HWKA: A Novel Homomorphic Encryption and k-center Algorithm for Secure Storage of English Data

Haiying Liu

(Corresponding author: Haiying Liu)

School of Foreign Languages, Zhengzhou University of Science and Technology

Email: ldadahai@163.com

(Received Dec. 28, 2022; Revised and Accepted Apr. 16, 2023; First Online Apr. 30, 2023)

The Special Issue on Computational Intelligence Networks for Privacy and Security in Evolving Internet of Multimedia Things

Special Editor: Prof. Shoulin Yin (Harbin Institute of Technology)

Abstract

Aiming at the security problem of English private data in a cloud environment, this paper puts forward a novel homomorphic encryption and k-center algorithm for the secure storage of English data. Firstly, we cluster the English data. The clustering method is improved by using the clustering process under the condition of changing the vertical data distribution. Secondly, Paillier homomorphic encryption is introduced in clustering. Then, this method is applied to the k-center clustering algorithm, which further guarantees data security. The experimental results show that the efficiency of the ciphertext computing model is greatly improved. Moreover, the model puts a lot of computing in the cloud environment, which can reduce the client's pressure and fully use the resources in the cloud environment.

Keywords: Clustering; Data Secure Storage; Homomorphic Encryption; k-center Algorithm

1 Introduction

At present, more and more enterprises and individuals begin to use cloud storage to store data and use cloud computing to process data [7, 9, 10, 16]. Cloud computing is easy to expand, it has low requirements on devices, and enhanced computing power, which improves resource utilization and reduces costs [5,11]. Moreover, the large-scale cluster and huge computing power of cloud environment enable cloud computing to process big data, carry out data mining on big data, and realize classification, clustering and image recognition through machine learning algorithm [2].

Generally, the structure of machine learning for privacy protection can be divided into two categories: privacy training and privacy classification. Existing studies focus on the first type, which not only protects the privacy of the training samples provided by the data provider, but also protects the classifier parameters of the evaluator and the prediction results of the client [20], that is, only the client can obtain the prediction results.

There are many algorithms for privacy data protection, such as naive Bayes, decision trees, linear discriminant classifiers and more general kernel methods. Reference [21] proposed a back-propagation neural network training algorithm that was suitable for randomly segmenting training data sets and protecting privacy. Reference [18] proposed to use a single homomorphic encryption scheme to train multiple machine learning classifiers. Reference [8] used parallel deep learning to design, implement and evaluate a deep learning for privacy protection. In the above studies, each participant trained the local data set with the same neural network model and used the sharing of the model's selective parameters as a technique to benefit from other participants' models without explicitly sharing training inputs. But this approach took up storage space for the participants who keep sensitive data [17]. At the same time, each participant had to go through some calculations to train.

With the deepening of research, homomorphic encryption algorithms begin to improve the resistance against attack algorithms. Indiscriminability (IND) and nonmalleability [15] are the most important things for encryption schemes because they are contradictory. In this paper, the definition of security objective of involved encryption scheme refers to indiscriminability. For all homomorphic encryption schemes, it has homomorphic properties that make it impossible for any of the all homomorphic encryption schemes to fulfill the security requirements of IND-CCA2. In the existing researches, the security of all homomorphic encryption is mostly considered from IND-CPA. In 2010, Loftus et al. implemented the first fully homomorphic encryption scheme with IND-CCA1 security [12]. Akleylek et al. [1] used "Modified Key" and

"Modified Decryption" to attack the homomorphic encryption scheme, and completed the attack on Gentry's encryption scheme. Hu's scheme could decrypt ciphertexts in specific subsets in Gentry's ciphertext space.

For the homomorphic encryption scheme, because it can realize homomorphic calculation of ciphertext, the attacker can access the decryption predictor after the ciphertext is given. Chen et al. [4] proposed a feedback attack algorithm to solve this problem. In their research, they assumed that the attacker was a cloud server, which expected to obtain users' private data while performing operations. If the plaintext space was set as $(0, 1)$, the attacker would add their selected ciphertext when returning the user ciphertext. After the user decrypted and calculated, the attacker would complete the feedback attack based on the observed result [14].

How to protect the privacy and security of English data while maintaining better data mining has become an important challenge. In this paper, the clustering method is improved by changing the clustering process under the condition of vertical distribution of data, and homomorphic encryption technology is introduced in the clustering. Then this method is used in the k-center clustering algorithm, so that the security of data is further guaranteed.

2 Preliminaries

2.1 k-center Algorithm

The description of cluster analysis is as follows. Set a data set that needs cluster analysis as $S = (S_1, S_2, \cdots, S_n)$. In this data set, each sample S is composed of its characteristic data into a vector with m dimensions $(S_{i1}, S_{i2}, \cdots, S_{im})$. When clustering S, A_i is one of the clusters, and all clusters satisfy the condition $U_{i=1}^{t} A_t = S$, and $A_i \cap A_j = \phi(i \neq j)$. Cluster analysis can be divided into static clustering and dynamic clustering. Static clustering refers to the fact that the number of clusters has been determined before the start of clustering, time t is a definite value. In dynamic clustering, the number of clusters is not determined in advance, but it is based on the actual situation of the sample data set. In the era of data information explosion, cluster analysis deals with a huge amount of data, and has a variety of forms. The traditional cluster analysis technology has been greatly challenged. These problems require cluster analysis to have new characteristics, that is, efficient scalability; It can process different attribute types in the data set, and find different clusters in the data set. The clustering is accurate and the data noise can be processed correctly. It performs well in high dimensional data sets. The clustering results have high reliability and so on.

The main steps of partitioning method are as follows. Firstly, it sets the partition k to be established. Partitions are created first with an initial partition and then using iterative relocation techniques to move objects from one group to another. To determine whether the result of partition is good or bad, the correlation degree of the objects in the cluster should be as high as possible and the difference between the clusters should be as large as possible. The traditional partition method can be extended to the subspace clustering, without traversing all the data space, reducing the amount of computation. In practical clustering applications, the most commonly used heuristic methods, such as k-mean method and k-center point algorithm, can gradually improve the quality of clustering to approximate the local optimal solution.

2.2 Paillier Homomorphic Encryption

Taking two large prime numbers p and q, set $n = pq$, and obtain $\varphi(p, q) = (p-1)(q-1)$. λ is defined by Carmichael function as the least common multiple of $(p-1)$ and $(q-1)$, that is, $\lambda = lcm(p-1, q-1)$, lcm means to take the least common multiple.

According to the definition of the $n-th$-order residual class puzzle, if an integer x is called the $n-th$-order residual class of module n^2, then there is an integer $y \in Z_{n^2}^*$, which makes $x = y^n mod n^2$ valid. Where mod is the modulus value. $Z_{n^2}^* = Z_n \times Z_n^*$. Z_n is the set of all non-negative integers. Z_n^* is the set of all numbers in set Z_n that satisfy $gcd(x, n) = 1$. gcd means to take the greatest common divisor.

Paillier homomorphic encryption [19] includes key generation, encryption and decryption.

1) Key generation. Randomly select two large prime numbers p, q and integer $y \in Z_{n^2}^*$, compute $n = pq$ and $\lambda = lcm(p-1, q-1)$, make $gcd[L(y^\lambda mod n^2), n] = 1$, the public key is (n, g), and the private key is λ. After the key is generated, the random number $r \in Z_n$ is selected to encrypt the data, and the ciphertext c is obtained, while m is the encrypted information. The calculation is shown in Equation (1).

$$c = E(m, r) = y^m r^n mod n^2. \tag{1}$$

2) Encryption. Based on the theory of compound residual hypothesis, the inverse operation of Equation (1) in the definition domain can be obtained by calculating λ from p and q. That is, for ciphertext c, the plaintext m can be obtained after being processed by Equation (2), where $L(i) = (i-1)/n$.

$$m = D(c, \lambda) = \frac{L(c^\lambda mod n^2)}{L(y^\lambda mod n^2)} mod n. \tag{2}$$

3) Decryption. According to the addition homomorphism property of Paillier encryption algorithm, for the ciphertext $E(x)$ and $E(t)$ of data x and t, the relation shown in Equation (3) is satisfied.

$$\begin{aligned} D[E(x) \oplus E(t) mod n^2] &= D[(y^x r_1^n) \oplus (y^t r_2^n) E(t) mod n^2] \\ &= D[y^{x+t} (r_1 r_2)^n mod n^2] \\ &= D[E(x+t) mod n^2] \end{aligned} \tag{3}$$

By processing $E(x)$ and $E(t)$, $E(x+t)$ can be obtained, and the specific value of data $x+t$ also can be obtained. This principle can be extended to the case of multiple groups of data. The properties of Paillier algorithm provide flexible ideas for tamper-proof and security protection of data encryption.

3 The Proposed Ciphertext Encryption Algorithm

3.1 Encryption Algorithm

Secret key generation (Keygen). First, it chooses two strong prime numbers p and q. And with the properties of strong prime numbers, we can get two more prime numbers p' and q', where $p'=(p-1)/2$, $q'=(q-1)/2$. And then, assume $N=pq$, $\lambda=lcm(p-1,q-1)/2$. It selects a generation factor $g\in Z^*_{N^2}$ (where $Z^*_{N^2}$ is a set of non-zero integers less than N^2, $g=N+1$) to obtain the public key $p_k=N$ and the private key $s_k=\lambda$.

Encryption (Enc). Given a plaintext $m\in Z_N$ and a random number $r\in Z_N$, the ciphertext can be expressed as:

$$[m]=g^m r^N modN^2=(1+mN)r^N modN^2.$$

Decryption (Dec). The private key is required to decrypt the ciphertext. First, it calculates:

$$[m]^\lambda modN^2=(1+mN)^\lambda r^{\lambda N} modN^2=1+m\lambda N.$$

Since $gcd(\lambda,N)=1$, it can obtain the plaintext:

$$m=L([m]^\lambda modN^2)\lambda^{-1}modN.$$

Where $L(x)=(x-1)/N$, λ^{-1} satisfies $\lambda^{-1}\lambda\equiv 1modN$, and then it uses the residual theorem to find the value of λ^{-1}.

Key decomposition. It selects a parameter δ so that $\delta\equiv 0mod\lambda$ and $\delta\equiv 1modN^2$. Define a polynomial,

$$q(x)=\delta+\sum_{i=1}^{k-1}\beta_i x^i.$$

Where β_i is any number in $Z^*_{\lambda N^2}$, where $Z^*_{\lambda N^2}$ is a set of non-zero integers less than λN^2. Let $\alpha_1,\alpha_2,\cdots,\alpha_n\in Z^*_{\lambda N^2}$ be n different non-zero numbers. Set $s_k^{(i)}=q(\alpha_i)$ as part of the secret key and send it to part i.

Partial decryption (PDec). After receiving the ciphertext $[m]$, partial secret key $s_k^{(i)}=q(\alpha_i)$ is used to partially decrypt the ciphertext, and partial plaintext $T^{(i)}$ is obtained, namely,

$$T^i=[m]^{q(\alpha_i)}modN^2.$$

Merge decryption (TDec). Once $d(d\geq k)$ partially decrypted ciphertexts are received, let $S=T^{\tau_1},T^{\tau_2},\cdots,T^{\tau_d}$, the algorithm can select any k ciphertexts in the set S to decrypt it.

$$T''=\prod_{l\in S}(T^{(l)})^{\Delta l,S(0)}modN^2.$$

Where $\Delta l,S(x)=\prod_{j\in S,j\neq l}\frac{x-\alpha_j}{\alpha_l-\alpha_j}$. So the plaintext m can be obtained by the following formula, namely,

$$m=L(T'').$$

Ciphertext refresh (CR). Once the ciphertext $[m]$ is received, CR algorithm can update the ciphertext without changing the plaintext. It selects a random number $r'\in Z_N$, and calculates,

$$[m]'=[m]r'^N=(rr')^N(1+mN)modN^2.$$

In addition, given $m\in Z_N$, there is

$$[m]^{N-1}=(1+(N-1)mN)r^{(N-1)N}modN^2=[-m].$$

3.2 Ciphertext Computing Algorithm

In order to ensure that the data can be calculated in the case of encryption, and to operate the encrypted data in the case of protecting the private data and the secret key from being leaked, the following three ciphertext computing algorithms are proposed.

A. Ciphertext multiplication Algorithm (CTMA)

CTMA computes $[xy]$ securely when cloud storage provides two encrypted data $[x]$ and $[y]$ as inputs.

Step 1. Cloud storage selects two random numbers r_x, $r_y\in Z_N$, it calculates:

$$\begin{aligned}X&=[r_x][x]=[x+r_x].\\Y&=[r_y][x]=[y+r_y].\\X_1&=P_{s_k^{(1)}}(X),Y_1=P_{s_k^{(1)}}(Y).\end{aligned}$$

$P_{s_k^{(1)}}$ is the partial decryption (PDec) and, it sends X, Y, X_1 and Y_1 to the cloud computing center.

Step 2. The cloud computing center receives X, Y, X_1 and Y_1 and calculates:

$$T_x^{(i)}=P_{s_k^{(i)}}(X),T_y^{(i)}=P_{s_k^{(i)}}(Y).$$

Cloud computing center uses TDec to decrypt X and Y, get $x'=x+r_x$ and $y'=y+r_y$, then calculates $h=x'y'$. It uses p_k to encrypt h, that is, $H=[h]$, and sends H to cloud storage.

Step 3. Once H is received, cloud storage computes:

$$\begin{aligned}S_1&=[x]^{N-r_y}=[-r_y x].\\S_2&=[y]^{N-r_x}=[-r_x y].\\S_3&=[r_y r_x]^{N-1}=[-r_y r_x].\end{aligned}$$

Then, cloud storage computes $HS_1S_2S_3 = [(x+r_x)(y+r_y)-r_yx-r_xy-r_yr_x] = [x,y]$. Therefore, cloud storage and cloud computing centers can jointly compute $[xy]$.

B. Ciphertext comparison algorithm (CTCA)

Given two encrypted numbers $[x]$ and $[y]$, CTCA can be used to determine the relationship between the plaintexts of two encrypted data (i.e. $x > y$, $x < y$ or $x = y$).

Step 1. It selects two random numbers r, $l \in Z_N$ and calculates:

$$\begin{aligned} E &= [x]^r[y]^{N-r}[l] = [r(x-y)+l] \\ E_1 &= P_{s_k^{(1)}}(E). \end{aligned}$$

And it sends E, E_1 and $[l]$ to the cloud computing center.

Step 2. The cloud computing center receives E, E_1 and $[l]$ and computes:

$$T_e^{(i)} = P_{s_k^{(i)}}(E).$$

And it uses TDec to decrypt E to get $e = r(x-y)+l$. Then the cloud computing center compares e and l. If $e > l$, it denotes $x > y$, then 1 is sent to cloud storage; If $e = l$, it denotes $x = y$, then 0 is sent to the cloud storage. Otherwise $x < y$, -1 is sent to the cloud storage system.

Step 3. The cloud storage system receives results from the cloud computing system. If 1 is received, it means $x > y$; If 0 is received, then $x = y$; If -1 is received, then $x < y$.

C. Ciphertext logarithm algorithm (CTLA)

Given an encrypted number $[x]$, CTLA will calculate $[lnx]$ securely.

Step 1. It selects a random number $r_x \in Z_N$, and calculates $X = [x]^{r_x} = [xr_x]$, $R = [Inr_x]$. Since the cloud storage knows $s_k^{(1)}$, it can calculate:

$$T^{(1)} = P_{s_k^{(l)}}(X).$$

It sends X, R, and $T(l)$ to the cloud computing center.

Step 2. The cloud computing center receives X, R, and $T(l)$. It computes:

$$T^{(i)} = P_{s_k^{(i)}}(X).$$

Then it decrypts X with TDec and gets $x' = xr_x$ and then calculates:

$$L = [lnx'] = [lnx + lnr_x].$$

The cloud computing center sends L to the cloud storage.

Step 3. The cloud storage system receives L and calculates:

$$\begin{aligned} LR^{N-1} &= [lnx + lnr_x][lnr_x]^{N-1} \\ &= [ln(x)] \end{aligned}$$

So it can obtain $[lnx]$.

4 Experiment and Performance Analysis

In order to verify the reliability of this new scheme (HWKA) for English data encryption, HWKA and other similar algorithms are tested under the same conditions in the simulation environment, and the results are analyzed and compared. The experimental environment is shown in Table 1.

Table 1: Experiment environment

Name	Statement
Operating system	Win11
RAM	32GB
CPU	Interl(R) Core TMi6

In the experimental environment of Table 1, the HWKA in this paper is tested and verified, and it is compared with FHE [13], MKHE [3] and LPPA [6] under the same conditions. The time of encryption and decryption and the accuracy of data interaction are analyzed and compared.

In the process of English data encryption and decryption by Paillier algorithm, the size of n in the public key determines the complexity of the key, and also indirectly determines the encryption and decryption time. In this paper, keys with n values of 32 bit, 64 bit, 128 bit, 256 bit, 512 bit, 1024 bite and 2048 bit are generated. Multiple encryption and decryption experiments are conducted on randomly generated data of the same length, and the results are sorted and analyzed. The relationships between the execution time of encryption, secondary encryption and decryption and the key length of Paillier homomorphic encryption algorithm are obtained, as shown in Table 2 and Figure 1. The decryption time includes the decryption of the "secondary encryption" ciphertext and the decryption of the transaction data.

As can be seen from Figure 1, when the key length is below 1024 bit, the encryption and decryption time of Paillier homomorphic encryption algorithm increases slightly but it has little change with the increase of the key length. When the key length exceeds 1024 bit, the key is quite complicated, and the encryption and decryption time spent also increases proportionately, and the requirements on hardware also increase.

On the premise of encrypting and decrypting English data of different data lengths, the encryption and decryp-

Table 2: Relation between execution time of HWKA and key length/t

key length	First encryption time	Secondary encryption time	Decryption time
0	12.1	12.1	14.6
32	14.7	15.4	15.8
64	16.9	17.2	25.2
128	18.4	18.6	28.7
256	23.5	23.9	35.4
512	51.6	53.4	62.8
1024	130.5	151.8	191.7

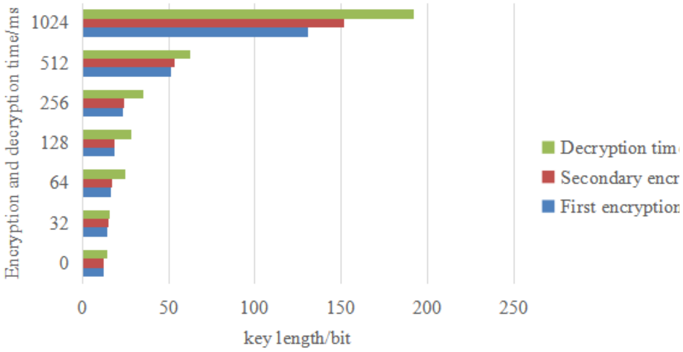

Figure 1: The visualization result of Table 2

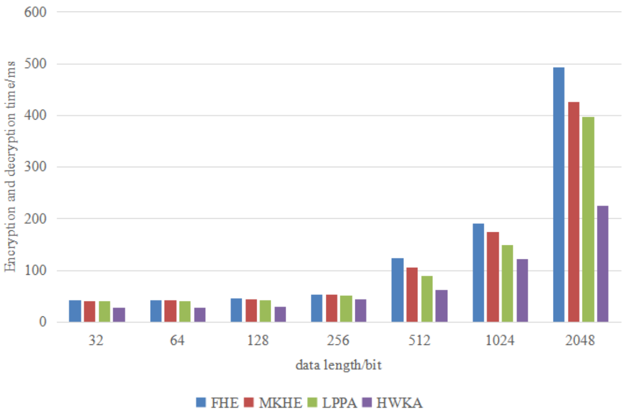

Figure 2: The visualization result of Table 3

tion execution time taken by the new method in this paper and other schemes is shown in Table 3 and Figure 2.

Table 3: Comparison of encryption and decryption execution time of different algorithms

data length	FHE	MKHE	LPPA	HWKA
32	41.6	40.7	39.8	27.6
64	42.7	41.8	40.9	28.3
128	44.9	43.7	42.6	29.8
256	53.8	52.4	52.1	44.7
512	123.7	105.2	89.7	61.8
1024	189.9	173.8	149.5	121.7
2048	492.7	426.5	397.2	224.5

Table 4: The accuracy of encryption and decryption with different algorithms/%

FHE	MKHE	LPPA	HWKA
90.8	92.7	95.3	98.6

As can be seen from Figure 2, compared with the encryption and decryption scheme of Paillier algorithm that only performs one-time encryption and decryption, the HWKA scheme slightly increases the execution time of encryption and decryption, but compared with other methods, the HWKA has certain advantages in the execution time of encryption and decryption process.

In addition to the efficiency of encryption and decryption, the success rate of English information encryption and decryption is also an important indicator to measure the data security scheme. The results are shown in Figure 3 and Table 4.

As can be seen from Figure 3, compared with other algorithms, under the same key length and the same encrypted information, the encryption and decryption accuracy rate of the proposed algorithm reaches 98.6%. This shows that the scheme has a positive effect on ensuring the reliability of data transmission in power trading process.

5 Conclusions

In this paper, we improve the clustering method by changing the clustering process under the condition of vertical distribution of data, and introduce homomorphic encryption technology in the clustering. Then this method is applied to the k-center clustering algorithm, which makes the security of data further guaranteed. In this paper, the complexity of communication and accuracy of the new algorithm are analyzed. In the last experiment, it is proved

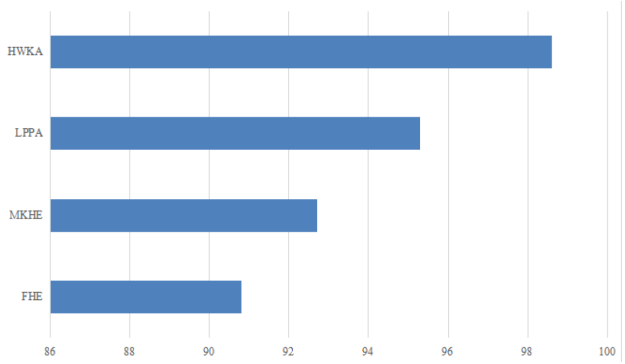

Figure 3: The accuracy of encryption and decryption with different algorithms

that the algorithm in this paper has better performance compared with other algorithms of the same type, and the communication encryption and decryption cost has been greatly reduced, which makes the computing load in distributed mode more balanced. On the other hand, this paper also studies and analyzes the privacy security of data mining, and proves the role of order preserving encryption and homomorphic encryption in the privacy security of clustering algorithm.

References

[1] S. Akleylek, K. Seyhan, "A probably secure bi-GISIS based modified AKE scheme with reusable keys," *IEEE Access*, vol. 8, pp. 26210-26222, 2020.

[2] R. Aversa, P. Coronica, C. De Nobili, *et al.*, "Deep learning, feature learning, and clustering analysis for sem image classification," *Data Intelligence*, vol. 2, no. 4, pp. 513-528, 2020.

[3] V. N. R. Bandaru, P. Visalakshi, "Block chain enabled auditing with optimal multi-key homomorphic encryption technique for public cloud computing environment," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 22, pp. e7128, 2022.

[4] Z. Chen, Z. Liu, L. Wang, "A modified model predictive control method for frequency regulation of microgrids under status feedback attacks and time-delay attacks," *International Journal of Electrical Power & Energy Systems*, vol. 137, pp. 107713, 2022.

[5] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments", *International Journal of Network Security*, vol. 16, no. 1, pp. 1-13, 2014.

[6] C. Guo, X. Jiang, K. K. R. Choo, *et al.* , "Lightweight privacy preserving data aggregation with batch verification for smart grid," *Future Generation Computer Systems*, vol. 112, pp. 512-523, 2020.

[7] M. S. Hwang, T. H. Sun, C. C. Lee, "Achieving dynamic data guarantee and data confidentiality of public auditing in cloud storage service," *Journal of Circuits, Systems, and Computers*, vol. 26, no. 5, 2017.

[8] H. Kim, S. H. Kim, J. Y. Hwang, *et al.*, "Efficient privacy-preserving machine learning for blockchain network," *IEEE Access*, vol. 7, pp. 136481-136495, 2019.

[9] C. W. Liu, W. F. Hsien, C. C. Yang, M. S. Hwang, "A survey of attribute-based access control with user revocation in cloud data storage", *International Journal of Network Security*, vol. 18, no. 5, pp. 900-916, 2016.

[10] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for shared data storage with user revocation in cloud computing", *International Journal of Network Security*, vol. 18, no. 4, pp. 650–666, 2016.

[11] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of attribute-based access control with user revocation in cloud data storage," *International Journal of Network Security*, vol. 18, no. 5, pp. 900–916, 2016.

[12] J. Loftus, A. May, N. P. Smart, *et al.*, "On cca-secure fully homomorphic encryption," *Cryptology ePrint Archive*, 2010.

[13] M. A. Mohammed, F. S. Abed, "Cloud storage protection scheme based on fully homomorphic encryption," *ARO-The Scientific Journal of Koya University*, vol. 8, no. 2, pp. 40-47, 2020.

[14] S. Nandi, S. Krishnaswamy, B. Zolfaghari, *et al.* , "Key-dependent feedback configuration matrix of primitive ρ–LFSR and resistance to some known plaintext attacks," *IEEE Access*, vol. 10, pp. 44840-44854, 2022.

[15] H. Orii, K. Hatano, H. Tanaka, *et al.*, "An image conversion method for color discriminability compensation of colorblindness using CycleGAN," *IEICE Proceedings Series*, vol. 69(RS2-6), 2022.

[16] A. K. Sandhu, "Big data with cloud computing: Discussions and challenges," *Big Data Mining and Analytics*, vol. 5, no. 1, pp. 32-40, 2021.

[17] X. Wang, S. Yin, M. Shafiq, *et al.*, "A new V-net convolutional neural network based on four-dimensional hyperchaotic system for medical image encryption," *Security and Communication Networks*, vol. 2022, pp. 1-14, 2022.

[18] A. Wood, K. Najarian, D. Kahrobaei, "Homomorphic encryption for machine learning in medicine and bioinformatics," *ACM Computing Surveys*, vol. 53, no. 4, pp. 1-35, 2020.

[19] S. Yin, H. Li, L. Teng, "A novel proxy re-encryption scheme based on identity property and stateless broadcast encryption under cloud environment," *International Journal of Network Security*, vol. 21, no. 5, pp. 797-803, 2019.

[20] M. Zhang, S. Huang, G. Shen, *et al.*, "PPNNP: A privacy-preserving neural network prediction with separated data providers using multi-client inner-product encryption," *Computer Standards & Interfaces*, vol. 84, pp. 103678, 2023.

[21] H. Zheng, Z. Gan, X. Li, *et al.*, "A green neural network with privacy preservation and interpretability," in *IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 606-614, 2022.

Biography

Haiying Liu biography. Haiying Liu is with School of Foreign Languages, Zhengzhou University of Science and Technology, born in Nanyang, Henan Province, 450064, China. Research interests: English data analysis.