Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.025159

ARTICLE

Multi-Source Data Privacy Protection Method Based on Homomorphic Encryption and Blockchain

Ze Xu and Sanxing Cao*

State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing, 100024, China

*Corresponding Author: Sanxing Cao. Email: sxcao@cuc.edu.cn

Received: 24 June 2022 Accepted: 13 September 2022

ABSTRACT

Multi-Source data plays an important role in the evolution of media convergence. Its fusion processing enables the further mining of data and utilization of data value and broadens the path for the sharing and dissemination of media data. However, it also faces serious problems in terms of protecting user and data privacy. Many privacy protection methods have been proposed to solve the problem of privacy leakage during the process of data sharing, but they suffer from two flaws: 1) the lack of algorithmic frameworks for specific scenarios such as dynamic datasets in the media domain; 2) the inability to solve the problem of the high computational complexity of ciphertext in multi-source data privacy protection, resulting in long encryption and decryption times. In this paper, we propose a multi-source data privacy protection method based on homomorphic encryption and blockchain technology, which solves the privacy protection problem of multi-source heterogeneous data in the dissemination of media and reduces ciphertext processing time. We deployed the proposed method on the Hyperledger platform for testing and compared it with the privacy protection schemes based on k-anonymity and differential privacy. The experimental results show that the key generation, encryption, and decryption times of the proposed method are lower than those in data privacy protection methods based on k-anonymity technology and differential privacy technology. This significantly reduces the processing time of multi-source data, which gives it potential for use in many applications.

KEYWORDS

Homomorphic encryption; blockchain technology; multi-source data; data privacy protection; privacy data processing

1 Introduction

With the evolution of the Internet, network topology is becoming more complex, and the scale is also expanding. The number of Internet users continues to grow, and the amount of global data is doubling every year. According to market intelligence firm IDC, the global data volume will increase to 175 zettabytes in 2025. The transmission, storage, and mining of the large amounts of multi-source data will become increasingly important. If privacy protection measures are not taken, users' data security and privacy will be compromised [1]. User data can be leaked by malicious attacks by hackers, phishing websites, webpage forgery, and theft, which can bring huge economic losses to enterprises and threaten the security of users' sensitive information. As long as the problem of data privacy goes unsolved, IT innovation will be restricted, which will hinder the further evolution of the Internet.

In an era where the Internet is constantly changing, the connotation of privacy protection is constantly changing too. The development of converged media means that users continuously generate data and disseminate and share them through converged media. However, these data will expose privacy risks. At present, most research on privacy protection issues focuses on three points:

- Achieving reliable and efficient privacy protection in a decentralized network environment. In most Internet architectures, data generated by users are managed and secured by a central server provided by the service provider. However, as the amount of data increases, so does the load carried by the central server. Although bottlenecks can be avoided by distributed solutions, it still does not solve the problem of a single point of failure.
- Maintaining a balance between the openness of data sharing and the efficiency of privacy protection. With the rise of the digital economy, data are increasingly shared by users, particularly for the mining and utilization of data. However, when it comes to data sharing, it is difficult to balance user requirements and privacy protection. Although strong privacy protection technology can bring about the safe use of data, it requires complex operations, resulting in low computational efficiency, which in turn restricts the ability to share data.
- Protecting both data privacy and identity privacy in complex network environments. In a complex and dynamic network environment, the data generated by users cannot be predicted in real-time through the network. Traditional data desensitization, data cleaning, and password technologies can only partly protect privacy. Dynamic adjustment of a combination of privacy protection schemes is needed to achieve complete data integrity. At the same time, for different scenarios, a single user identity cannot guarantee privacy and security. Users need to generate various virtual identities to obtain services in multiple security domains.

Existing privacy protection schemes are mainly based on specific algorithms for local datasets in specific scenarios. However, they lack an algorithmic framework for dynamic datasets in specific scenarios and lack a universal algorithm framework for dynamic datasets in multiple scenarios. Sun et al. [2] designed a location-selection attack algorithm for testing the security of the emerging Internet of Things (IoT). Yang et al. [3] proposed a privacy-preserving social media data publishing framework called PrivRank, which used a data perturbation mechanism to implement personalized ranking-based recommendations. Li et al. [4] introduced a differential privacy model based on data perturbation and proposed a privacy protection algorithm to achieve personalized and unified privacy protection of user trajectory data. To minimize the dependence on servers in a cloud environment, Yang et al. [5] proposed a decentralized secure cloud storage data access control scheme based on blockchain and attribute-based encryption (ABE). Protecting the privacy of shared information on social media platforms is a significant challenge. Zhu et al. [6] proposed a novel hybrid blockchain crowdsourcing platform to achieve decentralization and privacy preservation. However, the platform has a complex structure and uses a large amount of data, resulting in low operating efficiency.

To solve the abovementioned problems, we propose a multi-source data privacy protection method based on homomorphic encryption and blockchain technology. Homomorphic encryption can accomplish ciphertext data operation, and the decryption result directly corresponds to the operation result, thereby protecting data privacy. In addition, blockchain has the characteristics of decentralization and responsibility traceability. It can identify different sources in the link process and correspond the generated records to the same entity, to reduce the risk of leakage of users' information. Therefore, the proposed multi-source data privacy protection method encrypts the multi-source data and stores it in the blockchain. The information obtained after user access is encrypted data. This method can significantly reduce the processing time and encryption and decryption times of multi-source data and reduce the computational complexity of ciphertext in multi-source data privacy protection. We set the blockchain as a six-tier architecture and establish a data structure for ensuring data integrity. We design a data sharing protocol based on blockchain technology, send the protocol to the network in ciphertext, and generate blockchain parameters and public–private key pairs. A random number is decrypted using the blockchain parameters, and the approximate classification algorithm of ciphertext is designed using homomorphic encryption. The user decrypts the ciphertext with the private key and obtains the approximate classification results. On this basis, the proposed model protects the privacy of multi-source data.

The main contributions of this paper are summarized as follows:

- We propose a privacy protection method for multi-source heterogeneous data, which uses the characteristics of blockchain technology to provide safe and reliable conditions for the open sharing of media data and uses homomorphic encryption to process encrypted data to save computational time and speed up data circulation. The security of media data is guaranteed.
- To reduce the complexity of the ciphertext calculation, we propose a ciphertext approximate classification algorithm based on homomorphic encryption.
- Our experimental results show that our method has lower computational complexity and higher encryption and decryption efficiency than other privacy-preserving methods based on k-anonymity and differential privacy techniques.

The rest of this paper is organized as follows. Section 2 explains blockchain and homomorphic encryption and examines related work. Section 3 describes our proposed method. Section 4 describes the construction of the simulation experiment platform of our proposed method. Section 5 is devoted to a feasibility analysis of the proposed method. We present our final conclusions in Section 6.

2 Related Work

This section offers an overview of blockchain and homomorphic encryption. Then, we review the literature on privacy protection issues.

2.1 Blockchain

Blockchain is a distributed network architecture, involving functions such as data storage and file operations. In this paper, the blockchain network structure is set as a six-tier architecture, with each layer containing core content. Decentralized operations are performed through mutual coordination.

The six-tier architecture of blockchain is shown in Fig. 1. The bottom layer is the data layer, which uses data encryption and decryption technology to store transaction information, and each block is linked by hash pointers. In the network layer, the node realizes the user's communication function by receiving and sending information. The third layer is the consensus layer, which disperses and stores the node data. The fourth layer is the incentive layer. It encourages nodes to participate in blockchain transactions, and it charges transaction fees to control the balance of rewards and punishments for transactions. The fifth layer is the contract layer, which uses virtual machines to complete code operations, realize programmable operations, and improve transaction details. The top layer is the application layer, which achieves the information interaction between the application backend and the client through smart contracts [7]. The data layer is the main focus of this paper. To protect the privacy of multi-source data, the blockchain data structure is first constructed. Storage of a significant amount of multi-source data requires the establishment of a self-reference structure. First, a starting block is created, and then a block with the same structure as the starting block is generated, using the same rules [8]. Each block forms a chain structure in sequence, and the block of the previous block is connected to the block header of the next block. The data structure of the blockchain is shown in Fig. 2.

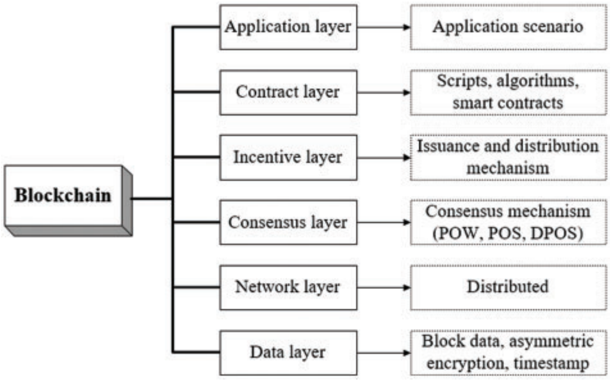
Blockchain
Application layer
Application scenario
Contract layer
Scripts, algorithms, smart contracts
Incentive layer
Issuance and distribution mechanism
Consensus layer
Consensus mechanism (POW, POS, DPOS)
Network layer
Distributed
Data layer
Block data, asymmetric encryption, timestamp

Figure 1: Six-layer architecture of blockchain

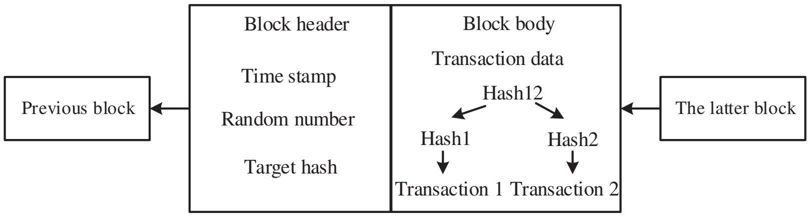

Figure 2: Blockchain data architecture

As shown in Fig. 2, the block header includes the version number, timestamp, transaction hash value, and random number. The hash calculation can convert data into a sign that has a close relationship with each byte of the source data. Each block obtains a pointer through hash calculation and holds it in the block header, then links the preceding and following blocks. This forms a linked-table storage structure. The block body contains the transaction information of the Merkle tree structure, and the data is divided into small data blocks, which correspond to the hash value one by one [9]. The hash value of the parent node is obtained by merging calculations upwards to update the data, the final root hash value is stored in the block body, and data integrity is ensured by fast induction.

2.2 Homomorphic Encryption

With the rapid growth of blockchain applications, research on privacy issues has become more important than ever [10]. The homomorphic encryption technology in cryptography has been introduced into the blockchain field to ensure the privacy of the blockchain in financial transaction scenarios. Homomorphic encryption is a special encryption method that directly processes the ciphertext and then encrypts the processing result after processing the plaintext, so that the obtained result is the same as the result of the same processing on the ciphertext. There are four types of homomorphism: addition, subtraction, multiplication, and division homomorphism.

Definition 1: Suppose $Enc(K, x)$ represents the use of the encryption algorithm *Enc* and key K to encrypt x, and F represents an n-ary operation. If the encryption algorithm *Enc* and the n-ary operation F satisfy:

$$Enc(K, F(x_1, x_2, \ldots, x_n)) = F(Enc(K, x_1), Enc(K, x_2), \ldots, Enc(K, x_n)). \tag{1}$$

The encryption algorithm *Enc* is homomorphic for the n-ary operation F. Homomorphic encryption can be partially, somewhat, hierarchical fully, or fully homomorphic encryption. Unlike most integer-based homomorphic encryption schemes, a novel homomorphic encryption method called CKKS was proposed by Cheon et al. [11] in 2017, which supported approximate addition and multiplication in the ciphertext state and could encrypt real numbers. This approach involved the evaluation of arbitrary circuits of bounded (pre-determined) depth. These circuits can include *ADD* (X-OR) and *Multiply* (AND). The HEAAN open-source homomorphic encryption software library uses a rescaling procedure for the size of the plaintext. It then produces an approximate rounding due to the truncation of the ciphertext into a smaller modulus. This method is especially useful in that it can be applied to carry-out encryption computations in parallel. Unfortunately, the ciphertext modulus can become so small that it cannot carry out any more operations. The HEAAN method uses approximate arithmetic over complex numbers (C) and is based on Ring Learning With Errors (RLWE). It focuses on defining an encryption error within the computational error that will occur in approximate computations.

Currently, the application of blockchain technology is still in its nascency. It was widely used in financial applications, but now there are attempts to apply it to the fields of copyright protection, supply chain management, gaming and entertainment, industrial IoT, social welfare, education, and medical care. In the traditional blockchain application model, the transaction data is open and transparent to every participant in the blockchain network. This means that the data has no privacy, and attackers can easily obtain key transaction information through network topology analysis [12], address tracking [13], transaction forwarding control [14], or malicious code mining technology [15]. Using homomorphic encryption technology, the lack of privacy in blockchain transactions has been resolved. During a financial transfer transaction, what is seen in the blockchain smart contract is the homomorphically encrypted ciphertext data, which is directly calculated from the ciphertext data to obtain the transferred amount. The data in the entire calculation process, including the data recorded in the blockchain ledger, is all encrypted by the homomorphic public key. Only the client node holding the corresponding private key can decrypt the personal data and view the plaintext. Other nodes cannot obtain the plaintext content, thereby ensuring the privacy of the entire financial transaction. Fig. 3 compares the traditional blockchain application model with the blockchain application model based on homomorphic encryption.

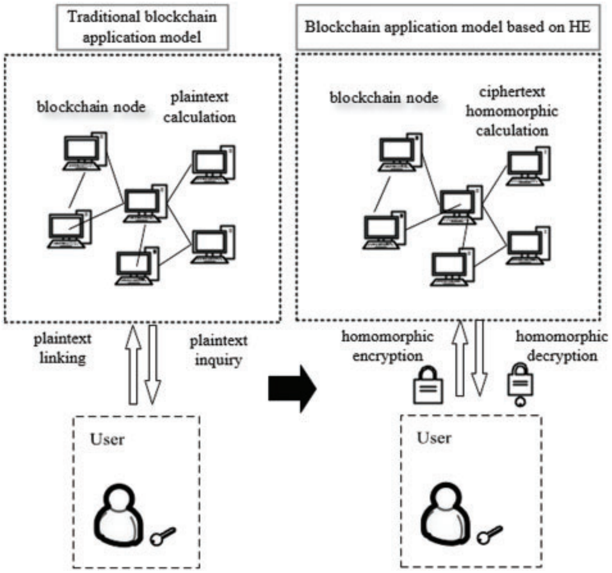

Figure 3: Comparison of two model architectures

2.3 Literature Review

In recent years, there has been a great deal of research to address privacy protection issues. Song et al. [16] proposed a k-anonymity privacy protection scheme based on bilinear pairing. Through the terminal, 2k false locations are uniformly generated within the Euclidean distance ring area, and k–1 false locations are filtered out using location entropy, location dispersion, and map background information, thereby achieving a better k-anonymity effect. This privacy protection scheme has strong security features and can resist tracking attacks. Feng et al. [17] proposed a study on location trajectory publishing technology based on a differential privacy model. They increased or decreased records in the statistical database to form two datasets, reduced the risk of privacy leakage by querying the datasets, and built a trajectory data protection model based on differential privacy technology to achieve real-time protection of continuous data. The existing location privacy enhancement technology based on k-anonymity technology and differential privacy technology to construct hidden areas has advantages in privacy protection and service quality. However, due to the large generation of hidden areas, the efficiency of query processing and communication is low. A data privacy protection method based on the DIKW architecture was proposed by Duan et al. [18]. By mapping the private content of multiple sources to the DIKW architecture data and information and using knowledge resources to model them, the content objects and relations are uniformly classified into typed resources of data, information, and knowledge. The architecture consists of a DIKW meta-model and extended data graph, information graph, and knowledge graph. The target privacy resources of data and information are divided into explicit and implicit, and protection solutions are proposed according to the explicit and implicit division of privacy targets of related types of data to achieve full data privacy protection. Qiao et al. [19] proposed an effective data privacy protection algorithm based on differential privacy and used a partitioning algorithm based on a greedy algorithm to obtain a better partition structure. They used wavelet transform to add noise. For the authenticity and usability of the histogram, the original histogram structure is restored. The complexity of the wavelet tree constructed by the wavelet transform is reduced, the query noise changes from linear growth to multiple logarithmic growth, and the accuracy of the histogram count query is improved.

Sun et al. [20] proposed a new privacy-preserving location-sharing service scheme for social networking applications, which enables users to browse the exact location of friends within a certain distance without sharing information with any other users or social media ISPs, so as not to reveal their location information. However, in the proposed scheme, location-sharing services do not take attacks from friends into account. Luo et al. [21] proposed research on multi-source data privacy protection based on an improved GBDT Federation integration method. The average gradient and the gradient of similar samples are taken as new gradients to improve the accuracy of the local model. Different ensemble learning methods are used to integrate the parameters of the local model, which improves the accuracy of the updated global model. Although all these methods can provide effective privacy protection, when new attacks appear, the computational complexity of the ciphertext is relatively high, the encryption and decryption times are relatively long, and there is still room for optimization.

The traditional method of privacy protection is a user authorization model. Its degree of protection depends on trusted third parties, and the breach of the centralized institution will lead to the leakage of data for a large number of users. The inherent privacy and security features of blockchain have inspired researchers to apply blockchain technology to solving privacy protection issues. Combined with cryptography, Qiao et al. [22] proposed a novel blockchain signature scheme based on an aggregated signature scheme for the privacy protection of transaction addresses in the blockchain. This scheme reduced the computational overhead of the signature and verification processes, reduced the storage overhead of the blockchain, and improved communication efficiency. Unlike the scheme proposed by Qiao et al. [22], Li et al. [23] constructed a blockchain privacy protection scheme based on ring signature. This scheme can ensure data security and user identity privacy security in blockchain applications, and its security is based on the elliptic curve discrete logarithm problem. Li et al. [24] systematically introduced the privacy protection scheme of encrypted currency—typically Monero, Zerocash, and Mixcoin—studied the privacy protection and supervision methods of blockchain user identity, and proposed a new direction of blockchain privacy protection anonymity and traceability technology. Muh et al. [25] discussed the integration of differential privacy at each layer of the blockchain and in some blockchain-based scenarios and proposed a new approach to protect blockchain data privacy using data perturbation strategies such as differential privacy. Liu et al. [26] designed a blockchain system based on the access scheme combined with the secure multi-computing protocol. Users in the system can only transmit data through authentication, which greatly improves privacy and security in complex multimedia communication scenarios.

3 Our Proposed Method

We first designed a data sharing protocol based on blockchain technology, in which data sharing is more secure. Then we generated blockchain parameters and public-private key pairs in blockchain transactions. To solve the high-order problem of ciphertext calculation, we introduced a ciphertext approximate classification algorithm based on homomorphic encryption. This led to our final multivariate data privacy protection model.

3.1 Data-Sharing Protocol Based on Blockchain Technology

Since transaction information is visible on the entire network, data can be easily mined and collected, causing an invasion of user privacy. The data protocol is sent to the blockchain network in the form of ciphertext, and the existence and rationality of the transaction can be verified without decryption. When the transaction initiator distributes information, it needs to provide its public key and the ciphertext of the recipient, and the ciphertext of the transaction contains the same plaintext information [27]. Due to the high cost of interaction with the blockchain, as the number of interactions increases, operating efficiency is significantly reduced. In this regard, the constructed data sharing protocol only needs to send a message to the receiver to realize the interaction, thereby reducing the amount of data interaction. The initiator uses its public key to encrypt the transaction amount to form a set of random vectors and uses the receiver's public key for incentives to obtain another set of random vectors [28]. At this time, the information in the ciphertext and the plaintext is equal. The formula for calculating the sum of the ciphertext of the transaction number and the random number is:

$$z_\alpha = x_j + y_i \tag{2}$$

where z_α represents the sum of ciphertexts; α epresents the initiator; x_j represents the ciphertext of the transaction amount; j represents the transaction amount; y_i represents the ciphertext of the random number; and i represents the random number. In the same way, the sum of the ciphertext of the receiver can be obtained. After receiving the ciphertext, we randomly select the number of bits from [0, 1] and give different answers to the receiver according to the number of bits. For example, when the number of bits is 0, the transmitted plaintext information of y_i and encrypted random vector are transmitted; when the number of bits is 1, the plaintext information of $x_j + y_i$ and encrypted random vector are sent. For the receiver on the blockchain to verify whether the transaction amount is consistent, the initiator only needs to publish the interactive information in the link once, and the ciphertext corresponds to the plaintext information or the random vector [29]. Assuming that the transaction amount and the random number have the private key of the homomorphic encryption scheme, one party has the point (α, β), and the other party has the point (χ, δ). The online formula for marking homomorphic encryption is as follows:

$$\frac{r\left(y_\delta - y_\beta\right)}{r\left(y_\chi - y_\alpha\right)} = \frac{y_\delta - y_\beta}{y_\chi - y_\alpha} = \frac{\Delta y}{\Delta x} \tag{3}$$

The verification random number can be used for the public key encryption of the initiator and the receiver, respectively. Since random numbers are uniformly distributed, the plaintext is also randomly distributed uniformly. When all ciphertexts are passed, the transaction satisfies the data-sharing agreement; otherwise, the receiver cannot obtain any information about the transaction.

3.2 Blockchain Parameters and Public–Private Key Pairs

When using homomorphic encryption technology to complete a transaction in the blockchain, the application layer not only needs to encrypt the data with the public key but also needs to generate evidence for the zero-knowledge proof. The generated evidence is then sent to each node together with the encrypted data, and the verification is completed. If the verification fails, the transaction is rejected and the result is returned to the application side [30]. If the verification is successful, the transaction data will be stored in the blockchain, and the transaction data update will be completed [31]. In this process, the transaction information of each node is in a ciphertext state, ensuring user privacy and security. Each blockchain network has its parameters, which, once set, cannot be modified, ensuring the uniqueness of the parameters during use. The process of parameter generation is as follows: First, two large prime numbers are randomly selected, and their product and common multiple parameters are calculated. The calculation formula is as follows:

$$\begin{cases} \chi = mn \\ \varphi = lcm\,(m-1, n-1) \end{cases} \tag{4}$$

where m and n represent random large prime numbers; χ represents a product parameter; φ represents a common multiple parameter; and lcm represents a least common multiple. Then, a random integer is selected and the common divisor parameter calculated. The calculation formula is as follows:

$$\begin{cases} \gamma = p \ mod \ \chi^2 \\ gcd\left(\chi, \dfrac{1}{\chi} p \bmod \chi^2\right) = 1 \end{cases} \tag{5}$$

In formula (5), γ represents the common divisor parameter; p represents a random integer; and gcd represents the greatest common divisor. The three parameters obtained above will be set and saved as blockchain parameters. After the parameters are generated, they are saved in the application and smart contract for use when the transaction is completed [32]. On this basis, the blockchain needs to generate public-private key pairs for each user. The public-private key pair is used as the transaction address at the same time, the public key is stored in the blockchain, and the private key is stored separately by the user [33]. The generation of public-private key pairs is related to blockchain parameters. In the range of less than χ, a random integer is selected as the user's private key, and the user's public key is generated according to the parameters χ and γ. This process can be expressed as:

$$k = \gamma \bmod \chi^2 \tag{6}$$

In formula (6), k represents the user's public key. When the application side initiates a transaction, it needs to encrypt the amount and other information. For plaintext information, we use blockchain parameters and public keys to generate the ciphertext to complete encryption. In the process of decrypting the plaintext, a random number is used to prove that the input and output data are equal, and the blockchain parameters are used to decrypt the random number.

3.3 Ciphertext Approximate Classification Algorithm Based on Homomorphic Encryption

Because the progressive order of the ciphertext is so high, the calculation order needs to be adjusted reasonably to reduce the complexity of the ciphertext calculation. This paper proposes a ciphertext approximate classification algorithm based on homomorphic encryption to reduce the approximate error of calculation and to improve security. The ciphertext data cannot be directly classified using a decision tree. In this paper, the decision tree model is transformed into a polynomial, and the homomorphic calculation of the ciphertext is promoted by the gradient. For a given decision tree, the threshold corresponding to its internal nodes corresponds to the category of the decision root node [34]. Among the generated polynomials, the ciphertext samples of unknown categories correspond to decision samples, the calculation result of decision depth is in ciphertext form, and the corresponding plaintext is 0 or 1. First, considering the number of ciphertext additions, the result of the left tree and the right tree of the internal node is added, which represents a ciphertext addition, and the result is still a ciphertext [35]. The ciphertext is recursively downward in turn until it reaches the leaf node. Then, considering the number of scalar multiplications of the ciphertext and the plaintext, in the classification decision, the internal nodes are ciphertext, and the leaf nodes are plaintext. Finally, considering the number of ciphertext multiplications, the ciphertext result is multiplied by the left-tree ciphertext, and a total of two multiplications occur. The number of leaf nodes is the same as the number of multiplications and is a multiple of the number of internal nodes. The specific multiple is the number of cross-classifications [36]. The approximate ciphertext calculation model is constructed as follows:

$$\sigma(x, \lambda) = \frac{1}{1 + e^{-\lambda x}} \tag{7}$$

In formula (7), $\sigma(x, \lambda)$ represents the approximate ciphertext; x represents transaction data; λ represents steepness factor; and e is the natural constant. In this paper, $\sigma(x, \lambda)$ is transformed into polynomial form by approximate substitution, and its process can be expressed as follows:

$$\sigma'(x, \lambda) = \cos(\delta \arccos x) \tag{8}$$

where $\sigma'(x, \lambda)$ represents the polynomial form, and δ represents the degree of Chebyshev polynomial. At this time, the constructed polynomial is in an orthogonal form, which can realize the optimal uniform approximation to obtain the nearest similar solution. Algorithm 1 describes the overall design process.

Algorithm 1: Blockchain-based ciphertext approximate classification algorithm

Input: Unclassified ciphertext dataset $C_t = \{x|x_i \in I, i = 1, 2, \ldots, N\}$, the minimum unit block size N

Output: Ciphertext dataset classification prediction results $\widetilde{C}_t = \emptyset(C_t)$

```
initialize $f_0 = argmin_\gamma \sum_{i=1}^{N} S(y_i, \gamma)$
for $m = 1$ to $M$ do
    for $i = 1, 2, \ldots, N$ do
    calculate
        $\gamma_{im} = \left[\frac{\partial S(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$
fit a regression tree to the targets $\gamma_{im}$ giving terminal regions $R_{jm}, j = 1, 2, \ldots, J_m$
for $(j = 1, 2, \ldots, J_m)$ do
    calculate
        $\gamma_{im} = argmin_\gamma \sum_{x_i \in R_{jm}} S(y_i, f_{m-1}(x_i) + \gamma)$
update $f_m(x) \leftarrow f_{m-1}(x) + \sum_{j=1}^{Jm} \gamma_{jm} I(x \in R_{jm})$
  //Step 1. Generate Gradient Boosting Decision Tree Models $DT = \{\gamma_m, f(x_i)\}$
for $(j = 1, 2, \ldots, M)$ do
    if $(\xi \leq \vartheta)$ then
            if$(n == 0)$
                $k = 1;$
            else if$(n == 1)$
                $k = x;$
            else
                $k = 2 * x * T(n-1, x) - T(n-2, x);$
          return $k;$
update $\mathcal{F}_i \leftarrow Chebyshev(y_i, \xi, x_{i-1})$
  //Step 2. Convert the decision tree $DT_i$ to a Chebychev polynomial $\mathcal{F}(\vartheta, x)$
for $(m = 1$ to $N)$ do
    Calculate
        $C_t(\mathcal{F}_m) \leftarrow HE.Add(\widetilde{C}_t(\mathcal{F}_{m-1})) \oplus HE.Mult(\widetilde{C}_t(\mathcal{F}_{m-1}) : \gamma_{im})$
  //Step 3. Perform additive and multiplicative homomorphic multilevel operations on datasets
return $\widetilde{C}_t$
```

Based on this polynomial transformation, the approximate classification of the ciphertext is realized by the sigmoid function. On the user side, the server classifies the data using a preset polynomial and sends the results to the user. The user decrypts the data with the private key and obtains approximate classification results.

3.4 Multiple Data Privacy Protection Model

The multivariate data privacy protection model includes a permission setting module, a privacy data access module, a privacy data coloring module, and a data monitoring module. The overall framework of the multivariate data privacy protection model is shown in Fig. 4.

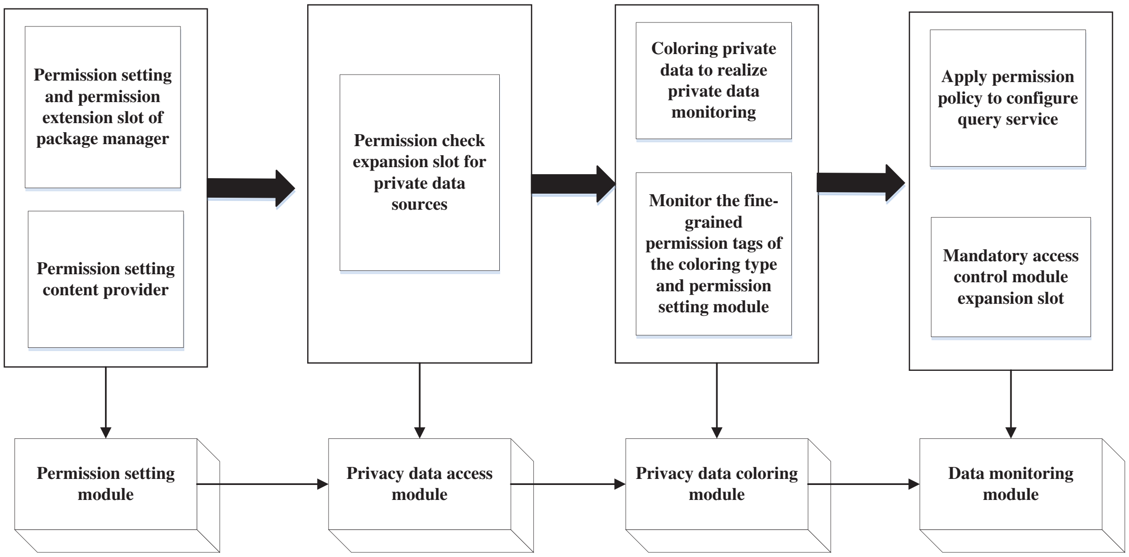
Permission setting and permission extension slot of package manager
Permission setting content provider
Permission check expansion slot for private data sources
Coloring private data to realize private data monitoring
Monitor the fine-grained permission tags of the coloring type and permission setting module
Apply permission policy to configure query service
Mandatory access control module expansion slot
Permission setting module
Privacy data access module
Privacy data coloring module
Data monitoring module

Figure 4: Overall framework of the multivariate data privacy protection model

As shown in Fig. 4, the permission setting module provides users with fine-grained permission settings for the application. These include the permission setting and the permission extension slot of the package manager and the permission setting content provider. The privacy data access module includes the permission check extension slot of the privacy data source, specifically referring to the expansion slot added in the privacy content provider and service module; the privacy data coloring module colorizes the privacy data at the middleware level to realize the privacy data monitoring, which includes monitoring the fine-grained permission labels of the coloring type and permission setting module. The data monitoring module includes the application permission policy configuration query service and the mandatory access control module expansion slot, which is triggered when data exchange occurs in the communication between programs.

Based on the approximate ciphertext classification, a multivariate data privacy protection model is established. The model can be expressed as:

$$w = (u, \vartheta, l, \beta, d, s, e) \tag{9}$$

In formula (9), w represents the privacy protection model; u represents the node-set; ϑ represents the mapping relationship between user and node; l represents the blockchain network; β represents the interactive operation of nodes; d represents multi-source data; s represents a collection of smart contracts; and e represents a collection of user requirements. The model uses the blockchain network based on a whitelist to complete node communication. Tuple data is stored in the block data structure in the form of transactions. The device information is saved in the whitelist in the form of a public key and data value. For tuple data privacy, it is deployed in the contract layer to ensure the security of sensitive information such as node location [37]. The received information and node information are stored in the block.

First, the hash value of the transaction is recursively calculated, and the root hash value is calculated using the dictionary structure to prevent data from being tampered with. Then, the secret key information is stored in the whitelist of the block body to verify the user's information. Because the blockchain nodes change dynamically, the interaction behavior is recorded in the block body structure, so that the descending level and compressed space are reduced. The device with minimal hardware configuration is set as a light node, which saves only the data of the block header, which can quickly verify the data. The location in the blockchain network is mapped into a tree structure, and its distance is obtained by XOR operation [38]. Through this calculation method, the real topology can avoid being exposed. Each node finds the nearest location and adds the corresponding whitelist information to its communication protocol. When a new node applies to join the blockchain, it needs to be guided by the original node to spread the information to the entire network. The original node adds the nearest location information to the new node, and the new node adds the returned information to its communication protocol [39]. Accordingly, other nodes complete the same data update operation. In this way, the same quasi-identifier of multi-source data is recorded in each node, and the attacker cannot obtain the user's privacy record through the connection record. Based on the above process, we designed the multi-source data privacy protection method.

The abovementioned analysis obtains the design process of the multiple data privacy protection model. The flowchart is shown in Fig. 5. It can be seen that the establishment of a blockchain data structure is the first step in the model. Because the transaction information is visible across the network, the data is easily mined and collected, and a data-sharing protocol needs to be designed. In the process of decrypting the plaintext, a random number is used to prove that the input and output data are equal, using the blockchain parameters to decrypt the random number. Adding homomorphic encryption to design an approximate ciphertext classification algorithm, based on the approximate ciphertext classification, a multivariate data privacy protection model is established.

4 Experimental Evaluation

We built a simulation experiment platform, created the initial settings, and analyzed the performance of the proposed method.

4.1 Establishing the Experimental Environment

To verify the performance of the proposed multi-source data privacy protection method based on homomorphic encryption and blockchain technology, we used the Fabric project to conduct experimental tests on the method. Based on building a fabric-based blockchain scenario, experiments were carried out on the application effects of the proposed method to verify its feasibility and efficiency in privacy protection. We used the Java programming language to create a homomorphic encryption scheme (HES) and used the JDK 14 software development kit. We used the Python 2.7 design language to build the blockchain network (BN). The details of the experimental settings are shown in Tables 1 and 2.

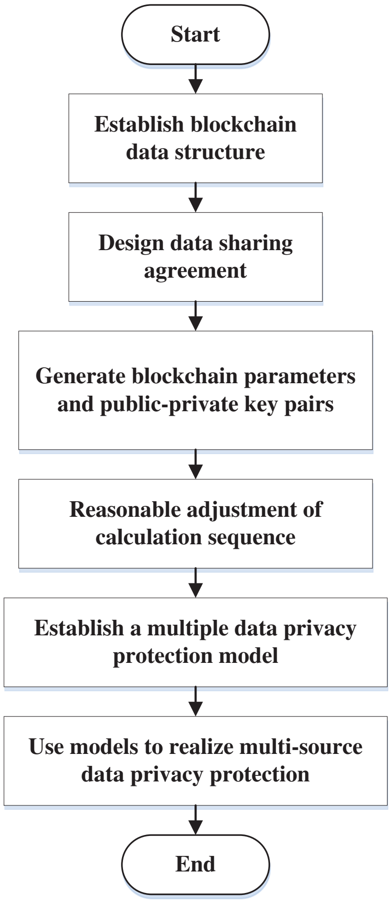
Start
Establish blockchain data structure
Design data sharing agreement
Generate blockchain parameters and public-private key pairs
Reasonable adjustment of calculation sequence
Establish a multiple data privacy protection model
Use models to realize multi-source data privacy protection
End

Figure 5: Design process of the multiple data privacy protection model

Table 1: Construction of the experimental platform

System support	Development tools	
Vmware 15.0.1 ubuntu18.04 (Intel® Xeon(R) CPU E5-2640 v4 @ 2.40 GHz × 4,4G RAM,64 bit)	ide	language
	visual studio code 1.60	JDK 14 for HES
		Python 2.7 for BN

Table 2: Construction of the blockchain network

Basic network support	Fabric 2.0
Membership structure	Org: Alice, Bob Peer nodes: cuc, agc Orderer node

(Continued)

Table 2 (continued)

Basic network support	Fabric 2.0
Services	client orderer.test.com cuc.Alice.test.com cuc.Bob.test.com agc.Alice.test.com agc.Bob.test.com chaincode (web application)

In this experiment, two organizations, **Alice** and **Bob**, are selected from the fabric network for basic networking. Each organization provides two-node maintenance networks, namely, **cuc** and **agc**. The network consists of an orderer node and peer nodes to form a *solo* consensus, and the certificates and keys distributed to the nodes are generated by the cryptogen utility. The application side accesses the blockchain service through the Go SDK and provides an external web interface for the client. To better test the adaptation effect of the data privacy protection method in the blockchain scenario, this experiment used a virtual machine to build a multinode environment in the VMware 15.0.1 mode to simulate the composition and release of multi-source data. In this experiment scenario, seven servers were set up, of which five constituted the peer operating environment, one the orderer environment, and the remaining one the web environment. In the fabric network, the docker pull command on the peer and orderer was used to pull the image and start the client container. After the image download was completed, the certificate and key of the blockchain network were generated, and the application configuration and transaction files were created. We wrote a file template to start the container on all nodes and started the container through the "compose up" subcommand. After setting the environmental variables, we used command operations to add the cuc node of Alice organization to the same channel and created an initial block file based on the configuration file. We copied the file to other peer nodes and joined the application channel in the same way. Based on this, the blockchain network was established. After the experimental environment was set up, the effect of the proposed data privacy protection method was tested, and its feasibility and efficiency were evaluated.

4.2 Experiment and Analysis of Results

Based on the construction of the experimental environment, using the transfer from user A to user B as an example, the secret key generation, encryption, and decryption processes in homomorphic encryption were tested. The application side needed to generate a variety of data, and the data were independent of each other, and in no chronological order. Therefore, multiple threads were used to generate and transmit data in parallel, and parallel verification was used to process data at the chaincode end, to shorten the running time of the application end and chaincode end. In the actual data environment, the efficiency of the privacy protection method was determined by the longest-running step, and the guarantee of security was based on the solution of a discrete logarithm. As long as the length of the secret key reached a certain length, the security of the method could be guaranteed. Generally, when the length of the secret key was 3072 bits, the security of the method was guaranteed. Therefore, the length of the secret key set in this experiment was less than 3072 bits. Considering the characteristics of large volume, multi-source heterogeneity, and the strong mobility of converged media data, we used the differential privacy scheme based on the exponential mechanism as the baseline and set privacy budget $\varepsilon = 1$. We compared the application effect of our method with the data privacy protection methods based on *k*-anonymity technology and differential privacy technology. *K*-anonymity technology replaces the quasi-identification column of data with general data with consistent semantics so that the attacker cannot distinguish the specific attributes of sensitive information. Differential privacy technology adds or reduces records in the statistical database to form two datasets and reduces the risk of privacy disclosure by querying the datasets. In the actual financial transactions, 64-bit integers were used to test the running time of our proposed privacy protection method. In the same experimental environment, the abovementioned two comparison methods and this method were used to test the privacy protection of transfer transactions. Each performance test was performed 100 times, and then they were averaged. The transaction data results are shown in Tables 3–5.

Table 3: Experimental results of key generation time (ms)

Key length (bits)	The method of this paper	The method based on *k*-anonymity technology	The method based on differential privacy technology
64	**3.1**	32.1	6.2
128	**3.8**	45.3	9.6
256	**4.3**	61.3	11.9
512	**6.0**	92.1	16.3
1024	**6.4**	109.3	23.4
2048	**7.1**	152.3	28.1
3072	**7.9**	183.3	34.2

Table 4: Experimental results of encryption time (ms)

Key length (bits)	The method of this paper	The method based on *k*-anonymity technology	The method based on differential privacy technology
64	**16.2**	110.7	78.4
128	**30.3**	214.6	112.9
256	**53.5**	418.3	216.2
512	**107.2**	725.7	323.8
1024	**186.4**	1245.2	640.6
2048	**310.9**	2155.1	1148.3
3072	**549.8**	2461.2	1442.9

Table 5: Experimental results of decryption time (ms)

Key length (bits)	The method of this paper	The method based on *k*-anonymity technology	The method based on differential privacy technology
64	**9.8**	94.3	43.8
128	**14.6**	126.1	85.6

(Continued)

Table 5 (continued)

Key length (bits)	The method of this paper	The method based on k-anonymity technology	The method based on differential privacy technology
256	**25.7**	219.2	138.7
512	**46.4**	433.8	212.1
1024	**90.2**	833.6	430.6
2048	**143.5**	1588.5	872.4
3072	**255.4**	2152.4	1410.7

According to the experimental results in Tables 3–5, with the increase in the length of the secret key, the secret key generation, encryption, and decryption times of all three methods increase, and the encryption time is greater than the decryption time. This shows that decryption is more sensitive to the change of secret key length. When the length of the secret key does not reach 1024 bits, the growth rate of encryption and decryption time is not obvious, and the time consumed for encryption and decryption for a single method is close under different key strengths. When 1024 bits are reached, the growth rate increases significantly, and it takes more time to implement encryption and decryption. When the key length is the same, the key generation, encryption, and decryption times of the proposed method are less than those of the methods based on k-anonymity technology and differential privacy technology. Using the key length of 2048 bits as an example, the encryption time of the proposed method is 310.9 ms, which is 1844.2 and 837.4 ms shorter, respectively, than the method based on k-anonymity technology and differential privacy technology. The key generation time of the proposed method is 7.1 ms, which is 145.2 and 21 ms shorter, respectively, than the method based on k-anonymity technology and differential privacy technology. The decryption time of the method in this paper is 143.5 ms, which is 1445 and 728.9 ms shorter, respectively, than the methods based on k-anonymity technology and differential privacy technology.

Obviously, due to the huge volume of media data, the privacy protection effect of k-anonymity is significantly worse than the other two methods. During data processing, a minimum of k records is required for the same quasi-identifier, which requires high database quality and increases the computational burden. Differential privacy protects privacy by adding noise. It needs to operate on a single data point in the database. As the amount of data increases, the computational burden of noise processing also increases. Homomorphic encryption processes all data uniformly, reducing the intermediate processing of a single data point, but the cryptographic operation process is complicated. In a simulation environment with relatively sufficient computing power, its privacy protection efficiency is still relatively high. Based on the above results, the method proposed in this paper can significantly reduce the processing time of multi-source data, creates little operating burden on the blockchain network, and consumes less time than the other methods. This means it is suitable for data privacy protection and has good potential for use in other applications.

Time complexity refers to the time it takes for an algorithm to run after it is written into an executable program. Three methods were used to test the running time of the program. The test results are shown in Fig. 6.

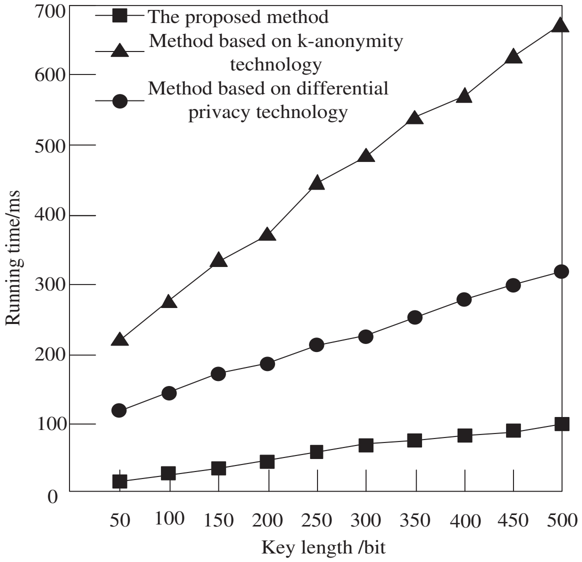

Figure 6: Comparison results of time complexity of the three methods

It can be seen from Fig. 6 that the running time of the three methods increases with the increase in key length. When the key length reaches 500 bits, the running time of our proposed method is about 100 ms, while the running time of the method based on k-anonymity is approximately 670 ms, and the running time based on the differential privacy technology method is almost 320 ms. These results prove that our method can significantly reduce the running time of multi-source data, creates little operating burden on the blockchain network, and is less time-consuming. Because this method uses the six-layer structure set up by the blockchain to establish the data structure, it ensures the integrity of the data, thereby greatly reducing the running time of multi-source data and reducing the complexity of ciphertext calculation.

5 Feasibility Analysis

We next analyzed the feasibility of applying the proposed method to multi-source data privacy protection. First, we conducted a demand analysis of media big data privacy protection. As part of the analysis, we examined the technical characteristics of blockchain technology.

5.1 Demand Analysis

The process of media convergence development is characterized by a large amount of data, many types of data, large differences between those types, and complex input and output operations. In recent years, the impact of media data leakage and invasions of user privacy has been increasing. For media data privacy protection, three requirements should be met:

- *An effective copyright management mechanism*: Digital media content spreads rapidly through the Internet. Much of it, including video, audio, IP creative, and other media big data, is facing copyright protection issues.
- *An environment conducive to media consumption*: Large user groups and the free release of information preclude the supervision of the rapid spread of media data, resulting in a large number of spam attacks.

- *A secure information sharing platform*: Because in the production of media data there are no restrictions put on anyone, the media database is flooded with all kinds of information. With users often unable to discern facts, they cannot know the authenticity and accuracy of the information itself. As a result, false information is widely spread, often in the form of rumors.

5.2 Description of Technical Characteristics

Blockchain technology can be used in many fields [40], because it has the following characteristics:

- *Decentralization*: In traditional transaction systems, there is a central authority that is generally trusted. Every transaction is verified by institutions, which, however, inevitably increases management costs and performance bottlenecks on central servers, as well as the risk of single points of failure and cyberattacks. In a blockchain network, all transactions can be verified between any two peer entities without the involvement of a central authority.
- *Immutability*: In the blockchain, the current block stores the hash value of the previous block in the form of a hash pointer to ensure connectivity between blocks. When any data on the chain is modified, the hash of the block will change and the system will not allow it. Every transaction generated needs to be confirmed and recorded in blocks backed up by the entire network, which is nearly impossible to tamper with, as all blocks are broadcast across the network and backed up by a single node. Additionally, each broadcast block will be verified by other nodes, confirming the authenticity and validity of the transaction. Therefore, any forgery is easy to spot.
- *Anonymity*: In a blockchain system, there is no need to trust data to interact with other users. Because the rules of the blockchain plan determine whether the activity is valid or not, parties do not have to disclose their identities. In addition, the blockchain provides each user with a secure key through cryptographic tools and interacts with the blockchain network through the generated addresses. Users can generate multiple different addresses to prevent revealing their true identity.
- *Auditability*: Because every transaction on the blockchain is verified and recorded using timestamps, users can easily verify and track previous records by accessing any node in the distributed network. Furthermore, each transaction in the blockchain can be iteratively traced back to previous transactions. This auditability improves the traceability and transparency of data stored in the blockchain.

There is no centralized server in the blockchain, and interactions are completed by peer nodes, which can eliminate the risks of a centralized server. In the blockchain system, the multi-party distributed accounting model is adopted to ensure that the data is visible and consistent to all participants, achieving multi-party sharing of data and solving the problem of information asymmetry. Through the use of chain storage and a node consensus mechanism to realize data confirmation and authorization, data sharing and circulation are promoted. The use of encryption and decryption authorization, zero-knowledge proof, and other cryptographic technologies enables the protection of data privacy. At the same time, users participate in network affairs anonymously to ensure that their identity is not leaked. Given the three main points of current research on privacy protection issues, the method proposed in this paper provides a concept that combines blockchain and homomorphic encryption.

6 Conclusion

In this paper, we use blockchain technology and homomorphic encryption technology to study the privacy protection of multi-source data in the media field, and propose a multi-source data privacy protection method. The experimental results show that the method can shorten the time required for key generation, encryption and decryption, and has high time efficiency, which can meet the needs of practical applications. However, the proposed method has some disadvantages. Due to the limited experimental environment, a virtual machine is used to simulate the blockchain environment. Some research results have been achieved under experimental theoretical conditions, but this is far from real-world industrial applications. Subsequent research should consider the effect of protecting data privacy under real blockchain conditions. In the future, we will focus on improving the efficiency of data sharing protocols and optimizing the generalization ability of ciphertext approximate clustering algorithms on the Media Alliance blockchain platform. In addition, in terms of ensuring data compatibility, we can consider establishing data standards in combination with current business processes to ensure efficient data flow between the initiator and the receiver.

Funding Statement: This work was funded by the High-Quality and Cutting-Edge Discipline Construction Project for Universities in Beijing (Internet Information, Communication University of China).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Fang, C., Guo, Y., Wang, N., Zhen, S., Tang, G. (2020). Differential private data publishing method based on generative adversarial network. *Acta Electronica Sinica, 48(10),* 1983–1992. DOI 10.3969/j.issn.0372-2112.2020.10.016.
2. Sun, G., Chang, V., Ramachandran, M., Sun, Z., Li, G. et al. (2017). Efficient location privacy algorithm for Internet of Things (IoT) services and applications. *Journal of Network and Computer Applications, 89,* 3–13. DOI 10.1016/j.jnca.2016.10.011.
3. Yang, D., Qu, B., Cudré-Mauroux, P. (2019). Privacy-preserving social media data publishing for personalized ranking-based recommendation. *IEEE Transactions on Knowledge and Data Engineering, 31(3),* 507–520. DOI 10.1109/TKDE.2018.2840974.
4. Li, F., Yang, J., Xue, L., Sun, D. (2018). Real-time trajectory data publishing method with differential privacy. *The 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, pp. 177–182. Shenyang, China. DOI 10.1109/MSN.2018.00029.
5. Yang, X., Chen, A., Wang, Z., Li, S. (2022). Cloud storage data access control scheme based on blockchain and attribute-based encryption. *Security and Communication Networks*, pp. 1939–0114. Hindawi, New York, USA. DOI 10.1155/2022/2204832.
6. Zhu, S., Hu, H., Li, Y., Li, W. (2019). Hybrid blockchain design for privacy preserving crowdsourcing platform. *IEEE International Conference on Blockchain (Blockchain)*, pp. 26–33. Atlanta, GA, USA. DOI 10.1109/Blockchain.2019.00013.
7. Fu, Y., Qin, Y., Shen, G. (2019). Multi-source data privacy protection based on transfer learning. *Computer Engineering and Science, 41(4),* 641–648.
8. Pandiaraja, P., Deepa, N. (2019). A novel data privacy-preserving protocol for multi-data users by using genetic algorithm. *Soft Computing, 23(18),* 8539–8553. DOI 10.1007/s00500-019-04239-1.
9. Li, M. (2019). Multi-source network data privacy protection simulation based on private blockchain. *Computer Simulation, 36(8),* 266–270.

10. Kang, H., Deng, J. (2021). Survey on blockchain data privacy protection. *Journal of Shandong University (Natural Science), 56(5),* 92–110. DOI 10.6040/j.issn.1671-9352.0.2020.595.
11. Cheon, J. H., Kim, D., Kim, Y., Song, Y. (2018). Ensemble method for privacy-preserving logistic regression based on homomorphic encryption. *IEEE Access, 6,* 46938–46948. DOI 10.1109/ACCESS.2018.2866697.
12. Yang, S., Ahmed, H., Robert, C., Lara, D. (2020). Topology-aware cooperative data protection in blockchain-based decentralized storage networks. *IEEE International Symposium on Information Theory (ISIT)*, pp. 622–627. Los Angeles, CA, USA. DOI 10.1109/ISIT44484.2020.9174443.
13. Merve, C. K. K., Albert, L. (2018). A survey on anonymity and privacy in bitcoin-like digital cash systems. *IEEE Communications Surveys & Tutorials, 20(3),* 2543–2585. DOI 10.1109/COMST.2018.2818623.
14. Koshy, P., Koshy, D., Mcdaniel, P. (2014). An analysis of anonymity in bitcoin using P2P network traffic. *International Conference on Financial Cryptography and Data Security*, pp. 469–485. Berlin Heidelberg, Germany, Springer. DOI 10.1007/978-3-662-45472-5_30.
15. Li, S., Li, Y., Han, W., Du, X., Mohsen, G. et al. (2021). Malicious mining code detection based on ensemble learning in cloud computing environment. *Simulation Modelling Practice and Theory, 113(7),* 102391. DOI 10.1016/j.simpat.2021.102391.
16. Song, C., Zhang, Y., Peng, W., Yan, X. (2019). Research on *k*-anonymous privacy protection scheme based on bilinear pairing. *Application Research of Computers, 36(5),* 1529–1532. DOI 10.19682/j.cnki.1005-8885.2018.0021.
17. Feng, D., Zhang, M., Ye, Y. (2020). Research on differentially private trajectory data publishing. *Journal of Electronics & Information Technology, 42(1),* 74–88. DOI 10.11999/JEIT190632.
18. Duan, Y., Lu, Z., Zhou, Z., Wu, J. (2019). Data privacy protection for edge computing of smart city in a DIKW architecture. *Engineering Applications of Artificial Intelligence, 81(5),* 323–335. DOI 10.1016/j.engappai.2019.03.002.
19. Qiao, Y., Liu, Z., Lv, H., Li, M., Li, Z. et al. (2019). An effective data privacy protection algorithm based on differential privacy in edge computing. *IEEE Access, 7,* 136203–136213. DOI 10.1109/ACCESS.2019.2939015.
20. Sun, G., Xie, Y., Liao, D., Yu, F., Chang, V. (2017). User-defined privacy location-sharing system in mobile online social networks. *Journal of Network and Computer Applications, 86(5),* 34–45. DOI 10.1016/j.jnca.2016.11.024.
21. Luo, C., Chen, X., Xu, J., Zhang, S. (2021). Research on privacy protection of multi source data based on improved GBDT federated ensemble method with different metrics. *Physical Communication, 49(2),* 101347. DOI 10.1016/j.phycom.2021.101347.
22. Qiao, K., Tang, H., You, W., Zhao, Y. (2019). Blockchain privacy protection scheme based on aggregate signature. *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pp. 492–497. Chengdu, China. DOI 10.1109/ICCCBDA.2019.8725693.
23. Li, X., Mei, Y., Gong, J., Xiang, F., Sun, Z. (2020). A blockchain privacy protection scheme based on ring signature. *IEEE Access, 8,* 76765–76772. DOI 10.1109/ACCESS.2020.2987831.
24. Li, P., Xu, H. (2020). Blockchain user anonymity and traceability technology. *Journal of Electronics & Information Technology, 42(5),* 1061–1067. DOI 10.11999/JEIT190813.
25. Hassan, M. U., Rehmani, M. H., Chen, J. J. (2020). Differential privacy in blockchain technology: A futuristic approach. *Journal of Parallel and Distributed Computing, 145(3),* 50–74.
26. Liu, J., Fan, K., Li, H., Yang, Y. (2021). A blockchain-based privacy preservation scheme in multimedia network. *Multimedia Tools and Applications, 80(2),* 1–15. DOI 10.1007/s11042-021-10513-y.
27. Yang, Y., Zhang, J., Ma, J. (2021). Method for using the blockchain to protect data privacy of IoV. *Journal of Xidian University (Natural Science), 48(3),* 21–30. DOI 10.19665/j.issn1001-2400.2021.03.003.
28. Chen, B., Xu, D., Yang, X., Sun, C. (2020). Modeling of rice supply chain traceability information protection based on blockchain. *Transactions of the Chinese Society for Agricultural Machinery, 51(8),* 328–335.

29. Kurri, V., Raja, V., Prakasam, P. (2021). Cellular traffic prediction on blockchain-based mobile networks using LSTM model in 4G LTE network. *Peer-to-Peer Networking and Applications, 14(3),* 1088–1105. DOI 10.1007/s12083-021-01085-7.

30. Mkrttchian, V. (2021). Avatars-based decision support system using blockchain and knowledge sharing for processes simulation: A natural intelligence implementation of the multi-chain open source platform. *International Journal of Knowledge Management, 17(1),* 72–92. DOI 10.4018/IJKM.2021010105.

31. Kibiwott, K. P., Zhang, F., Kimeli, V. K., Opoku-Mensah, E. (2019). Privacy preservation for ehealth big data in cloud accessed using resource-constrained devices: Survey. *International Journal of Network Security, 21(2),* 312–325. DOI 10.6633/IJNS.201903_21(2).16.

32. Benifa, J., Mini, G. V. (2020). Privacy based data publishing model for cloud computing environment. *Wireless Personal Communications, 113(4),* 2215–2241. DOI 10.1007/s11277-020-07320-3.

33. Saranya, K., Premalatha, K. (2020). Privacy-preserving data publishing based on sanitized probability matrix using transactional graph for improving the security in medical environment. *The Journal of Supercomputing, 76(8),* 5971–5980. DOI 10.1007/s11227-019-03102-2.

34. Kumar, M. M., Prasad, M., Raju, U. S. N. (2020). BMIAE: Blockchain-based multi-instance iris authentication using additive elgamal homomorphic encryption. *IET Biometrics, 9(4),* 165–177. DOI 10.1049/iet-bmt.2019.0169.

35. Jia, C., Li, R., Wang, Y. (2021). Privacy protection scheme of DBSCAN clustering based on homomorphic encryption. *Journal on Communications, 42(2),* 1–11. DOI 10.11959/j.issn.1000-436x.2021026.

36. Liu, J., Chen, F., Xu, C., Guo, H., Li, T. (2019). Full-domain anonymization algorithm based on fully homomorphic encryption in the cloud. *Chinese Journal of Computers, 42(4),* 837–850.

37. Long, H., Zhang, S., Zhang, L. (2020). Data fusion method based on privacy preserving in crowd sensing network. *Computer Engineering and Design, 41(12),* 3346–3352.

38. Yang, Y., Cai, J., Zhang, X., Yuan, Z. (2019). Privacy preserving scheme in blockchain with provably secure based on SM9 algorithm. *Journal of Software, 30(6),* 1692–1704.

39. Zhan, J., Wang, Q., Ouyang, X. (2019). Source-location privacy protection routing protocol in wireless sensor networks by avoiding attackers. *Computer Engineering and Applications, 55(12),* 103–109.

40. Drescher, D. (2017). *Blockchain basics*, pp. 55–68. Berkeley, CA: Apress.