds, and the more privacy gains. The result is shown in Fig. 5, the horizontal axis specifies ϵ, and the vertical axis shows the ratio of the captured variance. From the figure, it can be seen that the captured variance by the Gaussian mechanism almost maintained the same level for the given ϵ range, moreover, the value of ratio implies that the Gaussian mechanism captures a large proportion of the variance.

Authorized licensed use limited to: University of South Florida. Downloaded on November 13,2022 at 22:50:58 UTC from IEEE Xplore. Restrictions apply.

VI. Conclusion

In this paper, we present a highly efficient and largely scalable (ϵ, δ)-differentially private distributed PCA protocol, DPDPCA. We consider the scenario that the data are horizontally partitioned, and there is an untrustworthy data user wants to learn the principal components over the distributed data in a short time, such as in disaster management and emergency response. Comparing to previous work, DPDPCA offers a higher efficiency and better utility. Additionally, it is also able to incorporate with different symmetric matrix schemes to achieve the (ε, δ)-differential privacy. In the experiment, we evaluate DPDPCA over large dimensional and high volume real datasets in terms of the efficiency, utility and privacy, the results show that DPDPCA could maintain a high utility while preserving the privacy.

Acknowledgment

This material is based on research sponsored by the DARPA Brandeis Program under agreement number N66001-15-C4068.[1]

References

[1] A. Wiesel and A. O. Hero, "Decomposable principal component analysis," *IEEE Transactions on Signal Processing*, 2009.

[2] W. Dong, L. Luo, C. Chen, J. Bu, X. Liu, and Y. Liu, "Post-deployment anomaly detection and diagnosis in networked embedded systems by program profiling and symptom mining," *IEEE Transactions on Parallel and Distributed Systems*, 2016.

[3] D. Kim and B.-J. Yum, "Collaborative filtering based on iterative principal component analysis," *Expert Systems with Applications*, 2005.

[4] P. Howland, M. Jeon, and H. Park, "Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, 2003.

[5] X. Chen and N. A. Schmid, "Empirical capacity of a recognition channel for single- and multipose object recognition under the constraint of pca encoding," *IEEE Transactions on Image Processing*, 2009.

[6] H. Kargupta, W. Huang, K. Sivakumar, and E. Johnson, "Distributed clustering using collective principal component analysis," *Knowledge and Information Systems*, 2001.

[7] L. Huang, X. Nguyen, M. Garofalakis, M. I. Jordan, A. Joseph, and N. Taft, "In-network pca and anomaly detection," in *Conference on Neural Information Processing Systems*, 2006.

[8] Y. Liu, L. Zhang, and Y. Guan, "Sketch-based streaming pca algorithm for network-wide traffic anomaly detection," in *IEEE Thirtieth International Conference on Distributed Computing Systems*, 2010.

[9] M. A. Livani and M. Abadi, "Distributed pca-based anomaly detection in wireless sensor networks," in *International Conference for Internet Technology and Secured Transactions*, 2010.

[10] V. H. Tuulos and H. Tirri, "Combining topic models and social networks for chat data mining," in *IEEE/WIC/ACM international Conference on Web intelligence*, 2004.

[11] M. Lepinski, D. Levin, D. McCarthy, R. Watro, M. Lack, D. Hallenbeck, and D. Slater, "Privacy-enhanced android for smart cities applications," in *EAI International Conference on Smart Urban Mobility Services*, 2015.

[12] X. Shen, B. Tan, and C. Zhai, "Privacy protection in personalized search," in *ACM Special Interest Group in Information Retrieval Forum*, 2007.

[13] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *IEEE Symposium on Security and Privacy*, 2008.

[1]The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

[14] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*, 2008.

[15] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and D. Megías, "Individual differential privacy: A utility-preserving formulation of differential privacy guarantees," *IEEE Transactions on Information Forensics and Security*, 2017.

[16] T. Zhang and Q. Zhu, "Dynamic differential privacy for admm-based distributed classification learning," *IEEE Transactions on Information Forensics and Security*, 2017.

[17] T. Zhu, P. Xiong, G. Li, and W. Zhou, "Correlated differential privacy: hiding information in non-iid data set," *IEEE Transactions on Information Forensics and Security*, 2015.

[18] H. Imtiaz, R. Silva, B. Baker, S. M. Plis, A. D. Sarwate, and V. Calhoun, "Privacy-preserving source separation for distributed data using independent component analysis," in *Annual Conference on Information Science and Systems*, 2016.

[19] R. Chen, A. Reznichenko, P. Francis, and J. Gehrke, "Towards statistical queries over distributed private user data." in *NSDI*, 2012.

[20] R. Chen, I. E. Akkus, and P. Francis, "Splitx: high-performance private analytics," in *ACM SIGCOMM Computer Communication Review*, 2013.

[21] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: the sulq framework," in *Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2005.

[22] K. Chaudhuri, A. Sarwate, and K. Sinha, "Near-optimal differentially private principal components," in *Advances in Neural Information Processing Systems*, 2012.

[23] M. Hardt and A. Roth, "Beyond worst-case analysis in private singular vector computation," in *Forty-Fifth Annual ACM Symposium on Theory of Computing*, 2013.

[24] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang, "Analyze gauss: optimal bounds for privacy-preserving principal component analysis," in *Forty-sixth Annual ACM Symposium on Theory of Computing*, 2014.

[25] O. Sheffet, "Private approximations of the 2nd-moment matrix using existing techniques in linear regression," *arXiv preprint arXiv:1507.00056*, 2015.

[26] M. Pathak and B. Raj, "Privacy preserving protocols for eigenvector computation," in *International Workshop on Privacy and Security Issues in Data Mining and Machine Learning*, 2010.

[27] Y. Qu, G. Ostrouchov, N. Samatova, and A. Geist, "Principal component analysis for dimension reduction in massive distributed data sets," in *IEEE International Conference on Data Mining*, 2002.

[28] S. Govindarajan, P. Gasti, and K. S. Balagani, "Secure privacy-preserving protocols for outsourcing continuous authentication of smartphone users with touch data," in *IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems*, 2013.

[29] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference*, 2006.

[30] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, 2011.

[31] M. Al-Rubaie, P. Wu, J. Chang, and S. Kung, "Privacy-preserving pca on horizontally-partitioned data," *IEEE Conference on Dependable and Secure Computing*, 2017.

[32] "Pure python paillier homomorphic cryptosystem," https://github.com/mikeivanov/paillier.

[33] "Uci machine learning repository," http://archive.ics.uci.edu/ml/.

[34] "Libsvm," https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

Authorized licensed use limited to: University of South Florida. Downloaded on November 13,2022 at 22:50:58 UTC from IEEE Xplore. Restrictions apply.