Research Article

A Blockchain-Integrated Divided-Block Sparse Matrix Transformation Differential Privacy Data Publishing Model

Yiyang Hong ^{1,2} Xingwen Zhao ^{1,2} Hui Zhu ^{1,2} and Hui Li ^{1,2}

¹State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710126, China

²School of Cyber Engineering, Xidian University, Xi'an 710126, China

Correspondence should be addressed to Xingwen Zhao; sevenzhao@hotmail.com

Received 18 June 2021; Accepted 16 October 2021; Published 7 December 2021

Academic Editor: Shahram Babaie

Copyright © 2021 Yiyang Hong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of information technology, people benefit more and more from big data. At the same time, it becomes a great concern that how to obtain optimal outputs from big data publishing and sharing management while protecting privacy. Many researchers seek to realize differential privacy protection in massive high-dimensional datasets using the method of principal component analysis. However, these algorithms are inefficient in processing and do not take into account the different privacy protection needs of each attribute in high-dimensional datasets. To address the above problem, we design a Divided-block Sparse Matrix Transformation Differential Privacy Data Publishing Algorithm (DSMT-DP). In this algorithm, different levels of privacy budget parameters are assigned to different attributes according to the required privacy protection level of each attribute, taking into account the privacy protection needs of different levels of attributes. Meanwhile, the use of the divided-block scheme and the sparse matrix transformation scheme can improve the computational efficiency of the principal component analysis method for handling large amounts of high-dimensional sensitive data, and we demonstrate that the proposed algorithm satisfies differential privacy. Our experimental results show that the mean square error of the proposed algorithm is smaller than the traditional differential privacy algorithm with the same privacy parameters, and the computational efficiency can be improved. Further, we combine this algorithm with blockchain and propose an Efficient Privacy Data Publishing and Sharing Model based on the blockchain. Publishing and sharing private data on this model not only resist strong background knowledge attacks from adversaries outside the system but also prevent stealing and tampering of data by not-completely-honest participants inside the system.

1. Introduction

With the arrival of the era of big data and cloud computing, the data center of each city is full of all kinds of high-dimensional data. Many organizations related to medical, finance, public security, and other fields often need to outsource their data to third parties for analysis and to help them make decisions. However, the published data may contain extremely sensitive data, which can be collected and sold by third parties. The leakage of such sensitive information may lead to unpredictable consequences such as kidnapping or blackmail and so on.

In order to avoid the leakage of sensitive data, it is wise to take some measures to protect sensitive data. Traditionally, the sensitive data can be protected by the classical Laplace

algorithm to add noise perturbation directly to the data, which makes it difficult to know the exact single data records while ensuring the relative approximation of statistical results. However, data from medical care, finance, and public security are often of high dimensionality and have a large number of records. If the traditional method such as direct Laplace noise is adopted, it will add too much noise to the massive data, which will lead to a great distortion of the data. And then, analysis of these published data may lead to wrong or deviation results. In some cases, the dataset may contain the attributes of different privacy levels. Some attributes need to be kept secret strictly, so they should be confused carefully even at the expense of certain usability, while some attributes do not need strict confidentiality, so they can be handled based on the principle of ensuring usability and

reducing the deviation rates as much as possible. Therefore, it is of practical importance to develop different privacy protection schemes according to the privacy protection needs of different attributes, respectively.

Blockchain is a decentralized, collectively maintained, secure, and trustworthy structure [1], which is ideal for storing and protecting private data to avoid large-scale data loss or leakage caused by attacks on centralized institutions. Data can be packaged and written to the blockchain through hash operations, and security is ensured by consensus algorithms and asymmetric cryptography of nodes within the blockchain P2P network system. Therefore, it is important to incorporate the advantages of blockchain into the data publish safety field.

For dimensionality reduction methods, Principal Component Analysis (PCA) [2] is one of the most commonly used linear dimensionality reduction methods; its main goal is to find an optimal set of unit orthogonal vectors based on a linear transformation and reconstruct the original samples with their linear combinations to minimize the error between the reconstructed samples and the original samples. In many modern information systems, the amount of data is very large. The huge amount of data increases the difficulty of data analysis and processing, while PCA is able to simplify the dataset and make the data more likely to use, especially reducing the computational complexity of the algorithm. For example, the efficiency of face recognition is dramatically increased when projecting data to lower dimensions.

Compared to K-anonymity privacy-preserving algorithms, differential privacy is able to resist strong background knowledge attacks. Differential privacy can secure individual privacy by adding random noise to the data, making it impossible for an attacker to identify whether a certain record is in the dataset or not. Therefore, it can hide individual private information while keeping almost unchanged the basic statistical information of the entire dataset [3]. The commonly differential privacy Laplacian mechanism, on the other hand, can add noise based on the sensitivity (the maximum change that can be induced by changing one record in the dataset) of the dataset. Thus, a principal component analysis algorithm incorporating the differential privacy perturbations can transform the original variables into low-dimensional variables that reflect the majority of information about the original variables, and adding even a small perturbation to the data matrix of a low-dimensional variable can trigger a large change in its overall variables [4].

Specifically, the contribution of this paper consists of the following four main aspects:

- (1) To address the problem that existing principal component analysis differential privacy algorithms do not take into account the different privacy protection needs of each attribute in high-dimensional datasets. We design a Divided-block Sparse Matrix Transformation Differential Privacy Data Publishing Algorithm (DSMT-DP) with considering the different privacy protection requirements of each

attribute and add privacy noise perturbation to make the algorithm satisfy ϵ -differential privacy.

- (2) To address the problem that existing principal component analysis differential privacy algorithms are inefficient in processing high-dimensional datasets. We use a divided-block scheme and sparse matrix transformation scheme to significantly improve the computational efficiency of the principal component analysis differential privacy data publishing algorithm than the original algorithms.
- (3) We theoretically prove that the DSMT-DP satisfies ϵ -differential privacy. And for the DSMT-DP algorithm, we conducted experiments on real datasets to verify the performance of the DSMT-DP algorithm in terms of both mean square error and privacy processing operations. The experimental results show that, compared with the traditional principal component analysis differential privacy algorithm, it significantly improves the efficiency of computing and processing for dimensionality reduction, and the mean square error of this proposed algorithm is smaller than that of the traditional principal component analysis differential privacy algorithm.
- (4) Further, we combine this algorithm with blockchain and propose an Efficient Privacy Data Publishing and Sharing Model based on the blockchain. Publishing and sharing data on this model not only resist strong background knowledge attacks from adversaries outside the system but also prevent stealing and tampering of data by not-completely-honest participants inside the system.

2. Related Work

Traditional privacy-preserving algorithms for principal component analysis can be divided into several different types, and in terms of the privacy-preserving noise addition stage, there are two dominant types.

The first type is to add privacy-preserving perturbation noise after computing the approximate space. This type of method is called output perturbation, and the commonality of this type of algorithm is to add privacy-preserving perturbation to the low-dimensional approximate result so that the output is approximate to the original yet retains statistical information. For example, in 2013, Jiang et al. [5] handled the principal component analysis and linear discriminant analysis algorithms through a differential privacy mechanism, which adds noise to both the covariance matrix and the projection matrix. This algorithm is able to reduce the amount of noise and the distortion of the data with the same privacy-preserving parameter compared to the traditional scheme of direct perturbation by the Laplace mechanism. In 2020, Peng et al. [6] proposed the algorithm of MIC-PCA-DP for the deficiency that the traditional Pearson correlation coefficient can only capture the linear relationship of the data in the process of dimensionality reduction, which is improved by calculating the maximum information coefficient to perform dimensionality reduction

and then adding differential privacy perturbation noise to the low-dimensional approximation matrix. The algorithm can be applied not only to linear relations but also adapt to the dimensionality reduction process of data relations such as nonlinear relations and nonfunctional dependencies and retain the correlation linkage of the original information to the maximum extent. Similar related literatures of output perturbation are available in [7–10].

Another way is to add privacy-preserving perturbation noise to the covariance matrix, and this type of method is called input perturbation. The commonality of input perturbation algorithms is the addition of privacy-preserving perturbation noise to the covariance matrix before computing the eigenvector space. Since differential privacy satisfies the postprocessing invariance [11, 12], in other words, the output that satisfies differential privacy in the previous process will automatically retain the property of satisfying differential privacy as the input in the next stage, thus, afterward, any further singular value decomposition operations on the noise-added covariance matrix still maintain the satisfaction of differential privacy. And the output perturbation has better performance because it is not limited by the influence of the computed eigenvector space. In 2014, Dwork et al. [11] proposed a near-optimal low-dimensional approximation matrix generation algorithm that satisfies approximate (ϵ, δ) -differential privacy by adding Gaussian-distributed perturbation noise to the covariance matrix and further derived optimal bounds on the errors. And in 2016, Jiang et al. [13] added the matrix noise generated by the Laplace and Wishart distributions to the covariance matrix, respectively, and proposed a better principal component analysis based differential privacy protection mechanism that not only ensures the positive semidefinite of the noise-added covariance matrix but also achieves pure $(\epsilon, 0)$ -differential privacy protection. Similar related literature of input perturbation is available in [14].

Meanwhile, this paper conducts a study on the efficiency of dimensionality reduction of high-dimensional matrices. Traditional dimensionality reduction methods usually use QR decomposition Householder transform, which is less efficient in computing the eigenvalues and eigenvectors of matrices; in particular, when high-dimensional operations are processed, the covariance matrix will be very large and the speed of dimensionality reduction will be very slow. Therefore, simply using traditional QR decomposition Householder transform to reduce dimensionality in high-dimensional datasets is sometimes not always the optimal choice.

At the same time, research on the combination of blockchain and differential privacy has been increasing in recent years. In 2018, Alnemari et al. [15] proposed a novel approach that integrates existing access control mechanisms with blockchain and differential privacy to protect infrastructure data. And in 2020, Liu et al. [16] proposed a blockchain-based secure federal learning framework to create smart contracts and prevent malicious or unreliable participants from being involved in federal learning. And the central aggregator recognizes malicious and unreliable participants by automatically executing smart contracts to defend against poisoning attacks. Further, they use local

differential privacy techniques to prevent membership inference attacks. Then in 2021, in order to solve the problem of query restriction in supply chain financial blockchain system, Jiang et al. [17] proposed a privacy budget management and noise reusing method in a multichain blockchain environment based on Hyperledger multichannel technology and community clustering algorithm. A historical record book is established to manage the privacy budget according to historical query types, and a differential privacy protection algorithm based on noise reusing is used to generate and reuse noise.

3. Preliminaries

3.1. Differential Privacy. Differential privacy [18] is built on the concept of adjacent data sets. If two matrices D and D' are given, and the elements are the same between them except for only one element that is different, then the matrices D and D' are regarded as a pair of adjacent data matrices. The set of query functions $F = \{f_1, f_2, \dots, f_m\}$ is a set of query functions on matrix D . After processing the set of query functions F using a randomized algorithm K , modifying an element in the matrix D will hardly affect the query effect of the functions, and thus, the attacker can hardly distinguish this pair of adjacent matrices. The specific definition of differential privacy is as follows.

Definition 1 (differential privacy). In a randomized algorithm K , P_Z is the set of all output results of K when the input set is Z . For any two adjacent datasets $T \in Z$ and $T' \in Z$ or any subset S_Z of P_Z , if the algorithm K satisfies

$$\Pr[K(T) \in S_Z] \leq e^\epsilon \times \Pr[K(T') \in S_Z], \quad (1)$$

then, the algorithm K satisfies ϵ -differential privacy, where the parameter ϵ is called the privacy-preserving budget parameter. The value of the privacy-preserving budget parameter ϵ has an important impact on the ratio of the output results of the algorithm K in the adjacent data sets. When the value of ϵ is controlled to be small, it means that the modification of a certain element almost negligibly and slightly affects the change in the output result. Therefore, in order to achieve better data privacy protection, it needs the higher level of privacy-protected data for the sensitive attributes, the value of ϵ is often controlled to be small, so that attackers cannot easily obtain sensitive attribute values. However, in practice, the value of ϵ is not as small as possible, which is due to the fact that the scheme of differential privacy introduces additional noise, which leads to a decrease in the availability of data. Therefore, in practical application scenarios, the value of the parameter ϵ for privacy protection should be chosen reasonably so that differential privacy can maximize the safety protection of sensitive attributes while safeguarding the availability of data.

The main research in the differential privacy noninteractive publishing framework is how to design efficient publishing algorithms, which are mainly processed by first compressing or transforming the original data and then

adding perturbation noise to the transformed data. The main design goals of such methods are how to reduce publishing errors and how to improve data availability, etc.

3.2. Principal Component Analysis. Principal component analysis (PCA) [2] is a method that transforms the original matrix into linearly independent orthogonal components through linear transformations in the field of Linear Algebra, which is of great value in data dimensionality reduction. PCA first performs decomposition of the data covariance matrix to obtain the eigenvalues and corresponding eigenvectors. Then, the matrix is projected and mapped to the space where the eigenvectors are located, forming a low-dimensional mapping matrix that contains most of the information in the original data.

This is achieved by setting the matrix as $X_{m \times p}$, m as the number of samples, and p as the sample dimension, calculating the covariance matrix A , and then calculating the eigenvalues $\lambda_i (1 \leq i \leq p)$ and their corresponding eigenvectors $e_i (1 \leq i \leq p)$. Next, the eigenvectors are arranged in descending order according to their corresponding eigenvalues. The first k eigenvectors are taken to form a matrix of mutually orthogonal eigenvectors matrix E_k , and the original matrix can be projected into the eigenvector space to obtain the k -dimensional mapping matrix $F_{m \times k}$. After that, the low-rank approximation matrix of the original matrix can be obtained [19]:

$$F_{m \times k} = X_{m \times p} E_k^T. \quad (2)$$

3.3. Sparse Matrix Transformation. In the dimensionality reduction process of principal component analysis, accurate estimation of the mutually orthogonal eigenvector matrices is required, and this step is particularly critical in high-dimensional data matrices. Assume that the high-dimensional matrix is represented by X , where X has m vectors, each of which is of dimension p . The covariance matrix of the matrix X is A , and an unbiased estimate of A is S . To improve the accuracy of the maximum likelihood estimation of the orthogonal eigenvector matrix E after the covariance matrix eigendecomposition, Cao and Bouman [20] of Purdue University, USA, proposed a transformation [20, 21] that represents the orthogonal transformation matrix as the product of a series of Givens rotations $E = G_1 G_2 \dots G_H$, which can improve the accuracy of the maximum likelihood estimation of the orthogonal eigenvector matrix to a greater extent.

Suppose the eigendecomposition of the covariance matrix A is $A = E \Lambda E^T$, where Λ denotes a diagonal matrix composed of eigenvalue elements and E is represented as an orthogonal eigenvector matrix. If we assume that the columns of X can be represented as Y , which are independent and identically distributed Gaussian random vectors with mean zero, then the likelihood estimation of X is

$$P_A(X) = \frac{1}{(2\pi)^{(mp/2)} |A|^{-(m/2)}} \exp \left\{ -\frac{1}{2} \text{trace} \{ X^T A^{-1} X \} \right\}. \quad (3)$$

After taking the logarithm,

$$\begin{aligned} \log P_{(E, \Lambda)}(X) = & -\frac{m}{2} \text{trace} \{ \text{diag}(E^T S E) \Lambda^{-1} \} \\ & -\frac{m}{2} \log |\Lambda| - \frac{mp}{2} \log(2\pi). \end{aligned} \quad (4)$$

The results regarding E and Λ can be derived from the joint maximum likelihood estimation:

$$\begin{aligned} \hat{E} = & \arg \min_{E \in \Omega} \left\{ \left| \text{diag}(E^T S E) \right| \right\}, \\ \hat{\Lambda} = & \text{diag}(\hat{E}^T S \hat{E}), \end{aligned} \quad (5)$$

where Ω is the set of orthogonal transformations and these orthogonal transformations can be expressed as the product of a series of Givens rotations. Express E in terms of sparse matrix transformations of H order mutually orthogonal vectors, i.e., the product of a series of Givens rotations:

$$E = \prod_{h=1}^{h=H} G_h = G_1 G_2 \dots G_H. \quad (6)$$

Each sparse matrix G_h in equation (6) is a Givens rotation of a pair of coordinates (i_h, j_h) , which is around two axes i_h and j_h at a rotation angle θ_h , whose expression is given by

$$G_h = I + \Theta(i_h, j_h, \theta_h), \quad (7)$$

where I is the unit matrix and the specific expansion of $\Theta(i_h, j_h, \theta_h)$ is as follows:

$$\Theta(i_h, j_h, \theta_h) = \begin{cases} \cos(\theta) - 1, & r = s = i \text{ or } r = s = j, \\ \sin(\theta), & r = i, s = j, \\ -\sin(\theta), & r = j, s = i, \\ 0, & \text{else.} \end{cases} \quad (8)$$

When changing incrementally from $h = 1$ to $h = H$, constant iterations are required to satisfy the following conditions:

$$\begin{aligned} G_h^* = & \arg \min_{G_h} \left| \text{diag}(G_h^T S G_h) \right|, \\ S_{h+1} = & G_h^{*T} S_h G_h^*. \end{aligned} \quad (9)$$

The goal of the sparse matrix transformation is to obtain an estimate of the eigenvector matrix E using a finite number of Givens rotations. And before each iteration, the coordinates (i_h, j_h) and angles θ_h need to be determined.

$$\begin{aligned} (i_h, j_h) \leftarrow & \arg \min_{(i,j)} \left(1 - \frac{[S_h]_{ij}^2}{[S_h]_{ii} [S_h]_{jj}} \right), \\ \theta_h = & \frac{1}{2} a \tan(-2 [S_h]_{i_h j_h}, [S_h]_{i_h i_h} - [S_h]_{j_h j_h}). \end{aligned} \quad (10)$$

In this way, G^* of each step is obtained, which gives the maximum likelihood estimate of the orthogonal transformation matrix E :

$$\hat{E} = \prod_{h=1}^{h=H} G_h^*. \quad (11)$$

4. DSMT-DP Data Publishing Algorithm and the Efficient Privacy Data Publishing and Sharing Model Based on the Blockchain

4.1. Introduction of DSMT-DP Data Publishing Algorithm. The Divided-block Sparse Matrix Transformation Differential Privacy Data Publishing Algorithm involves the data publisher and the data requester. First, the data publisher processes the original data using the DSMT-DP data publishing algorithm to form the intermediate data; after that, the intermediate data is transmitted through a secure channel and recorded to the blockchain. Then, the data requester initiates a request to obtain the key from the publisher, and finally, the corresponding intermediate data can be recovered for data analysis and mining.

The specific process of the proposed algorithm containing nine stages is shown in Figure 1.

For the data publisher, the process of the proposed algorithm contains the first seven stages: data collection stage, data preprocessing stage, data privacy protection level assessment stage, data perturbation stage, data transformation stage, encrypted transmission stage, and intermediate data publishing stage. After that, the intermediate data would be published and uploaded to the blockchain platform.

For the data requester, after the data requester initiates a request to obtain the key from the publisher, the corresponding intermediate data can be recovered in the eighth stage, the data recovery stage. Finally, in the data stitching stage, the data requester forms the privacy protection data matrix with the same scale as the original that can be analyzed and mined.

4.2. DSMT-DP Algorithm Specific Process. Abbreviations and symbols used in the algorithm are explained in Table 1.

Input: accept a total of m data samples of the input original data, each sample has n dimensions, and the total privacy protection parameter to be set is ϵ .

Output: a privacy-protected version of m data samples, each sample has n dimensions, and the privacy-protected version of the perturbed data is used as the final data for publishing. The following are the specific nine stages.

4.2.1. Data Collection Stage. Data collection stage contains two substages: the data collection stage and the parameter selection stage.

Data collection stage: data from each individual sample is collected, the specific values in the n attributes in each sample are determined, and the values m of the

individual sample dimensions in the dataset are updated in real time.

Parameter selection stage: ϵ indicates the privacy budget parameter. The higher the level of privacy, the smaller the budget parameter ϵ and the worse the usability of the data. Therefore, the selection of budget parameter value needs to be adjusted according to the actual privacy protection level demand, and finally, the appropriate total privacy budget parameter ϵ is determined. The cumulative contribution percentage of eigenvalues in the dimensionality reduction stage is also determined to w in this stage.

4.2.2. Data Preprocessing Stage. Data preprocessing stage: scan each sample separately, and if there exists a certain attribute with a null value of the sample, it is filled with value zero to ensure that there is a value for each attribute in n dimensions. All data are arranged into a matrix with m rows and n columns as a whole, the number of samples in this matrix is m , and the dimension of each sample is n .

4.2.3. Data Privacy Protection Level Assessment Stage. The data privacy protection level assessment stage includes four substages: attribute privacy level assessment stage, attribute rearrangement stage, dataset division stage, and divided-block privacy level labeling stage.

Attribute privacy level assessment stage: the sensitivity level of each column of attributes is assessed, and each column is labeled according to the relatively high, medium, and low levels.

Attribute rearrangement stage: after labeling the privacy levels, the dataset is rearranged according to the privacy level of each attribute, and the higher the privacy level of the attribute is, the lower (more forward) the dimension it is arranged. After rearranging all attributes by privacy level, a new matrix with m rows and n columns is reformed.

Data set division stage: since the dimension number of the matrix is usually large, it will affect the efficiency of the covariance matrix calculation in the subsequent data processing stage. Therefore, it is very important to divide the n -dimensional matrix into several low-dimensional matrices to improve the efficiency of covariance processing in the subsequent stage. The number of dimensions of each divided-block matrix within ten is more appropriate. Therefore, a new matrix of m rows and n columns is formed by arranging all the attributes according to the privacy level, and then, dividing is performed according to the dimensionality threshold of p ($p \leq 10$). The last divided-block matrix with a dimension less than p is filled with the value zero to form a p -dimensional filled matrix.

Divided-block matrix sensitivity level labeling stage: for the divided-block matrix with m rows and p columns, the level labeling will be performed. Since the privacy level of each attribute has been labeled according to

FIGURE 1: Process of DSMT-DP algorithm.

TABLE 1: Explanation of abbreviations and symbols used in the algorithm.

Abbreviations or symbols	Full name or explanations
DSMT-DP	Divided-block Sparse Matrix Transformation Differential Privacy Data Publishing Algorithm
m	The number of data samples (row of the matrix)
n	The number of data attributions (column of matrix)
ϵ	The budget parameter of differential privacy
w	The cumulative contribution percentage of eigenvalues
p	The divided-block dimensionality threshold
q	The number of divided-block
X	The data matrix
A	The covariance matrix
C	The positive semidefinite matrix
W	The noisy sample matrix
A'	The noise-added covariance matrix
S	The unbiased estimate of A'
H	The total number of iterations
h	The current number of iterations
(i, j)	The pair of coordinates
θ	The Givens rotation angle
G	The sparse matrix
I	The unit matrix
k	The selected number of eigenvectors
λ	The eigenvalue
Λ	The eigenvalue sequence
E	The eigenvector matrix
F	The low-dimensional projection matrix
Y	The recovery matrix

high, medium, and low, and the attributes also have been sorted, therefore, the privacy level of the divided-block matrix in this stage can also be more easily labeled to high, medium, and low levels; i.e., the privacy protection level of the overall divided-block matrix is classified into high, medium, and low levels.

So far, a complete matrix containing m -row and n -column is divided to form some m -row and p -column

divided-block matrices, the number of the divided-block is q , and the divided-block matrix is classified into high, medium, and low levels according to the needs of different privacy protection levels.

4.2.4. Data Perturbation Stage. The data perturbation stage contains five substages: data normalization stage, calculation of covariance matrix stage, privacy budget parameter

assignment stage, privacy noise extraction stage, and data privacy noise addition stage.

Data normalization stage: to avoid the influence of the magnitude, each column is normalized, after which a normalized matrix X is formed.

Calculation of covariance matrix stage: each p -dimensional divided-block matrix X after normalization is computed with its covariance matrix A , $A = (1/p)XX^T$.

Privacy budget parameter assignment stage: assigning different privacy budget parameters according to the level of privacy protection for each attribute. For differential privacy, the smaller the budget parameter, the higher the corresponding privacy protection level and the lower the availability of data. Therefore, this stage is assigning the privacy parameter to each attribute in the ratio 1:9:90 according to the high-level privacy attribute, medium-level privacy attribute, and low-level sensitive attribute. After that, the privacy parameters of all attributes in each divided-block matrix are summed to obtain the privacy parameter of each divided-block matrix ϵ_i ($1 \leq i \leq q$).

Privacy noise extraction stage: since the covariance matrix has properties such as symmetric and positive semidefinite, the perturbation noise added to the covariance needs to satisfy these two properties. Specifically, for a divided-block matrix with m rows and p columns, a positive semidefinite matrix C with m rows and m columns is first generated, which is required to satisfy that all m eigenvalues are equal and all eigenvalues are set to $(3/(2p\epsilon_i))$. Then, a noisy sample matrix W with m rows and p columns is extracted from the Wishart distribution $W(m+1, C)$.

Data privacy noise addition stage: the noise sample matrix W noise is added to the covariance matrix A , which forms the noise-added covariance matrix A' , $A' = A + W$. This ensures that the entire divided-block matrix satisfies differential privacy. For adjacent divided-block matrix matrices (two divided-block matrices differ in only one element), their outputs for statistical results are almost identical after noise perturbation. In this way, even if an attacker has all the information except a certain element, he cannot get the specific value of that element, which provides strong privacy protection.

4.2.5. Data Transformation Stage. Data transformation stage: an algorithm based on the sparse matrix transformation of the eigenvector matrix estimation is used to construct a low-dimensional projection. And the orthogonal eigenmatrix of the sparse matrix transformation can be represented as a series of finite successive Givens rotations. The following S is an unbiased estimate of A' ; the number of iterations is H . The specific implementation using the sparse matrix transform is as follows.

For $h = 1$ to H :

(i) Adjust i, j so that it satisfies

$$(i_h, j_h) \leftarrow \arg \min_{(i,j)} \left(1 - \frac{[S_h]_{ij}^2}{[S_h]_{ii}[S_h]_{jj}} \right). \quad (12)$$

(ii) Calculate the angle

$$\theta_h = \frac{1}{2} a \tan(-2[S_h]_{i_h j_h}, [S_h]_{i_h i_h} - [S_h]_{j_h j_h}). \quad (13)$$

(iii) Each round of the sparse matrix G_h is a Givens rotation transformation of the coordinates (i_h, j_h) , and I is a unit matrix.

$$G_h = I + \Theta(i_h, j_h, \theta_h). \quad (14)$$

(iv) Compute

$$G_h^* = \arg \min_{G_h} |\text{diag}(G_h^T S_h G_h)|. \quad (15)$$

(v) Calculate

$$S_{h+1} = G_h^{*T} S_h G_h^*, \quad (16)$$

where G_h is obtained at each step using the iterative method, and the number of iterations H can be obtained using a cross-validation procedure [20]. Then, using this sparse matrix transformation method, the eigenvector matrix

$$\hat{E} = G_1^* G_2^* \dots G_H^*, \quad (17)$$

is obtained, and then

$$\hat{\Lambda} = \text{diag}(\hat{E}^T S_H \hat{E}), \quad (18)$$

is obtained to get the eigenvalues, after which the eigenvectors are arranged from the largest to smallest according to their corresponding eigenvalues, then, the cumulative contribution $\sum_{i=1}^k \lambda_i$ is calculated, the eigenvalues whose sum of the first k maxima of cumulative contribution percentage is greater than w are formed $\Lambda_k = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$, and the matrix of eigenvectors corresponding to these eigenvalues is obtained E_k . Usually, the more forward the order of the principal component is, the more sufficient information it contains. Then, the eigenvectors matrix is transposed and multiplied with the normalized matrix to obtain a low-dimensional projection matrix $F_{m \times k} = X_{m \times p} E_{p \times k}$ of k -dimensions ($k \leq p$).

4.2.6. Encrypted Transmission Stage. Encrypted transmission stage: the publisher uses the k -dimensional eigenvector matrix $E_{p \times k}$ as the key and delivered it to the legitimate data requester over a secure channel.

4.2.7. Intermediate Data Publishing Stage. The data publisher uploads the low-dimensional projection matrix $F_{m \times k}$ to the blockchain for data publishing and sharing. The

intermediate data (projection matrix) is protected by differential privacy technology and published to blockchain to ensure security. The decentralization, time-series data, collective maintenance, and security and trustworthiness of blockchain technology facilitate real-time supervision of the sharing process, record the sharing events, and improve the effectiveness of the data publishing process.

Sections 4.2.8 and 4.2.9 address the data requester for data recovery.

4.2.8. Data Recovery Stage. The data recovery stage contains two substages, the low-dimensional matrix recovery stage and the normalization recovery stage.

Low-dimensional matrix recovery stage: after the data requestor initiates a request to the data publisher and gets the permission, the k -dimensional eigenvector matrix $E_{p \times k}$ can be obtained. After that, the eigenvector matrix is multiplied with the k -dimensional projection matrix to get the recovery matrix. $Y_{m \times p} = F_{m \times k}(E_{p \times k})^T$.

Normalization recovery stage: each element of the recovery matrix $Y_{m \times p}$ would be transformed by inverse operation of the normalization; also, the column filled with zeros at the data preprocessing stage in the last divided block is removed.

4.2.9. Data Stitching Stage. All the divided-block matrices are stitched together to form a new matrix with m rows and n columns, and the corresponding attribute names are added to the data table header. After that, the attributes of the new matrix are adjusted and recovered in the order of the attributes in the original data table. At the end of this stage, the new data table has the same rows and attribute columns as the original input data table. Then, the new data can be analyzed for statistical significance by the data requester.

4.3. Differential Privacy Protection Analysis. For a massive high-dimensional matrix (m rows and n columns) and the total privacy budget parameter ϵ , after being divided into q matrices, the privacy parameter corresponding to each divided-block matrix is ϵ_i . And $\epsilon = \sum_{i=1}^q \epsilon_i$.

The noise-added projection matrix $F_{m \times k}$ satisfies the ϵ_i -differential privacy of its corresponding privacy parameter ϵ_i in the divided-block matrix, and the specific proof process is as follows.

For each divided-block matrix X (m rows and p columns) and its corresponding assigned privacy parameter ϵ_i , the privacy noise extraction stage of Section 4.2.4 uses Wishart distribution privacy-preserving noise addition, and according to [13], for each divided-block matrix, after being added privacy noise perturbation on covariance, the new noise-added covariance of each divided-block matrix satisfies ϵ_i -differential privacy. Assume that for adjacent inputs X and X' (only one element of the matrix X is changed to obtain X') the output is almost identical and the output is denoted by $A + W_0$.

$$\frac{\Pr(A + W = A + W_0)}{\Pr(A' + W = A + W_0)} \leq e^{\epsilon_i}. \quad (19)$$

By the postprocessing invariance of differential privacy [12, 13], given any algorithm A_1 satisfying ϵ_i -differential privacy, for any algorithm A_2 (A_2 does not necessarily satisfy differential privacy), there is $A(*) = A_2(A_1(*))$ satisfying ϵ_i -differential privacy. This property illustrates that differential privacy is conducive to postprocessing; i.e., the output that satisfies ϵ_i -differential privacy in the previous process will automatically retain the property of satisfying ϵ_i -differential privacy as the input to the next stage. Therefore, after adding the privacy-preserving noise perturbation satisfying ϵ_i -differential privacy to the covariance matrix of the divided-block matrix, the ϵ_i -differential privacy is still satisfied after the sparse matrix transformation to reduce the dimensionality. That is, it still satisfies the following:

$$\frac{\Pr(A + W = A + W_0)}{\Pr(A' + W = A + W_0)} \leq e^{\epsilon_i}. \quad (20)$$

When each divided-block matrix satisfies differential privacy, according to the property of parallel composition [22] of differential privacy, when there are multiple sequences of algorithms acting on several different subsets of a dataset, the final differential privacy budget is equivalent to the maximum of all algorithm budgets in the sequence of algorithms, i.e., satisfying $\max_{i \in [1, 2, \dots, q]} \epsilon_i$ -differential privacy.

4.4. Efficient Privacy Data Publishing and Sharing Model Based on the Blockchain. When the owners publish the intermediate data (projection matrix with privacy protection) to be shared, it is stored on the blockchain after differential privacy processing to ensure the information security and tamper-resistant of data content. At the same time, in the process of data publishing and sharing, the original privacy data owned by each participant does not want to be directly accessed by other requesting participants. Compared with the public blockchain, the consortium blockchain has a smaller degree of openness and scale to provide more privacy protection for the solution. Therefore, we depend on the consortium blockchain system to cope with the problem of mutual distrust when the data is published and shared. The Consensus Ledger ensures that no participant can tamper with the data. At the same time, in the whole system, no participant involved in data publishing and sharing has direct access to the original real data of a certain data publisher. A schematic diagram of the Efficient Privacy Data Publishing and Sharing Model based on the blockchain is shown in Figure 2.

4.4.1. Efficient Privacy Data Publishing and Sharing Model Based on the Blockchain Includes the Following Three Parts

- (1) Data publisher: it includes data owners such as government agencies and enterprise companies. In the system, the data publisher will join the consortium blockchain as a node.

FIGURE 2: A schematic diagram of Efficient Privacy Data Publishing and Sharing Model based on the blockchain.

- (2) Consortium blockchain platform: the data from the publisher will form intermediate data (projection matrix with privacy protection), and the intermediate data will be uploaded and stored to the platform for sharing through a secure channel.
- (3) Data requester: after the data requester obtains the published intermediate data from the consortium blockchain, it needs to request the key (eigenvector matrix) provided by the data publisher to complete the recovery calculation of the intermediate data so as to obtain the privacy-protected data, after which data analysis and mining can be performed.

4.4.2. System Workflow

- (1) Each publisher stores the privacy-protected intermediate data (projection matrix) into the node of the consortium blockchain.
- (2) The privacy-protected intermediate data (projection matrix) recorded in the Consensus Ledger can be viewed in the consortium blockchain. Any participant within the system can access the intermediate data published and shared with other participants. Then, the data requester can complete the recovery calculation of the intermediate data by using the keys (eigenvector matrix) provided by the data publisher.
- (3) Since the statistical significance of the recovered data which is protected by differential privacy is almost the same as that of the original data, data analysis and mining can be performed by the data requester.

4.4.3. Algorithm Security Analysis. This algorithm protects the secure data publishment in two links: the differential privacy protection and secure channel encryption transmission the key (k -dimensional eigenvector matrix $E_{p \times k}$).

The first link: differential privacy perturbation is added to the covariance matrix of the divided-block matrix to secure the privacy.

For each divided-block matrix, if the privacy perturbation noise is not added in the whole data publishing stage, after dimensionality reduction, the attacker may steal the low-dimensional matrix $F_{m \times k} = X_{m \times p} E_{p \times k}$ from the data publisher and the recovery matrix $Y_{m \times p} = F_{m \times k} (E_{p \times k})^T$ from the data requester. The attacker can compute $F_{m \times k}^{-1} Y_{m \times p}$ to obtain the k -dimensional eigenvector $E_{p \times k}$, which in turn can compute $Y_{m \times p} (E_{p \times k}^T)^{-1} (E_{p \times k})^{-1}$ to obtain the original divided-block matrix to get the real data. Similarly, if the attacker can obtain the low-dimensional matrix $F_{m \times k}$ and the recovery matrix $Y_{m \times p}$ of each divided-block matrix, the complete real dataset can be recovered, and the whole information can be deciphered. After adding the differential privacy noise to the covariance matrix of the divided-block matrix, the low-dimensional matrix $F_{m \times k}$ and recovery matrix $Y_{m \times p}$ are generated with differential privacy noise perturbation and are not the original values. Therefore, for the adversary, the calculation result of $Y_{m \times p} (E_{p \times k}^T)^{-1} (E_{p \times k})^{-1}$ includes noise perturbation in each step. As a result, the noise perturbation finally becomes very large after constant multiplication and accumulation and thus deviate from the original data. Therefore, this also guarantees the secure publishment of the private data.

The second link: secure channel to transfer the key (k -dimensional eigenvector matrix $E_{p \times k}$).

Each divided-block matrix generates the projection matrix $F_{m \times k}$ and k -dimensional eigenvector matrix $E_{p \times k}$ after the dimensionality reduction process. We know from the first link analysis that $E_{p \times k}$ has a crucial role for the adversary to decipher and recover the original information. Once the adversary has directly obtained $E_{p \times k}$ and stolen the $Y_{m \times p}$, when there is no differential

privacy perturbation, $Y_{m \times p} (E_{p \times k}^T)^{-1} (E_{p \times k})^{-1}$ can be calculated to obtain the original divided-block matrix. Therefore, it is also important to protect the k -dimensional eigenvector matrix $E_{p \times k}$. Therefore, we design such a step as follows: the publisher needs to assign the k -dimensional eigenvector matrix $E_{p \times k}$ as a key. Then, the data requester needs to pass the authentication of the data publisher to obtain the k -dimensional eigenvector matrix $E_{p \times k}$ by a secure channel. And the adversary does not have the ability to get the k -dimensional eigenvector matrix $E_{p \times k}$ in the middle of the channel.

The above two links work in a two-pronged way to double guarantee the secure publishment of sensitive private information.

4.4.4. Analysis of Adversary Attack Capabilities

- (1) Network eavesdropper: the attacker illegally obtains the data in the transmission process through network eavesdropping or data interception to achieve some purposes.
- (2) Not-completely-honest participants: in the process of data publishing and sharing, the data requester may be incompletely honest participants. Such participants usually follow the established process and perform each step correctly to achieve the original purpose, but at the same time, they want to steal the data of other participants for some purposes and recover, infer, and snoop on the data of other participating publishers by computing the intermediate data.
- (3) Strong background knowledge attack adversary: it means that the adversary has a large amount of background knowledge and is able to break and deduce some conventional data encryption methods.

4.4.5. Analysis of the System Security Capability to Resist Attacks

- (1) The first is to make it impossible for a network eavesdropper to steal the private data in the communication traffic of two or more participants, during the communication between the participants of the system
- (2) The second is to ensure that the system is able to protect individual privacy while keeping the statistical significance almost constant, even in the face of strong background knowledge attacks
- (3) The third is to ensure that any data requester can recover from the intermediate data only after getting the key (k -dimensional eigenvector matrix $E_{p \times k}$) of the designated data publisher and to prevent the data requester from getting the key of the nondesignated data publisher
- (4) The fourth is to prevent the data published and shared in the system from being maliciously

tampered with by not-completely-honest participants during storage

4.4.6. System Security Capability Analysis. To achieve the first security objective, the data needs to be transmitted over a secure communication channel in order to effectively prevent attacks by network eavesdroppers.

To achieve the second security objective, the projection matrix is perturbed by differential privacy to make it resistant to strong background knowledge attacks.

To achieve the third security objective, the system encrypts the k -dimensional eigenvector matrix $E_{p \times k}$ of the data publisher in the dimensionality reduction, so that the data requester can recover the intermediate data only after getting the key (k -dimensional eigenvector matrix $E_{p \times k}$) of the data publisher. As long as the attacker cannot obtain the key of the designated publisher, even if he obtains the intermediate data (projection matrix) shared on the blockchain, he cannot recover the data, which ensures the privacy and security of the data. At the same time, when an internal participant wishes to obtain the private information of other non-designated publishers even if he has an illegal purpose, he will not be able to recover the data if he cannot obtain the key from the designated data publisher. This also resists stealing by not-completely-honest participants inside the system.

To achieve the fourth security objective of reliable storage of shared data, the system uses a decentralized storage method with the blockchain platform to ensure that intermediate data is open and tamper-proof between participants, and then, the data privacy is guaranteed. This system can resist tampering attacks by not-completely-honest participants inside the system.

In summary, the system proposed in this paper can effectively protect the privacy and security of data publishing and sharing among participants.

5. Experimental Results and Analysis

5.1. Experimental Setup. In order to reflect the accuracy and scientificness, this experiment uses the publicly available dataset Residential Building [23] from the UCI machine learning database for simulation analysis, which contains 372 records and 109 attributes, including numerous sensitive information types. One computer (with Intel Core i7-1065G7 1.30 GHz CPU, 16 GB RAM, and Windows 10 operating system, as the main running environment) is used for this simulation experiment. Since the Laplace noise perturbation is random, each group of experiments is repeated 5 times, and the average value is taken as the result to record.

5.2. Comparison of Mean Square Error. In this section, mean square error (MSE) is selected as the indicator to evaluate the impact of sparse matrix transform on data availability in the dimensionality reduction stage; we also compare the performance of the algorithm proposed in this paper with other traditional classical algorithms in terms of mean square error.

5.2.1. Exploring the Effect of Traditional QR Decomposition Householder Transform and Sparse Matrix Transform on the Mean Square Error in the Dimensionality Reduction Stage. In this experiment, we will compare the performance of the sparse matrix transformation scheme and the traditional QR decomposition Householder transform method in terms of mean square error with different privacy budget parameters of differential privacy.

From Table 2 and Figure 3, it can be concluded that the mean square error of the sparse matrix transformation scheme is smaller than that of the QR decomposition Householder transformation. In order to explore the specific reasons for this result, the experiment in Section 5.2.2 is then performed.

5.2.2. Exploration to Compare the Cumulative Contribution Percentage of Eigenvalues of the Traditional QR Decomposition Householder Transform Scheme with the Sparse Matrix Transform Scheme. This experiment will compare the eigenvalue contribution percentage ($\sum_{i=1}^k \lambda_i / \sum_{i=1}^p \lambda_i$) of the sparse matrix transform with the traditional QR decomposition Householder transform in the dimensionality reduction stage. The eigenvalue contribution percentages of the QR decomposition Householder transform and sparse matrix transform are shown in Table 3.

Table 3 shows the cumulative contribution percentage after processing the Residential Building dataset by two dimensionality reduction methods, respectively. And combined with Figure 4, the cumulative contribution of principal components of sparse matrix transform is higher to QR decomposition Householder transform in the same dimensionality, and the sparse matrix transform retains more information than QR decomposition transform. When the same privacy-preserving noise is applied to the sparse matrix transform scheme compared to the QR decomposition Householder transform, the sparse matrix transform scheme is relatively less perturbed, so the mean square error is relatively smaller using the sparse matrix transform scheme compared to the traditional QR decomposition Householder transform. And in the same cumulative contribution percentage, the number of principal components of sparse matrix transform is less than that of QR decomposition Householder transform, so the total noise added is smaller for the same privacy-preserving budget.

5.2.3. Exploring the Effect of Privacy Parameter on Different Data Publishing Algorithms. This experiment tests the effect of privacy parameter ϵ on the DSMT-DP publishing algorithm and compares it with other traditional schemes. To evaluate the effectiveness of the DSMT-DP algorithm, mean square error (MSE) is selected as the evaluation metric in this paper. Meanwhile, 50 numerical attributes of 109 attributes in the Residential Building dataset are selected for this experiment.

Table 4 and Figure 5 show the MSE comparison of DSMT-DP with the traditional differential privacy algorithm of the Laplace mechanism and PCA-based PPDP [5] in different

privacy budget parameters ϵ . The results show that MSE of DSMT-DP is smaller than that of the Laplace mechanism, which has obvious advantages in effectiveness; when comparing the experimental effects with the PCA-based PPDP mechanism for differential privacy, the number of principal components is selected according to the same contribution rate, which follows the univariate principle of the comparison experiment. The comparison results show that the MSE of DSMT-DP is slightly lower than that of the PCA-based PPDP mechanism.

5.3. Comparison of Computing Time Efficiency

5.3.1. Experimental Comparison for the Effect of the Divided-Block Scheme. The DSMT-DP algorithm which uses a divided-block scheme to calculate covariance is compared with the SMT-DP (without divided-block) scheme in privacy-preserving operation processing time. In this experiment, we select 50 numerical attributes of 109 attributes of the Residential Building dataset for the experiment.

From Table 5, it can be obtained that the DSMT-DP scheme, which uses a divided-block scheme to compute covariance, is more efficient than the SMT-DP (without divided-block) scheme. The reason is that the covariance matrix for computing high-dimensional datasets is very computationally intensive while converting a large covariance matrix operation into multiple parallel small covariance operations can reduce the number of calculation operations appropriately and save more computing overhead.

5.3.2. Exploring the Comparison of the Efficiency of Different Dimensionality Reduction Schemes. Different dimensionality reduction schemes are conducted for experiments on the same dataset, and the traditional QR decomposition Householder transform and divided-block sparse matrix transformation are used for comparison, respectively. In this experiment, we select 50 numerical attributes of the Residential Building dataset for the experiment.

We also selected 30 attributes of 35 attributes in the Predict keywords activities in Online Social Media Data Set [24] in UCI for the experiment.

From Tables 6 and 7, it can be concluded that the dimensionality reduction using sparse matrix transformation scheme has smaller computing time and higher efficiency than the traditional QR decomposition Householder transform scheme.

5.3.3. Exploring the Comparison of DSMT-DP with the Traditional Privacy-Preserving Publishing Algorithm. DSMT-DP is compared with the traditional principal component analysis differential privacy algorithm PCA-based PPDP on the same dataset for the processing time of privacy-preserving operations. In this experiment, we select 50 numerical attributes of 109 attributes in the Residential Building dataset for this experiment.

We also selected 30 numerical attributes of 35 attributes in the social media dataset in UCI for this experiment.

TABLE 2: Effect of traditional Householder transform and sparse matrix transform on the mean square error.

Privacy budget parameters	0.1	0.4	0.7	1	1.3	1.6
Wishart perturbation with Householder transform dimensionality reduction	92432	31585	8246	4329	2054	317
DSMT-DP (with Wishart perturbation and SMT dimensionality reduction)	64972	21426	5373	2324	1508	219

FIGURE 3: Effect of traditional Householder transform and sparse matrix transform on the mean square error.

TABLE 3: Cumulative contribution percentage of eigenvalues of the tradition QR decomposition Householder transform and sparse matrix transform.

Cumulative number of principal components	Sparse matrix transforms (%)	Householder transform (%)
1	62.62	56.62
2	74.53	68.22
3	80.12	75.04
4	85.14	80.95
5	89.73	85.78
6	92.50	90.07
7	95.12	93.49

FIGURE 4: Comparison of the cumulative contribution percentage of eigenvalues using the traditional QR decomposition Householder transform scheme and the sparse matrix transform scheme.

TABLE 4: The impact of privacy parameters ϵ on MSE of different publishing algorithms compared.

	$\epsilon = 0.1$	$\epsilon = 0.4$	$\epsilon = 0.7$	$\epsilon = 1.0$	$\epsilon = 1.3$	$\epsilon = 1.6$
Laplace	663215	212546	74296	35459	8954	6923
PCA-based PPDP	95835	53638	23673	8162	3349	1766
DSMT-DP	64972	21426	5373	2324	1508	219

FIGURE 5: The impact of privacy parameters ϵ on MSE of different publishing algorithms compared.

TABLE 5: Comparison of DSMT-DP algorithm and SMT-DP algorithm in operation time consumption.

Algorithm	Operation time consumption (seconds)
DSMT-DP	$2.17 \times 10^{-1}s$
SMT-DP (without divided-block)	$6.39 \times 10^{-1}s$

TABLE 6: Comparison of operation time consumption using Householder transform and sparse matrix transform on Residential Building dataset.

Dimensionality reduction program	Operation time consumption (seconds)
Householder	$4.76 \times 10^{-1}s$
Divided-block sparse matrix transformation	$2.17 \times 10^{-1}s$

TABLE 7: Comparison of operation time consumption using Householder transform and sparse matrix transform on the Predict keywords activities in online social media dataset.

Dimensionality reduction program	Operation time consumption (seconds)
Householder	$2.08 \times 10^{-1}s$
Divided-block sparse matrix transformation	$4.06 \times 10^{-2}s$

TABLE 8: Comparison of operation time consumption using DSMT-DP and PCA-based-PPDP on Residential Building dataset.

Algorithm	Operation time consumption (seconds)
DSMT-DP	$2.17 \times 10^{-1}s$
PCA-based PPDP	$8.28 \times 10^{-1}s$

TABLE 9: Comparison of operation time consumption using DSMT-DP and PCA-based PPDP on the Predict keywords activities in an online social media dataset.

Algorithm	Operation time consumption (seconds)
DSMT-DP	$4.06 \times 10^{-2}s$
PCA-based PPDP	$3.92 \times 10^{-1}s$

From Tables 8 and 9, it can be concluded that DSMT-DP takes less time than PCA-based PPDP in terms of privacy-preserving operation processing time. This indicates that the

DSMT-DP algorithm is better than the traditional PCA-based PPDP algorithm in terms of operation time consumption.

5.4. Summary of Experimental Results. Combined with the above experimental analysis results, we can conclude that, with the same privacy budget parameter, the sparse matrix transformation scheme can relatively reduce the mean square error. And the mean square error and the data distortion of DSMT-DP are smaller compared with the traditional principal component analysis differential privacy algorithm; at the same time, the divided-block scheme can indeed further optimize the covariance matrix operation on top of the original one, which has a significant improvement on the operation efficiency; furthermore, DSMT-DP algorithm scheme is more efficient than the traditional principal component analysis differential privacy algorithm PCA-based PPDP in the operation processing of the privacy dataset.

6. Conclusions

For the privacy protection problem of massive high-dimensional datasets, this paper designs a Divided-block Sparse Matrix Transformation Differential Privacy Data Publishing Algorithm (DSMT-DP) and proves that this algorithm satisfies differential privacy. Compared with the traditional principal component analysis differential privacy algorithm, this algorithm takes into account the different privacy protection requirements of each attribute and assigns the privacy budget parameters according to the sensitivity level of each attribute; the mean square error of this method is smaller than that of the traditional principal component analysis differential privacy algorithm with the same privacy budget parameters; the method of the divided-block scheme and sparse matrix transformation can take less time in the dimensionality reduction stage and improve the computing efficiency of privacy-protected data publication.

Further, we combine this algorithm with blockchain and propose an Efficient Privacy Data Publishing and Sharing Model based on the blockchain. Publishing and sharing data on this model not only resist strong background knowledge attacks from adversaries outside the system but also prevent stealing and tampering of data by not-completely-honest participants inside the system.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61932015), Shaanxi Innovation Team Project (2018TD-007), and China 111 Project (B16037).

References

- [1] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: a survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
- [2] S. Wold, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1, pp. 37–52, 1987.
- [3] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proceedings of the SIGACT. 41st ACM Symposium on Theory of Computing (STOC 2009)*, pp. 371–380, ACM, Bethesda, MD, USA, May 2009.
- [4] M. Hardt and A. Roth, "Beating randomized response on incoherent matrices," in *Proceedings of the SIGACT. 44th ACM Symposium on Theory of Computing (STOC 2012)*, pp. 1255–1268, ACM, New York, USA, May 2012.
- [5] X. Jiang, Z. Ji, S. Wang, N. Mohammed, S. Cheng, and L. Ohno-Machado, "Differential-private data publishing through component analysis," *Transactions on data privacy*, vol. 6, no. 1, pp. 19–34, 2013 Apr, PMID: 24409205; PMCID: PMC3883117.
- [6] C. Peng, Y. Zhao, and M. Fan, "A differential private data publishing algorithm via principal component analysis based on maximum information coefficient," *Netinfo Security*, vol. 20, no. 2, pp. 37–48, 2020.
- [7] M. Hardt and A. Roth, "Beyond worst-case analysis in private singular vector computation," in *Proceedings of the forty-fifth Annual ACM Symposium on Theory of computing*, pp. 331–340, ACM, Palo Alto, CA, USA, June 2013.
- [8] M. Hardt and E. Price, "The noisy power method: a meta algorithm with applications," *Advances in Neural Information Processing Systems*, vol. 2014, no. 27, pp. 2861–2869, 2014.
- [9] K. Chaudhuri, A. Sarwate, and K. Sinha, "Near optimal differentially private principal components," *Advances in Neural Information Processing Systems*, vol. 4, pp. 989–997, 2012.
- [10] M. Kapralov and K. Talwar, "On differentially private low rank approximation," in *Proceedings of the Twenty Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1395–1414, SIAM, New Orleans Louisiana, January 2013.
- [11] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang, "Analyze gauss: optimal bounds for privacy-preserving principal component analysis," in *Proceedings of the SIGACT. 46th ACM Symposium on Theory of Computing (STOC 2014)*, pp. 11–20, ACM, New York, USA, New York, May 2014.
- [12] D. Kifer and B. R. Lin, "Towards an axiomatization of statistical privacy and utility," in *Proceedings of the Twenty-Ninth ACM Sigmod-Sigact-Sigart Symposium on Principles of Database Systems, PODS 2010*, pp. 147–158, Indianapolis, Indiana, USA, June 2010.
- [13] W. Jiang, C. Xie, and Z. Zhang, "Wishart mechanism for differentially private principal components analysis," in *Thirtieth Conference on Artificial Intelligence (AAAI-16) was held at the Phoenix Convention Center and the Hyatt Regency Phoenix in*, vol. 30, no. 1, Phoenix, AZ, USA, February 2016.
- [14] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: the sulq framework," in *Proceedings of the Twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 128–138, ACM, Baltimore Maryland, June 2005.
- [15] A. Alnemari, S. Arodi, V. Sosa et al., "Protecting infrastructure data via enhanced access control, blockchain and differential privacy," in *Proceedings of the 12th International Conference on Critical Infrastructure Protection (ICCIP)*, Arlington, VA, United States, Mar 2018.

- [16] Y. Liu, J. Peng, J. Kang, A. M. Ilyasu, D. Niyato, and A. A. Abd El-Latif, "A secure federated learning framework for 5G networks," *IEEE Wireless Communications*, vol. 27, 2020-05-12.
- [17] W. Jiang, Z. Ma, S. Li, H. Xiao, and J. Yang, "Privacy budget management and noise reusing in multichain environment," *International Journal of Intelligent Systems*, pp. 1–14, 2021.
- [18] C. Dwork, "Differential privacy," in *Proceedings of the International Colloquium on Automata, Languages, and Programming*, pp. 1–12, Springer, Berlin, Heidelberg, July 2006.
- [19] N. Mohammed, R. Chen, B. C. M. Fung, and P. S. Yu, "Differentially private data release for data mining," in *Proceedings of the SIGACT. 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 493–501, ACM, San Diego, California, USA, August 21–24, 2011.
- [20] G. Cao and C. A. Bouman, "Covariance estimation for high dimensional data vectors using the sparse matrix transform," vol. 21, pp. 225–232, in *Proceedings of the 21st International Conference on Neural Information Processing Systems*, vol. 21, pp. 225–232, MIT Press, Vancouver BritishCanada, December 2008.
- [21] J. Peng and T. Luo, "Sparse matrix transform-based linear discriminant analysis for hyperspectral image classification," *Signal, Image and Video Processing*, vol. 10, no. 4, pp. 761–768, 2016.
- [22] F. Mcsherry, "Privacy integrated queries," *Communications of the ACM*, vol. 53, no. 9, pp. 89–97, 2010.
- [23] <https://archivebeta.ics.uci.edu/ml/datasets/residential+building+data+set>.
- [24] <http://archive.ics.uci.edu/ml/datasets/Predict+keywords+activities+in+a+online+social+media>.