Research on Homomorphic Retrieval Method of Private Database Secrets in Multi-server Environment

Fu-lian Zhong[1(✉)] and Jin-hua Liu[2]

[1] School of Mathematics and Computer Science, Xinyu University, Xinyu 338031, China
zhongfulian682@163.com
[2] Department of Professional and Continuing Education, Xinyu University, Xinyu 338031, China

Abstract. The traditional homomorphism retrieval method of privacy database is very complicated. In order to reduce the running time, this paper designs a homomorphic secret retrieval method for private database in multi-server environment. After the establishment of the secret homomorphism vector model, the semantic classification of the secret homomorphism ciphertext is carried out. Then, according to the characteristics of the neighborhood structure, the mapping interval is divided, and the HASH function is used to perform operations in the mapping interval. This process can reduce the computational complexity of the secret homomorphic ciphertext. Finally, a secret homomorphic retrieval model is established and an optimized retrieval algorithm is designed. Design experiments and compare the three conventional retrieval methods. According to the experimental data, in different mapping intervals, the average retrieval time of this method is 14.32 s, while the average retrieval time of the three control groups is 23.74 s, 29.03 s, At 20.92 s, the retrieval time of this method is shorter than that of the conventional method, which makes the homomorphic retrieval method of private databases more concise.

Keyword: Multi-server environment · Privacy database · Secret homomorphism · Data retrieval

1 Introduction

Database system is the core component of computer information system, and its security problem is an important aspect to be studied in the field of information security. For some important or sensitive data, users are eager to store and transmit in the form of ciphertext, and can operate on the database information without revealing the content. The database encryption method can be applied to different environments, but there is a common problem that the formed ciphertext database cannot be operated. In other words, for a ciphertext database, if you want to perform mathematical

S. Wang et al. (Eds.): IoTCare 2021, LNICST 414, pp. 244–259, 2022.
https://doi.org/10.1007/978-3-030-94185-7_17

operations such as statistics, averaging, and summation on certain fields, you must first decrypt these fields, then perform mathematical operations on the plaintext, and then encrypt [1]. In this way, firstly, the space-time overhead is increased; secondly, in actual applications, for some important or sensitive data, it cannot meet the needs of users to operate on them without letting users know the information in them. If you can perform mathematical operations and regular database operations on the ciphertext database, you can obviously solve the above problems, and can greatly reduce the time and space overhead required for encryption/decryption, and improve the operating efficiency of the database. Secret homomorphism technology is an effective method to solve the above problems.

A homomorphism retrieval method for database secret is designed in Reference [2]. This technology uses a specific algorithm to segment the data first, and obtains its index number. The index number represents the data of the corresponding data segment. Although this method can greatly reduce the scope of the search, it can only improve the accuracy of the search because of the increase of the search steps, but cannot reduce the search time. Reference [3] uses the interaction between the server and the client to reduce the retrieval time. However, in the retrieval process of this method, the values that have been recorded with the same attributes are always mapped to the same set. After rigorous statistical processing, they are vulnerable to attacks. Reference [4] proposed a method of using the divide-and-conquer principle to build indexes on non-homomorphic ciphertexts, which realized fast data retrieval. However, such a retrieval method is very clear to the opponent, so it can quickly track and locate the database access process, and it is easy to expose the partial order relationship of the data. This method is difficult to deal with the dynamic analysis of the adversary, which makes it difficult to guarantee the security of the database.

In order to solve the shortcomings of the above methods, a new homomorphism retrieval method for private database in multi-server environment is designed in this paper. The design idea of the new method is as follows:

(1) Based on the establishment of the secret homomorphic vector model, the semantic classification of the secret homomorphic ciphertext is carried out, so as to effectively reduce the search scope.
(2) Divide the mapping interval according to the features of the ciphertext neighborhood structure, and then use the HASH function to perform operations within the mapping interval to reduce the computational complexity of the secret homomorphic ciphertext.
(3) Establish the secret homomorphic retrieval model and design the retrieval algorithm to optimize the homomorphic retrieval process in the multi-server environment.

2 Design of a Homomorphic Retrieval Method for Private Database Secrets Based on a Multi-server Environment

2.1 Establish a Secret Homomorphic Vector Model

Firstly, the key parameters in the secret homomorphism of the private database are preprocessed, and a vector composed of multiple key parameters is used to represent the vector model of the secret homomorphism of the database to improve the accuracy of retrieval. On this basis, the secret homomorphism in the database is classified into ciphertext semantics, and the ciphertext semantic feature vector is extracted. At this time, a vector model should be established first, so that the vector can represent the secret homomorphism in this feature space, and all the ciphertext semantic sentences related to the secret homomorphism in the database are extracted to establish this vector model. Each feature in the vector model has its feature value, that is, the weight of the feature. Each ciphertext semantic keyword is one of the characteristic dimensions. Assuming there are x ciphertext semantic keywords in the database, the characteristic dimension in the database is also x. For a secret homomorphism d_j, the frequency of the ciphertext semantic keyword in the database can be used to calculate the weight of the ciphertext semantic keyword, and the similarity can be used as its continuous measurement index. The formula for calculating the frequency of occurrence of ciphertext semantic keywords is as follows:

$$TF : f_{ij} = \frac{freq_{ij}}{Maxfreq_j} \tag{1}$$

Among them, $TF : f_{ij}$ represents the frequency of ciphertext semantic keywords, $freq_{ij}$ represents the number of times the ciphertext semantic keywords appear in the secret homomorphism, and $Maxfreq_j$ is the number of times the ciphertext semantic keywords appear most frequently in the secret homomorphism [5]. This formula is mainly used to calculate the relative frequency of a ciphertext semantic keyword in a secret homomorphism.

The ciphertext semantic keywords appearing in the secret homomorphism are converted into a vector in the order of frequency, the contents of the secret homomorphism are extracted, attributes and attribute values are extracted, and the ciphertext semantic vector describing the contents of the secret homomorphism is described. Let E_a and E_b represent the encryption and decryption steps of the plaintext, then the elements in the plaintext data space are a finite set that can become a secret homomorphic space with $(E_{a1}, E_{a2}, E_{a3}, \cdots, E_{an})$ as the main body [6]. In order to improve the efficiency of secret homomorphic retrieval, classify the secret homomorphic ciphertext semantics. Based on the above extraction of secret homomorphic ciphertext keywords, classify the relevant ciphertext semantic vectors and describe the hierarchical concept tree As shown in Fig. 1.

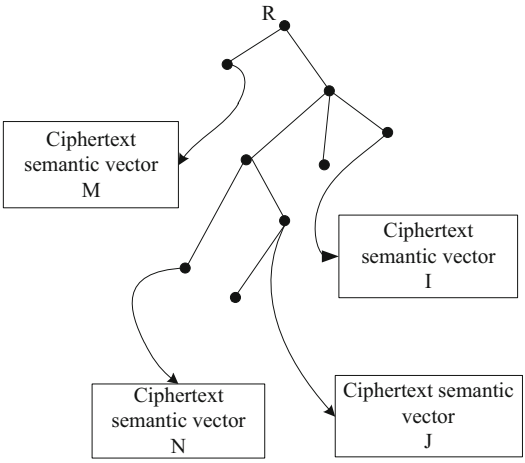


Fig. 1. Semantic classification of secret homomorphic ciphertext

As shown in Fig. 1, R is the source of the hierarchical concept tree. When the retrieval method in this paper classifies the ciphertext semantic vector, one source can be used as the origin of the hierarchical concept tree to derive other ciphertext semantic vectors, and finally deduced Multiple different ciphertext semantic vector branches provide a theoretical basis for the secret homomorphic ciphertext semantic classification index.

2.2 Refine Search Scope Based on Ciphertext Semantic Similarity

Each secret homomorphism has different ciphertext semantic characteristics, so when calculating the similarity of secret homomorphism, the problem of conceptual attributes needs to be considered first. Suppose concept A has an instance M. At this time, the instance M can be represented as $M = A[P]$, where $P = (P_1, P_2, \cdots, P_m)$ is the same as in the instance $N = A[Q]$, $Q = (Q_1, Q_2, \cdots, Q_n)$. At this time, the similarity of instance N and M can be calculated. First, the attribute vector of instance N and M is a common attribute vector through the above method, and then the similarity of the secret homomorphism is calculated according to the attribute value, and the attributes of the two instances are compared. Value and similarity, the obtained formula is shown below.

$$\begin{aligned} Sim_p(P, Q) &= Sim_p(P', Q') \\ &= \sum_{t=1}^{r} \frac{\mu_1 + \gamma_1}{2} Sim_p(p', q') \end{aligned} \tag{2}$$

Among them, μ_1 and γ_1 are the weight coefficients of attributes p' and q' in each vector, which is a preset parameter, usually the statistical value obtained after preprocessing the secret homomorphism, and the secret homomorphism is determined by this

statistical value. The final weight value of the state, the value range is [0.1]. Use deep learning to explore the search range of ciphertext semantic similarity. Generally, a mathematical model is established through a convolutional neural network, and the time sequence information of the text data is considered. The data information output at the current moment is regarded as the data information output at the previous moment, and input at the current moment. The superposition of data information [7, 8]. And build the entire convolutional neural network into a three-in-one model structure including the input layer, hidden layer, and output layer. Through the training of the neural network, a more detailed search range of ciphertext semantic similarity is obtained, and the text data information is processed Classification processing, the specific structure of this classification model is shown in the figure below.

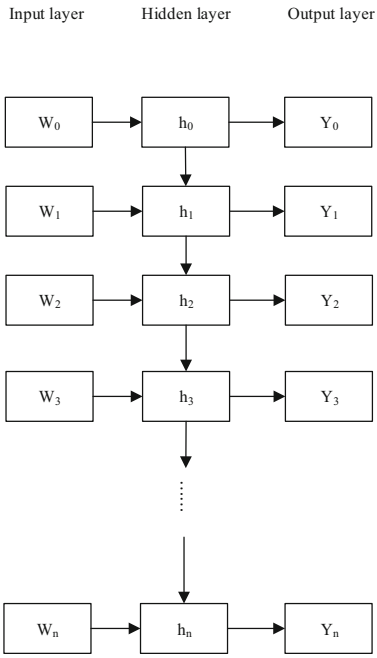


Fig. 2. Deep learning model structure

As shown in Fig. 1, assuming that the data text of the input layer is w = (w0, w1, w2, w3, …, wn), the text data of the input layer is sequentially converted to the hidden layer, so that the data becomes h = (h1, h2, h3, ……, hn), and then through the trained neural network model, the hidden layer data is converted into the data text y of the output layer through calculation. The calculation process is as follows:

$$\begin{cases} h_t = f(W_{xh}x_t + W_{hh}h_t - 1 + b_h) \\ y_t = W_{hy}h_t + b_y \end{cases} \tag{3}$$

Among them, W_{xh} represents the function converted from the input layer to the hidden layer, W_{hh} represents the internal conversion function of the hidden layer, W_{hy} represents the function from the hidden layer to the output layer, b_h represents the deflection vector from the input layer to the hidden layer, and b_y represents the deflection vector from the input layer to the hidden layer. The deflection vector from the hidden layer to the output layer. The update mode of the input layer, hidden layer and output layer can be obtained by formula (3), and the state information of the input layer and output layer at the previous moment can be obtained by training the hidden layer, and then combined with the convolutional neural network to obtain the input at the current moment And output text information, and to achieve the purpose of reducing the search range of ciphertext semantic similarity [9]. Train the objective function through the convolutional neural network, and redefine the function that changes after passing through the hidden layer:

$$L = -\sum_{i=1}^{T} Y_i \log(y_i) \tag{4}$$

Among them, T represents the total amount of text information that needs to be classified, Y_i represents the product of the probability distribution of each category after the text information is classified and its predicted value, and y_i represents the probability distribution value of each category after the text information is classified. Then, the neighborhood structure of edge points and noise points in the search range of ciphertext semantic similarity is set as shown in Fig. 3.

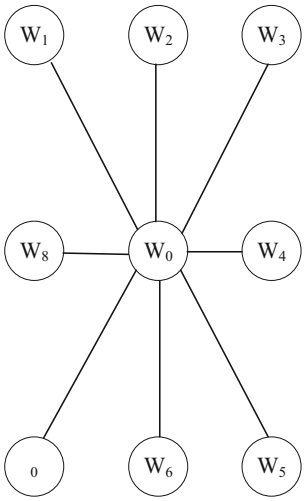


Fig. 3. Neighborhood structure features

As shown in the figure above, W_0 is the center of the entire image, and there are 8 neighborhoods around it. The connection between each neighborhood and the point W_0 of the image center is defined as:

$$Set_t = \{(0,t)/1,2,\cdots,8\} \tag{5}$$

In the formula, t is the connection from any point t to the center 0 point. Therefore, it can be considered that when the center 0 point is at the edge of the image, any Set_t is around the center 0 point, that is, the edge of the neighborhood structure [10]. The average gradient direction of the center 0 point and any point t in its surrounding neighborhood is defined as Ang_t. The noise point of Ang_t is usually large. Therefore, it is necessary to suppress noise and construct a neighborhood structure through a deep learning model to reduce the search range.

2.3 Divide the Mapping Interval

After narrowing the scope of database secret homomorphism retrieval, the method designed in this paper can basically obtain the similarity measurement method of ciphertext semantics, and accurately reflect the similarity between two ciphertexts of the same type. It directly combines the two ciphertexts. The text is quantified as a numerical value to indicate the degree of similarity for better retrieval. At this time, the similarity of the two ciphertexts can be expressed by the method of mapping intervals. The greater the distance, the greater the difference between the two ciphertexts, and the greater the distance, the smaller the difference between the two ciphertexts. The distance formula used is different, which will directly affect the accuracy of the retrieval algorithm. This paper uses the chi-square distance method to measure the similarity between two ciphertexts. The formula for calculating the chi-square distance is as follows:

$$D(a,b) = \sum_t \frac{(a_t - b_t)^2}{(a_t + b_t)} \tag{6}$$

Among them, a_t represents the sample image A, b_t represents the image B to be retrieved in the image database; $D(a,b)$ represents the chi-square distance between the sample image A and the image B to be retrieved [11, 12]. In this paper, the similarity measurement algorithm of two ciphertexts is constructed by the chi-square distance. When the chi-square distance is 0, it means that the two ciphertexts are exactly the same. After the distance of the mapping interval is obtained, all the necessary data contained in it can be obtained through the secret homomorphism search that has reduced the search space. To perform a secret homomorphic search on the ciphertext database, the ciphertext data must be divided into intervals. According to the retrieval conditions, several divided contents are returned, and the returned contents must contain all the required data. According to the obtained index value, we divide it according to the specified rules. The division must meet the following conditions: First, all division intervals must cover all data. Second, the division intervals cannot overlap. Third, there is and can only correspond to one interval after the specific value mapping. Assuming that the divided intervals are continuous, the maximum and minimum values of each interval can be found. Use the traditional equation to define the partition function, which is described as follows:

$$\Psi(W_i, W_j) = \{\gamma_1, \gamma_2, \gamma_3, \cdots, \gamma_n\} \tag{7}$$

In the formula, W_i represents the traditional divided interval that has been abandoned; W_j represents the divided interval currently in use; $\{\gamma_1, \gamma_2, \gamma_3, \cdots, \gamma_n\}$ represents the sub-interval segmentation of an interval. According to the above division function, select an appropriate division method, then the data to be retrieved can be divided into n parts. However, for a large database, there is a lot of data in each divided segment. After the secret homomorphic retrieval, the server returns the result of the secret homomorphic retrieval to the client for accurate retrieval, and the scope of the retrieval is also very large. This paper proposes a sub-division method. On the basis of the traditional division function, each division segment is continued to be sub-divided. Such a division method has very important significance for the accuracy of future secret homomorphic retrieval. Different fields may require different division rules and functions, such as division by breadth, division by depth, etc. [13]. There are many ways to divide, and the selection requirements must meet the three requirements mentioned above. After the division is successful, in order to facilitate the secret homomorphic retrieval, the divided interval is replaced with a specific value or similar value that can be marked. Here, a collision-free HASH function is needed, and the HASH function maps the divided interval to a specific mark value. At the same time, the HASH function can be used to mark the divided intervals one by one. For example, at this time, the upper bound of the interval and the divisor of 40 can be selected as the interval marker. The specific interval markers are shown in Fig. 4.

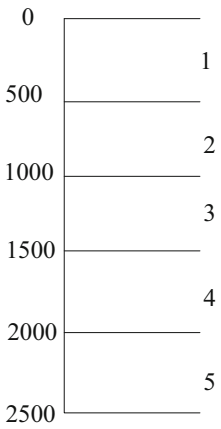


Fig. 4. Secret homomorphic segmentation interval division

It can be seen that the data processing is divided into two steps. The first step is to divide the overall data into equal parts. In the second step, the sub-division interval is refined according to the division function of the first step. The marking of the interval does not necessarily need to be in an increasing order, as long as it is satisfied that the

HASH is collision-free within the specified range, the HASH function can be used to perform the marking operation.

2.4 Establish a Secret Homomorphic Retrieval Model

The secret homomorphic search is performed on the basis of the newly established ciphertext index. Figure 3, 4 shows the specific implementation process. There are three main steps in secret homomorphic retrieval, condition mapping, retrieval sentence conversion, and retrieval. The main purpose of secret homomorphic search is to reduce the search scope of precise search, so that the client search time can be greatly shortened [14]. The idea of secret homomorphic retrieval introduced in this article is.

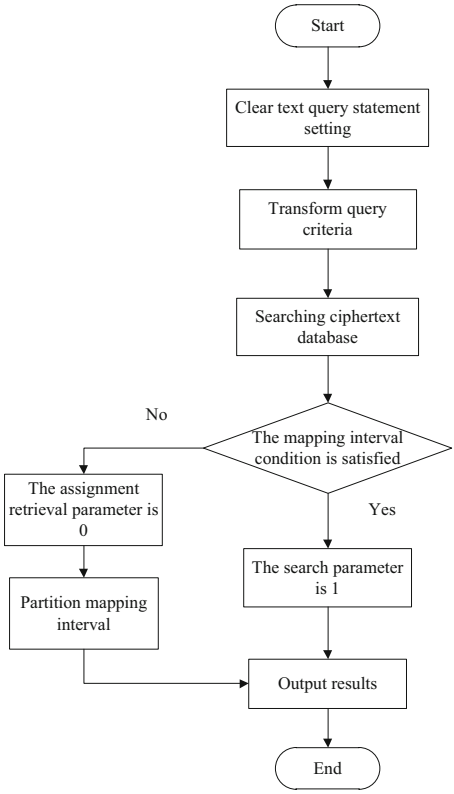


Fig. 5. Algorithm structure design

As shown in Fig. 5, when inserting data into the database, the data is processed first. According to the characteristics of the data itself, the data is grouped for storage. And to select a unique value to mark the group. Transform the search conditions and associate them with the grouping mark, so that the plaintext and the ciphertext are reconnected. When storing ciphertext data, add a flag option after each record. The

initial value of the flag option is 0. In the second step, it is judged whether all the divided intervals meet the conditions. If all the conditions are met, the flag flag is set to 1. Otherwise, compare the search conditions with each partition according to the search conditions and the divided partitions, exclude the partitions that do not meet the conditions, and return the partition data that contains the correct search object. For partitions that partially meet the conditions, the search conditions are compared with their sub-partition marks. Return the numerical region corresponding to the sub-division mark as the result of the secret homomorphic search. In this process, dividing the interval is a key factor affecting the accuracy and time consumption of secret homomorphic retrieval. The finer the division interval, the higher the accuracy of the retrieval.

3 Experimental Study

3.1 Experiment Preparation

Compared with translating ciphertext into plaintext form and then performing mathematical calculations, secret homomorphism has a great advantage of simplicity. Especially in the field of ciphertext retrieval, traditional retrieval methods are very complicated, but secret homomorphism can be fast and efficient. Get the information that needs to be retrieved and display it to the customer. This experiment is mainly used to test whether the database secret homomorphism in a multi-server environment can be accurately searched while improving retrieval efficiency, and the answer can be obtained through the comparison of retrieval time. The server in this experiment is divided into a master and a client, and its overall architecture is shown in Fig. 6.

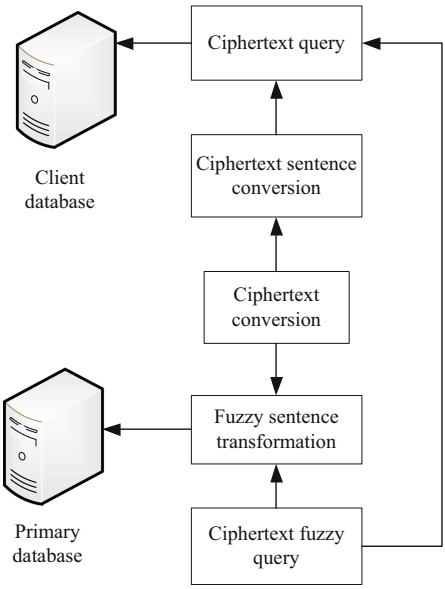


Fig. 6. Server architecture

As shown in Fig. 6, the main frame of the server architecture design is divided into two parts, the client part and the master part. The main task of the server part is to conduct secret homomorphic search, narrow the search scope of precise search, and reduce the search time of the client. The client mainly performs precise search and decrypts the information returned by the server. Among them, the flow of search sentences and non-search sentences is different [15]. Since the retrieval statement has a specific return result, it needs to be temporarily stored on the client side to facilitate accurate retrieval. In Fig. 6, the exact search sentence and the secret homomorphic search sentence are different secret homomorphic search sentences, and the search conditions need to be mapped to the ciphertext state according to the ciphertext index to perform the secret homomorphic search. Since the ciphertext index is excluded from the results returned by the secret homomorphic search sentence, the precise search sentence only homomorphically encrypts the search condition to facilitate accurate search. This experiment is mainly divided into two aspects. First, the performance of the retrieval method is tested, and the retrieval performance of the master server and the client server are tested respectively. Second, it takes time to test the ciphertext retrieval function of the secret homomorphism technology. According to the test needs, this experiment will give the time consumption of the plaintext to judge. Compare the conventional retrieval method with the retrieval method in the text, and analyze whether the method in this paper has realized the optimization of convenience.

3.2 Classification Performance Test

The test analysis of homomorphic technology is mainly to detect the ciphertext expansion after the homomorphic encryption processing. In this process, it is necessary to detect the ciphertext expansion of the client database and the master database separately. When calculating the expansion of the data, it is necessary to determine the main factors of its expansion according to the definition of modulo operation in the previous article, and combine the main factor n to obtain the passive factor P. On the premise that P is a large prime number, the effect of data expansion is very significant. The database table design of this experiment is shown in Table 1.

Table 1. Database ciphertext expansion test table

Field	Symbol	Character type	Is it empty?
Plaintext 1	N1	Float	No
Plaintext 2	N2	Float	No
Plaintext 1+ Plaintext 2	N1 + N2	Float	No
Plaintext 1 × Plaintext 2	N1 × N2	Float	No
Ciphertext 1	M1	Float	No
Ciphertext 2	M2	Float	No
Ciphertext 1 + ciphertext 2	M1 + M2	Float	No
Ciphertext 1 × ciphertext 2	M1 × M2	Float	No
Plaintext 1 to ciphertext 1	N1 → M1	Float	No
Plaintext 2 Convert ciphertext 2	N2 → M2	Float	No

Combining the storage comparison of plaintext data and ciphertext data, the ciphertext storage data of the main-end database and the client-side database can be obtained as shown in Table 2 and Table 3. Among them, the retrieval method of this article is set as the experimental group, and the conventional retrieval method is set as the control group 1-the control group 3.

Table 2. Main-end database ciphertext storage

	Test group	Control group 1	Control group 2	Control group 3
N1	4.9	−5.6	7.2	9.3
N2	8.6	12.3	−3.5	10.7
N1 + N2	16.7	13.2	12.9	−10.1
N1 × N2	52.253641	24.639745	−12.369742	39.621452
M1	25324.202	12536.745	16328.963	−45863.142
M2	25639.142	12574.369	−14256.745	41253.963
M1 + M2	425639.112	458963.745	−125798.312	256364.756
M1 × M2	225896343.526	274896512.369	−147963428.258	546328961.248
N1 → M1	256342.96	412563.75	−142586.48	127463.52
N2 → M2	563214.85	−178563.42	145286.42	463285.96

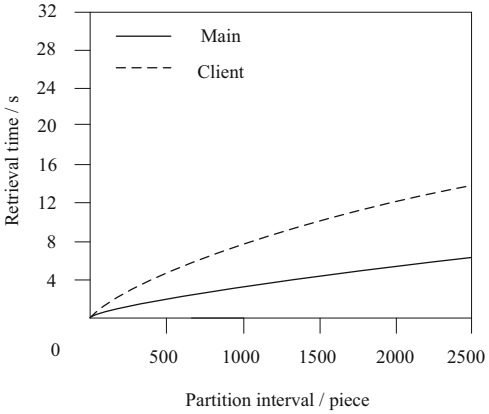
Table 3. Client database ciphertext storage

	Test group	Control group 1	Control group 2	Control group 3
N1	2.3	−3.5	7.5	8.3
N2	8.4	12.1	−2.3	9.7
N1 + N2	12.6	15.8	16.1	−10.4
N1 × N2	56.32414	23.56915	−14.36982	34.12572
M1	12356.74	45698.25	53152.75	−14763.52
M2	45628.362	41526.760	−14863.285	47856.251
M1 + M2	425479.168	428568.775	−136947.307	275463.149
M1 × M2	456975246.185	852643257.125	−964531752.365	415862463.75
N1 → M1	258963.14	128476.25	−163725.69	158963.12
N2 → M2	746321.85	−145286.25	756395.12	415963.14

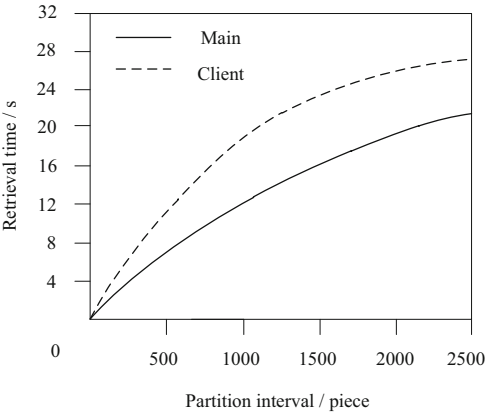
From Table 2 and Table 3, it can be seen that the ciphertext data in the main-end database and the client-side database are expanded very obviously, and the specific expansion multiple is related to the size of r. The storage size of the ciphertext of the experimental group is about three times that of the three control groups. To four times. And such ciphertext expansion coefficient proves that the secret homomorphic retrieval method of private database based on multi-server environment in this paper has good performance.

3.3 Actual Retrieval Time Test

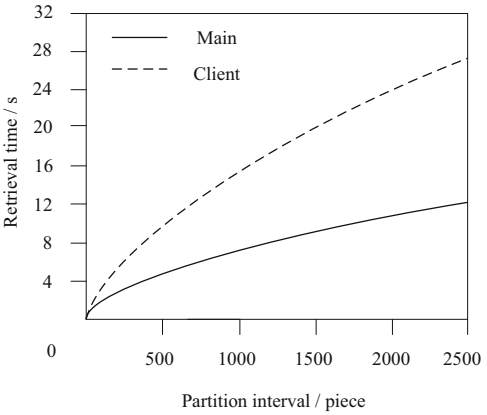
The biggest factor that affects the retrieval time is the size of the data ciphertext division interval, or the number of intervals. Figure 7 shows the time consumption



(a) test group

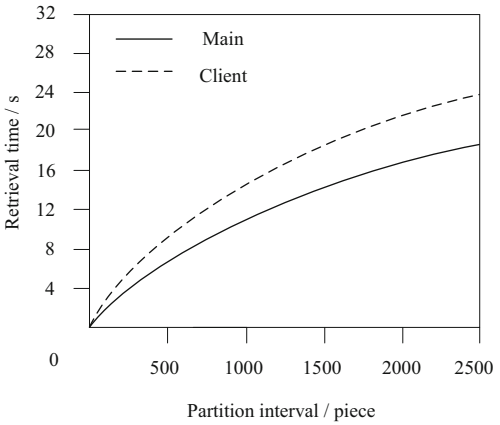


(b) Control group 1



(c) Control group 2

Fig. 7. Retrieval time test



(c) Control group 3

Fig. 7. (*continued*)

change diagram of the system in the search operation with join, as the division interval increases, and the number of sub-division intervals does not change.

In general, the total time consumed by the host and the client is shown in Table 4 under the premise that the division interval of the four retrieval methods is increasing.

Table 4. Total time consumption

	Test group	Control group 1	Control group 2	Control group 3
100	2.14	2.75	5.62	4.15
200	3.25	4.36	8.63	6.26
300	4.58	7.41	11.41	8.42
400	7.12	10.25	14.25	10.85
500	8.09	13.59	17.42	12.28
600	9.16	14.75	19.14	14.25
700	10.23	16.62	21.26	15.34
800	11.34	18.51	23.34	17.41
900	12.45	20.76	25.75	18.28
1000	13.17	22.18	26.17	19.05
1100	14.03	23.16	27.14	20.36
1200	14.84	24.28	28.45	21.42
1300	15.46	26.36	30.36	22.96
1400	15.75	27.15	32.75	23.42
1500	16.23	28.52	33.42	24.08
1600	16.94	29.36	34.42	25.14

(*continued*)

Table 4. (*continued*)

	Test group	Control group 1	Control group 2	Control group 3
1700	17.65	29.41	36.52	26.32
1800	18.12	30.85	37.61	26.85
1900	18.97	31.14	39.07	27.64
2000	19.56	32.15	40.12	28.05
2100	20.36	32.47	41.14	28.95
2200	21.04	33.52	42.15	29.42
2300	21.85	34.63	42.16	29.96
2400	22.52	35.46	43.32	30.53
2500	23.25	36.78	44.15	31.52

According to Fig. 7 and Table 4, it can be seen that the running time of the four retrieval methods is related to the division function. If the fields to be operated are divided into more intervals, the overall time spent is reduced. The finer the interval division, the finer the data retrieved by the secret homomorphic search. However, when the interval is divided to a certain extent, the time consumption tends to stabilize and approach a stable value. Explain that although the division of intervals has a great influence on the system time consumption, within a certain range, the consumption time will decrease as the division of intervals increases. Beyond a certain time limit, it remains stable. Among them, the average retrieval time of the experimental group was 14.32 s, while the average retrieval time of the three control groups were 23.74 s, 29.03 s, and 20.92 s. According to the above experimental results, the proposed homomorphic secret retrieval method of private database has a shorter running time than the three traditional methods, which proves that its retrieval process is simpler.

4 Conclusion

The privacy of database is always the core problem of information security. However, the process of converting plaintext to ciphertext also makes the computation cumbersome. After a lot of practice, the secret homomorphism of database is the key to solve the problem of convenient ciphertext calculation of database. To solve this problem, this paper designs an effective homomorphism retrieval method for privacy database. In the experiment part, the conversion and computation time of ciphertext data are tested. Compared with the three traditional methods, the proposed method consumes much less time, so it can be seen that the proposed method can calculate ciphertext data more quickly and conveniently.

In the following research, the method in this paper will be further optimized to improve its retrieval accuracy.

References

1. Zhou, N., Zhang, M.-q., Liu, M.-m.: Reversible data hiding algorithm in homomorphic encrypted image based on secret sharing. Sci. Technol. Eng. **20**(19), 7780–7786 (2020)
2. Tan, Y., Lu, L., Wang, J.: Ciphertext-policy attribute encryption scheme based on homomorphic encryption. Comput. Eng. Appl. **55**(19), 115–120+127 (2019)
3. Li, S., Jing, Z.: Analysis and development strategy of cross-language retrieval function for "the belt and road" multilingual shared database. Library Inf. Ser. **65**(3), 20–27 (2021)
4. Liu, Y.-j.: Design of rapid retrieval system of archives information database based on MapReduce. Electron. Des. Eng. **28**(13), 45–49 (2020)
5. Wang, N., Zheng, K., Fu, J., et al.: Method of ciphertext retrieval in mobile edge computing based on block segmentation. J. Commun. **41**(7), 95–102 (2020)
6. Xiang, C.H.E.N., Heng, H.E., Peng, L.I., et al.: Ciphertext image retrieval scheme based on target detection in cloud environment. Comput. Eng. Appl. **56**(11), 75–82 (2020)
7. Yang, X., Chen, G., Li, T., et al.: Multi-user ciphertext retrieval scheme based on certificateless cryptosystem. Comput. Eng. **46**(9), 129–135 (2020)
8. Liu, J., Zheng, X., Zheng, D., et al.: Secure attribute based encryption enabled cloud storage system with ciphertext search. Netinfo Secur. (7), 50–58 (2019)
9. Liu, S., Liu, D., Muhammad, K., Ding, W.: Effective template update mechanism in visual tracking with background clutter. Neurocomputing (2020). https://doi.org/10.1016/j.neucom.2019.12.143
10. Zhongyuan, Q.I.N., Yin, H.A.N., Xuejin, Z.H.U.: Research on ciphertext full-text retrieval of cloud storage based on improved DGHV algorithm. Netinfo Secur. **1**, 8–15 (2019)
11. Liu, S., Liu, X., Wang, S., Muhammad, K.: Fuzzy-aided solution for out-of-view challenge in visual tracking under IoT assisted complex environment. Neural Comput. Appl. **33**(4), 1055–1065 (2021)
12. Dai, H., Yang, G., Min, Z.: Multi-keyword parallel ciphertext retrieval scheme in distributed environment. J. Comput. Appl. **39**(10), 2948–2954 (2019)
13. Zhang, Y., Liu, X., Lang, X., et al.: Multi-server key aggregation searchable encryption scheme in cloud environment. J. Electron. Inf. Technol. **41**(3), 674–679 (2019)
14. Liu, S., Liu, D., Srivastava, G., Połap, D., Woźniak, M.: Overview and methods of correlation filter algorithms in object tracking. Complex Intell. Syst. **7**(4), 1895–1917 (2020). https://doi.org/10.1007/s40747-020-00161-4
15. Shu, Q., Wang, S., Han, L.: Analysis and improvement of a three-factor authentication protocol in the multi-server environment. J. Hangzhou Normal Univ. (Nat. Sci. Ed.) **20**(1), 91–94+112 (2021)