Postprint of Deep Semantic Encrypted Search Based on Hybrid Cloud Architecture

Authors: Li Jian, robust

Date: 2022-05-18T16:08:25+00:00

Abstract

Traditional ciphertext retrieval schemes in cloud environments generate document vectors and query vectors based on statistical models, without considering the deep semantic information of documents and requests. This paper proposes a deep semantic ciphertext retrieval model based on a hybrid cloud architecture. Specifically, a vector generation model is constructed through a federated learning neural network model in the private cloud, while ciphertext data is stored in the public cloud. Additionally, this paper proposes an encrypted inverted index table to store document vectors, which improves retrieval efficiency during the search process in the public cloud while ensuring that retrieval information is not leaked. Analysis and experiments on real datasets demonstrate that our scheme outperforms current similar ciphertext retrieval schemes in terms of both security and search efficiency.

Full Text

Preamble

Vol. 39 No. 10
Application Research of Computers
ChinaXiv Partner Journal

Deep Semantic Ciphertext Retrieval Based on Hybrid Cloud Architecture

Li Jian, Jiao Jian
(School of Artificial Intelligence, Beijing University of Posts & Telecommunications, Beijing 100876, China)

Abstract: Traditional ciphertext retrieval schemes in cloud environments generate file vectors and retrieval vectors based on statistical models without considering the deep semantic information of files and requests. To address this limitation, we propose a deep semantic ciphertext retrieval model based on hybrid cloud architecture. The vector generation model is constructed through a private cloud federated learning neural network, while ciphertext data is stored in the public cloud. Additionally, we propose an encrypted inverted index table to store file vectors, which improves retrieval efficiency during public cloud search processes while ensuring that retrieval information remains confidential. Analysis and experiments on real datasets demonstrate that our proposed scheme outperforms existing ciphertext retrieval schemes of the same type in terms of both security and search efficiency.

Keywords: ciphertext retrieval; hybrid cloud; federated learning; encrypted inverted index table
Classification: TP309
doi: 10.19734/j.issn.1001-3695.2022.02.0101

0 Introduction

With the development of network technologies, expanding business demands, and the advancement of cloud computing and big data, an increasing number of organizations and companies are migrating data to public cloud servers to improve efficiency. However, data privacy remains a significant obstacle to the development of cloud computing applications. Although cloud service providers claim that mechanisms such as firewalls can enhance user data security, public cloud servers have complete control over outsourced data, and "honest-but-curious" cloud servers may leak sensitive data that data owners are unwilling to disclose. Consequently, data owners encrypt documents before uploading them to semi-honest clouds and store data as ciphertext to ensure document security. On the other hand, enterprises and governments that build private cloud servers to guarantee data security often face the problem of data silos, where large amounts of valuable internal resources cannot be utilized. Therefore, proposing an efficient ciphertext retrieval scheme that enhances performance through data sharing in cloud environments is of great significance.

Data encryption poses substantial challenges for data retrieval. In recent years, researchers have proposed numerous text retrieval strategies. In [1], Cao et al. first adopted the vector space model to compute the inner product of document vectors and query vectors, using the Secure kNN algorithm (Sec-kNN) for encryption. Based on inner product operations, they proposed a multi-keyword ciphertext retrieval result ranking (MRSE) scheme. Subsequently, many extended methods have been proposed [2~5]. While these methods offer provable security, they all employ traditional TF-IDF weighted statistical computation rules in the information retrieval domain. Keyword-based rules cannot effectively capture word context. Moreover, these schemes suffer from high vector dimensionality, high storage requirements, and high time complexity. In information retrieval, latent semantic models map queries to relevant documents when word matching fails, addressing language differences between documents and queries. Specifically, even if query keywords do not directly appear in a document, they can be constructed as low-dimensional semantic vectors with high similarity.

Semantic search represents an important research direction for information retrieval on both plaintext and encrypted data [6~11]. Semantic analysis eliminates language differences between queries and documents. Fu et al. [12] established a user interest model with the support of semantic ontology to achieve personalized keyword exact search. In [13,14], mutual information models were used to construct semantic expansion schemes. For example, Jadhav utilized mutual information models to expand query keywords and then calculated document relevance scores. Fuzzy keyword search technology was proposed in [15] to expand keyword sets. Fu et al. [15] developed a simplex-based hierarchical multi-keyword fuzzy search scheme without predefined fuzzy sets. Yang et al. [16] proposed utilizing EMD distance in a verifiable semantic scheme that describes the word transportation problem between queries and documents, where the minimum cost of word transportation is defined as the similarity between the query and each document.

This paper employs a hybrid cloud architecture to solve the data silo problem, uses federated learning to construct a deep neural network model for extracting deep semantics from data, and proposes an encrypted inverted index table structure to shorten retrieval time. This scheme ensures data security while achieving high retrieval accuracy and efficiency.

1.1 System Model

As shown in Figure 1, our scheme involves five entities: data owner, data user, private cloud server, public cloud server, and parameter server.

1) **Data Owner:** The data owner possesses valuable data. After encrypting the data, the data owner uploads ciphertext to the public cloud while uploading plaintext data to the private cloud for model training. Based on the trained neural network model, file vectors are generated to construct an encrypted inverted index table that is uploaded to the public cloud.

2) **Data User:** The data user sends search keywords and personal information to all data owners for authorization authentication, keyword mapping, and key acquisition. After sending search keywords to the parameter server and receiving the retrieval vector, the data user generates an encrypted trapdoor based on the key and sends it to the public cloud, receiving the TOP-K relevant file results returned by the public cloud.

3) **Private Cloud Server:** Each data owner has a private cloud server. The "honest and trustworthy" private cloud server receives plaintext data and network models for individual training, sharing training parameters

with the parameter server after each training round. The trained neural network model is transmitted back to the data owner.

4) **Public Cloud Server:** The public cloud server is an "honest but untrusted" entity that stores encrypted data and encrypted inverted index tables from data owners. Upon receiving the request trapdoor from the parameter server, it computes and identifies relevant encrypted files to send to data users.

5) **Parameter Server:** The parameter server is also a private cloud server that is "honest and trustworthy" and jointly maintained by all data owners. It serves as the central server for federated learning model training. After receiving search keywords from data users, it generates retrieval vectors and sends them to data users.

1.2 Threat Model

In our scheme, we assume the parameter server is trustworthy, while the public cloud is an "honest but curious" server [2]. Based on the information known by the semi-honest public cloud server, we investigate two threat models.

Known Ciphertext Threat Model: The public cloud only knows encrypted documents, encrypted data indexes, and secure query trapdoors. In this scenario, the public cloud server conducts attacks using ciphertext-only attack patterns.

Known Background Threat Model: The public cloud server should know more information than in the known ciphertext model. This knowledge includes correlations between trapdoors and statistical information related to the dataset. The public server uses known trapdoor information to attack.

1.3 Notations

- $Q = \{q_1, q_2, ..., q_k\}$: Search keyword set
- $D = \{D_1, D_2, ..., D_n\}$: Plaintext data set, where D_i represents the plaintext data set of the i-th data owner
- $E = \{E_1, E_2, ..., E_n\}$: Ciphertext data set, where E_i represents the ciphertext data set of the i-th data owner
- $P = \{P_{11}, P_{12}, ..., P_{1j}, ..., P_{nt}\}$: File vector set, where P_{it} represents the file vector corresponding to the t-th plaintext data of the i-th data owner
- $I = \{I_{11}, I_{12}, ..., I_{1j}, ..., I_{nt}\}$: Encrypted file vector set, where I_{it} represents the encrypted file vector corresponding to the t-th plaintext data of the i-th data owner

- $N = \{N_{11}, N_{12}, ..., N_{1k}, ..., N_{nk}\}$: Search keyword mapping number set, where N_{ti} represents the mapped numerical expression of the t-th search keyword for the i-th data owner
- V: Search vector
- T: Encrypted search vector

1.4 Design Goals

To effectively enable ranked search using outsourced cloud data under the aforementioned model, our scheme must ensure retrieval accuracy and efficiency.

Retrieval Accuracy: Classical encrypted retrieval models using statistical features cannot capture word context information and deep semantics of documents. Our scheme aims to research deep-level semantics of files rather than using statistical features as the basis for file retrieval. Instead of employing statistical feature methods, our model optimizes through neural networks, achieving higher retrieval accuracy than previous methods.

Efficiency: Efficiency encompasses both search and storage aspects. Naturally, reducing the dimensionality of generated vectors decreases storage and computational resource consumption, while designing an appropriate document index vector management structure can also improve retrieval efficiency.

2.1 Private Cloud Federated Learning Model

Federated learning is a manifestation of distributed machine learning architecture, comprising training servers and a central parameter server. All servers share the machine learning model to be trained and share parameters from each training round. After all training servers transmit parameters in ciphertext form to the central server, the central server unifies the parameters. The federated learning architecture enables centralized training on all data without sharing data, solving the sensitive data silo problem among various data owners.

Federated learning consists of a parameter server and data training parties, with all trainers sharing the training model [17]. Each data trainer trains their own data separately while sharing training parameters. Traditional ciphertext retrieval schemes generate file vectors and retrieval request vectors based on statistical models according to keyword term frequency and inverse document frequency. Schemes that improve retrieval accuracy on this basis involve retrieval keyword expansion, such as user interest models, expansion of similar keywords derived from deep learning, or weight updates based on keyword positions. Due to data security concerns, plaintext data has not been trained through deep neural networks to mine deep article semantics. This paper proposes a federated learning model based on private cloud architecture that updates the generation method of article vectors and retrieval vectors from traditional statistical models to deep learning models. Considering that computing performance varies among all data trainers, we design a time window management model to improve communication and training efficiency.

As shown in Figure 2, the time for one round of federated learning network model update consists of the data owner' s training time T_1 and parameter transmission and parameter server update time T_2. Considering that each data owner has different data volumes and computing capabilities, this paper designs a time window management model. Before federated learning begins, the entire system tests communication time, setting a communication window T_1 and overall window time T. The data owner' s training time is $T_2 = T - T_1$. For data owner 1, the first training parameter result is W_1^1, after which local training continues within time T_2. The number of local training rounds varies depending on the data owner' s data volume and private cloud computing capability. The training parameter result at the end is denoted as W_1^2 and sent to the parameter server. This scheme ensures that all user data owners participate in training when each round of model unification occurs, while also considering that data owners with strong computing capabilities can conduct multiple local training sessions to improve federated learning training effectiveness.

2.2 Neural Network Vector Generation Model

Unlike traditional statistically generated vector models, this paper utilizes a private cloud federated learning model to train a neural network vector generation model. The selected model is DSSM [18]. File vectors and retrieval vectors are mapped to the same dimensional deep semantic space through this model.

The DSSM model input consists of N documents and 1 query request. The neural network architecture comprises five layers. The first layer has an input dimension of 500k. After passing through the Word-n-gram layer, the dimension is reduced to 30k. This is followed by two fully connected deep neural network layers with dimensions of 300 and 300, respectively, outputting a 128-dimensional vector. The activation function for each layer of this model is tanh.

During training, the $N + 1$ 128-bit vectors generated by this model correspond to documents and query requests. To simplify ciphertext retrieval complexity, after generating the 128-dimensional vector, we normalize the vector as follows:

The relevance score is the dot product result of the query vector and file vector. The model objective is to optimize the likelihood of clicked documents, with the loss function shown in Equation (2):

where the posterior probability of relevance scores between forward documents and query requests is calculated through the softmax function, γ is the smoothing coefficient derived from real data testing background, D represents all documents including 4 non-clicked documents D^- and 1 clicked document D^+. In our scheme, parameters are randomly initialized, and each private cloud uses stochastic gradient algorithms for distributed training as shown in Equation (3):

In this paper, each data owner trains the DSSM model separately through a private cloud server using their own data, with parameter sharing conducted through the parameter server.

2.3 Encrypted Inverted Index Table

Inverted index tables are widely used in information retrieval, constructed in KEY-VALUE format where KEY is the document keyword set and VALUE is the document vector. In ciphertext retrieval, since the untrusted public cloud stores the retrieval table and we must prevent the public cloud server from analyzing data users' retrieval patterns, KEY values cannot be directly stored in plaintext keyword form on the public cloud server. This paper utilizes the discrete logarithm problem to map keywords to irregular numbers and updates the KEY values of the ciphertext inverted index table during each retrieval process. The encrypted inverted index table ensures the security of both data and data users, with the specific process shown in Figure 3.

During the initial retrieval stage, data users must send identity authentication to all data owners. After authentication, the data user possesses n key sets $\{S_1, S_2, ..., S_n\}$, where S_i represents the prime number p_i, integer g_i, and file key K_i shared between the i-th data owner and data user.

The data user generates a public key Y_{user} through a private key X_{user} as shown in Equation (4):

and sends it along with the search keyword set $Q = \{q_1, q_2, ..., q_k\}$ to data owner i. For data owner i, based on k keywords, k private keys X_{pcs} are generated, and subsequently Y_{pcs} and the search keyword set are transmitted, with the public key sent to the data user.

The data user calculates k keyword mapping numbers through Equation (5):

The data owner calculates k keyword mapping numbers through Equation (6):

After the data user conducts identity authentication and keyword mapping with all data owners, $n \times k$ mapping numbers are generated. The data user sends the mapping numbers and search keywords to the parameter server. Each data user updates the corresponding KEY values of the encrypted inverted index table based on their k mapping numbers, while randomly generating and sending the remaining KEY values to the public cloud.

3.1 Specific Scheme

Data owners encrypt documents and indexes by outputting keys through a random key generation algorithm and inputting a security parameter l. SK is a key set including an invertible matrix M_1, an n-bit vector S, and two $(n+1) \times (n+u+1)$ matrices M_2. Additionally, K is a symmetric key, and a is a primitive root of q.

After identity authentication, data users send search keywords to all data owners. Each data owner and the data user generate mapping numbers for search keywords based on a, q, and their respective X_{pcs} and Y_{pcs}, as described in Section 2.2. Data owners generate KEY values for the encrypted inverted index table based on the generated keyword mapping numbers. Data users possess $n \times k$ keyword mapping numbers, where k is the number of keywords and n is the number of data owners.

a) Federated Learning Deep Neural Network Model

Data owners transmit data information to private clouds to jointly train a prespecified neural network model. We stipulate that neural network parameters are unified in the initial round, and after each training round ends and before the next round begins, each private cloud server downloads the latest parameters from the parameter server for re-unification. The neural network model used in this paper is DSSM [18], with structure described in Section 2.2. The federated learning training method follows Section 2.1, considering that all users participate in each round of training while ensuring that users with high computing power can conduct multiple training sessions to improve federated learning efficiency.

Through the trained vector model from private clouds, all data owners generate 128-dimensional file vectors P_{it} from plaintext data D_{it}, extending them to $(128+u+1)$-dimensional vectors P'_{it}, where the last extension bit is set to 1 and others are set to random numbers. The file vector is encrypted using key SK to generate encrypted file vector I_{it}, specifically by splitting vector P'_{it} into P'_i and P''_i using S, with splitting rules shown in Equation (7):

Data users construct an inverted index table and place encrypted file vectors in the VALUE positions, uploading them to the public cloud.

Data users encrypt the dataset D using key K.

Data users send search keywords to the parameter server, which generates retrieval vectors based on the trained model and sends them to data users. Data users extend the received 128-dimensional retrieval vector V_t to a $(128+u+1)$-dimensional vector V'_t, where the last extension bit is a random number t, and other positions are supplemented with 0 or 1. Using S, the vector is split into V'_t and V''_t according to Equation (8):

The encrypted search trapdoor T is then generated, and data users send the trapdoor T and keyword mapping set N to the public cloud.

After receiving the encrypted search trapdoor T uploaded by data users, the public cloud compares the keyword mapping set N with the encrypted inverted index table to find all encrypted file vectors I in the corresponding VALUE positions where the KEY values match the keyword mapping set N. It calculates the top-k files through Equation (9) and sends them to data users.

After receiving the result file set E_q, data users decrypt using key K to obtain the plaintext information of result files.

3.2 Scheme Process

The specific workflow of the ciphertext retrieval scheme proposed in this hybrid cloud architecture is decoupled into two parts: the non-retrieval phase and the retrieval phase, as shown in Figure 4.

The non-retrieval phase includes building the vector generation model, which can generate retrieval vectors and file vectors. The federated learning model for hybrid cloud architecture proposed in Section 2.1 is used to train the neural network model in Section 2.2, generating the neural network vector generation model. Each data owner uses this model to input plaintext data and obtain file vectors, which are encrypted and managed using the encrypted inverted index table structure proposed in Section 2.3.

In the retrieval phase, data users send search requests to the parameter server that possesses the neural network vector generation model. After receiving the search request, the parameter server generates retrieval vectors and sends them to data users. Data users encrypt the retrieval vectors and send them to the public cloud, which retrieves relevant files through the encrypted inverted index table and sends them to data users. Data users then decrypt the target plaintext using their keys.

Due to data updates, the neural network vector generation model continuously evolves, causing the retrieval and non-retrieval phases to recur independently.

4 Security Analysis

First, trapdoors and document indexes are generated in the neural network model and dimensionality reduction is applied. Document and query content cannot be directly reflected in vectors. Additionally, pseudo-keywords, random splitting, and two $(n+1) \times (n+u+1)$ encryption matrices are introduced. As proven in [19], adversaries cannot construct sufficient equations to completely compute the matrices. Therefore, our proposed scheme effectively resists the known ciphertext threat model.

We reduce security under the known background knowledge model to understanding the internal relationship between document indexes and retrieval trapdoors. To further prevent curious public cloud servers from leaking and minimizing document information based on known background knowledge, we dynamically change trapdoor expressions. We use S to split vectors in the secret key SK. Consequently, even when users search for the same query multiple times, the received search request trapdoors differ. Simultaneously, all vectors introduce random numbers ϵ_j with uniform distribution, whose values follow a mean of μ and variance of $\sigma^2 = c/3$. According to the central limit theorem, $U(\mu - c, \mu + c)$ follows $N(\mu, \sigma^2)$. Higher c values increase security but reduce retrieval accuracy. Appropriate c values can effectively resist statistical analysis attacks. Our proposed scheme is also secure against known background knowledge threat models.

5 Performance Evaluation

We implemented the scheme using PYTHON on a computer with an Intel Core CPU at 2.9GHz, Windows 10 server, and 4GB RAM. The performance of our proposed system was compared with the MRSE scheme [2], PRSE scheme [12], and FMRSM scheme [20]. In experiments, we evaluated overall performance on a real dataset (hereinafter referred to as the evaluation dataset) and utilized the DSSM [18] network as the federated learning model. We randomly selected 20,000 English query samples from one year of query document log files. During model training, each retrieval request corresponded to 4 non-relevant documents and 1 relevant document. We conducted performance analysis from the perspectives of retrieval efficiency and document retrieval accuracy. Each simulation was repeated 10 times, with average results analyzed and presented. System modules and specific implementation schemes are shown in Table 1.

Table 1. Module Introduction and Implementation Method

Module	Implementation Method
Private Cloud Federated Learning Module	FedML open-source framework
Federated Learning Model	DSSM [20]
Encrypted Inverted Index Table Module	Custom algorithm
File Retrieval Module	Custom algorithm
File Encryption/Decryption Module	Crypt library

We compared document retrieval efficiency between our scheme and the aforementioned schemes (MRSE, FMRSM, and PRSE). As shown in Figure 5, retrieval time for all four schemes increases with the number of documents in the collection. MRSE' s search time grows approximately linearly with document set size, which is reasonable considering that the public cloud server needs to scan all document indexes during the search phase. FMRSM and PRSE schemes perform better, but our scheme outperforms all the above schemes. Since our scheme' s search process is based on encrypted inverted index tables and features lower vector dimensionality, it becomes more effective as the document set contains more files.

As shown in Figure 6, search time for the three schemes (MRSE, PRSE, and FMRSM) remains roughly constant regardless of the number of keywords. However, all three schemes' retrieval times are significantly higher than our scheme.

We measure document retrieval accuracy by the proportion of relevant documents among all returned results. As shown in Figure 7, compared with MRSE, our scheme' s search accuracy remains above 95% regardless of document quantity. In contrast, MRSE' s search efficiency gradually decreases from nearly 90% to 80%.

6 Conclusion

This paper proposes a deep semantic ciphertext retrieval scheme under a hybrid cloud architecture. We utilize private clouds for federated model learning, solving data silo and security issues, extracting deeper semantic information from files, and improving retrieval accuracy. Using an encrypted inverted index table structure, we enhance retrieval efficiency while ensuring that data users' search keywords are not recorded by the public cloud. Analysis and simulation results demonstrate that our scheme provides secure and efficient encrypted document search for data users.

In future work, we plan to optimize the neural network structure or employ better models to mine deeper semantics of documents in encrypted retrieval. Additionally, we will combine specific models to construct vector storage models that improve retrieval efficiency.

References

[1] Cao Ning, Wang Cong, Li Ming, et al. Privacy-Preserving multi-keyword ranking [J]. Journal of Cloud Computing. 2014, 3 (1): 1-11.

[2] Xia, Zhihua, Chen Li, Sun Xingming, et al. A multi-keyword ranked search over encrypted cloud data supporting semantic extension [J]. International Journal of Multimedia and Ubiquitous Engineering, 2016, 11. 8: 107-120.

[3] Cai Chengjun, Weng Jian, Yuan Xingliang. et al. Enabling reliable keyword search in encrypted decentralized storage with fairness [J]. IEEE Transactions on Dependable and Secure, 2021, 18 (1): 131-144.

[4] Li Feng, Ma Jianfeng, Miao Yinbin, et al. Verifiable and dynamic multi-keyword search over encrypted cloud Data Using Bitmap [J]. IEEE Transactions on Cloud Computing. 2021: 1-1.

[5] Liu Lianggui and Chen Qiuxia. A novel feature matching fanked search mechanism over encrypted cloud data [J]. IEEE Access, 2020, 8: 114057-114069.

[6] Zhang Dong, Fan Qing, Qiao Hongyi, et al. A public-key encryption with multi-keyword search scheme for cloud-based smart grids [C]// 2021 IEEE Conference on Dependable and Secure Computing, 2021: 1-6.

[7] 沈学利，崔海韵，陈鑫彤．一种支持撤销的位置分层属性加密研究 [J]．计算机应用研究，2019, 37 (1): 216-220. (Shen Xueli, Cui Haiyun, Chen Xintong. Research on encryption of location hierarchical attribute supporting revocation [J]. Application Research of Computers, 2019, 37 (1): 216-220.)

[8] Miao Yinbin, R. Deng, K. K. R., et al. Threshold multi-keyword search for cloud-based group data sharing [J]. IEEE Transactions on Cloud Computing, 2020, 99: 1-1.

[9] 路宏琳，王利明．面向用户的支持用户掉线的联邦学习数据隐私保护方法 [J]．信息网络安全，2021, 21 (3): 64-71. (Lu Honglin, Wang Liming. User-oriented Data Privacy Preserving Method for Federated Learning that Supports User Disconnection [J]. Netinfo Security, 2021, 21 (3): 64-71.)

[10] 张佳乐，赵彦超，陈兵，等．边缘计算数据安全与隐私保护研究综述 [J]．通信学报，2018, 39 (3): 1-21. (Zhang J L, Zhao Y C, Chen B, et al. Survey on data security and privacy-preserving for the research of edge computing [J]. Journal on Communications, 2018, 39 (3): 1-21.)

[11] Zhang ke, Long Jiahuan, Wang Xiaofen, et al. Lightweight searchable encryption protocol for industrial internet of things [J]. IEEE Transactions on Industrial Informatics, 2020, 17 (6): 4248-4259.

[12] Fu Zhangjie, Ren Kui, Shu Jiangang, et al. Enabling personalized search over encrypted outsourced data with efficiency improvement [J]. IEEE Transactions on Parallel Distributed Systems, 2016, 27 (9): 2546–2559.

[13] J. Nagesh, N. Jyoti, B. Sayli. Semantic search supporting similarity ranking over encrypted private cloud data [J]. Int. J. Emerging Eng. Res. Technol, 2014, 2 (7): 215–219.

[14] Xia Zhihua, Zhu Yanling, Sun Xingming, et al. Secure semantic expansion based search over encrypted cloud data supporting similarity ranked search over encrypted cloud data [J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25 (2): 222-233.

[15] Fu Zhangjie, Wu Xinle, Guan Chaowen, et al. Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement [J]. IEEE Transactions on Information Forensics and Security. 2017, 11 (12): 2706–2716.

[16] Yang Wenyuan, Zhu Yuesheng. A verifiable semantic searching scheme by optimal matching over encrypted data in public cloud [J]. IEEE Transactions on Information Forensics and Security. 2020, 16: 100–115.

[17] Wu Wentai, He Ligang, Lin Weiwei, et al. SAFA: a semi-asynchronous protocol for fast federated learning with low overhead [J]. IEEE Transactions on Computers. 2020: 1-1.

[18] Shen Yelong, He Xiaodong, Gao Jianfeng, et al. A latent semantic model with convolutional-pooling structure for information retrieval [C]// Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. New York: ACM, 2014: 101-110.

[19] Chen Chi, Zhu Xiaojie, Shen Peisong, et al. An efficient privacy-preserving ranked keyword search method [J]. IEEE Transactions on Parallel and Distributed Systems. 2016, 27 (4): 951-963.

[20] Wong Kit Wong, D. W. Cheung, B. Kao, et al. Mamoulis. Secure knn computation on encrypted databases [C]// Proc. ACM SIGMOD Int' l Conf. Management of Data (SIGMOD). 2009: 139-152.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.