A Heterogeneous Database Encryption Algorithm Based on B/S Structure

Jun Chen [1, a*], Feng Zhang [2, b]

[a]2010034@wtu.edu.cn, [b]305650255@qq.com

[1]Wuhan Textile University, Wuhan Hubei 430200, China

[2]Wistron ITS., Wuhan Hubei 430200, China

Abstract: In order to solve the problem that the key of the traditional method is uneven between the user and the authority center, which leads to the long encryption time of a heterogeneous database, a heterogeneous database encryption algorithm based on the B/S structure is proposed. By comparing the advantages and disadvantages of the C/S structure and B/S structure, the framework of heterogeneous database encryption based on the B/S structure is constructed. Based on the encryption architecture framework of the B/S structure, the key management mechanism is given, and the key of the authority management center at all levels in the server user information management system is generated, as well as the user's key and the user's master key in the corresponding authority center. Combined with the access policy tree encryption to calculate the plaintext data on the server-side of the heterogeneous database, the data that need to be protected is converted into ciphertext form, and the heterogeneous database encryption is completed. The simulation results show that the proposed algorithm can effectively complete the heterogeneous database encryption and protect the database security, and the response time to each service of the database server is shorter than the encryption algorithm compared with the experiment.

Keywords: B/S structure; Heterogeneous database; Encryption; Data protection

1. Introduction

A heterogeneous database is a collection of several databases that are associated. It can realize data sharing and transparent access. Each database already exists and has autonomy before it is added to the heterogeneous database. Furthermore, it still has its application characteristics, integrity, and security control while sharing data through heterogeneous databases. Because of the wide use of heterogeneous databases, the number of database invasions, change, destruction, leakage, and other problems is also increasing. Therefore, protecting the high performance and high availability of database systems, improving data security, and ensuring that key data is not leaked have been highly valued by relevant experts [1][2].

Database security protection needs overall consideration of data integrity, confidentiality, and availability. Data protection techniques such as user authentication, authorization management,

ICIDC 2022, June 17-19, Qingdao, People's Republic of China

DOI 10.4108/eai.17-6-2022.2322707

and safety auditing are commonly used in current database management systems [3]. Nowadays, database encryption methods can be divided into three ways: in-library encryption, out-library encryption, and hardware encryption [4].

Aiming at the problem of data security protection of heterogeneous databases, a heterogeneous database encryption algorithm based on B/S structure is proposed, and the algorithm's performance is verified by simulation.

2. Database encryption architecture based on B/S structure

B/S stands for Browser/Sever. Clients can access all kinds of databases built in the server only through the browser in this structure. Under this structure, a user interface is accessed entirely through the browser, and the browser realizes the data interaction between the client and the database through the webserver. In the B/S structure, the primary transaction logic is implemented on the server-side. Compared with the C/S structure, the B/S structure does not require a private network, and it can run on a vast area network. It has a vast user base, better component reuse, a low system development and maintenance cost, and better information interaction. However, due to the broad coverage of WAN and the complexity and dispersion of customer groups, B/S has relatively weak security control ability. Security services such as user access control, data encryption, and identity authentication are required to ensure database security. The heterogeneous database encryption architecture based on B/S structure is shown in Figure 1

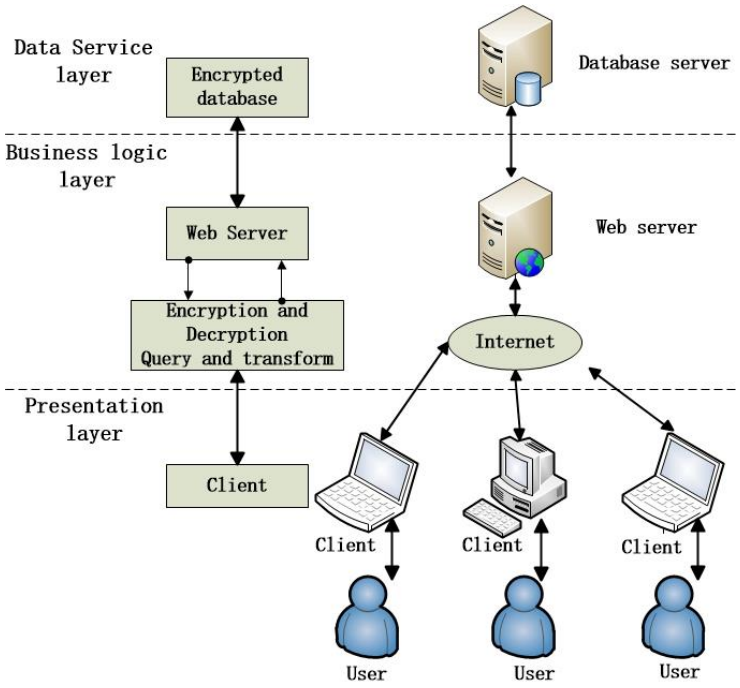


Figure 1 heterogeneous database encryption architecture based on B/S structure

According to Figure 1, after the user sends a command to the database server through the PC, the WEB server decrypts the required data in the encrypted database and feeds back the result through the client. The specific process follows (1) Client input instructions, account login, and user authority verification. (2) User access request and key input, and the database server determines the user request and queries the corresponding data content. (3) The WEB server determines the encryption key, encryption type, and data algorithm. The WEB server converts user instructions into operable instructions and transmits them to the database server to transfer query instructions. (4) The database server transmits the encrypted data to the ciphertext management system, identifying the security level and providing decryption feedback based on user permissions. (5) After the user completes the data access, the termination instruction is sent to the service network, and the data transmission is disconnected [5].

The heterogeneous database encryption system using the B/S structure has advantages: (1) The interaction model is more superficial. (2) Simplify the complexity of development and maintenance with lower costs. (3) Easier to operate.

3. Heterogeneous database encryption algorithm based on B/S structure

Through the above analysis, the heterogeneous database multi-level encryption and management based on the B/S encryption architecture is realized by the following steps: Heterogeneous database encryption calculation, generate a key management mechanism, calculation of the total user key, and set of the access permission and operation permission for the database.

3.1 Key management mechanism and key generation calculation

The core of database encryption is the encryption algorithm. The security of data depends on the security of the key. The premise of preventing a database system from being invaded is to ensure the security of the key. Ensuring the database system's security and data encryption efficiency is an essential task of key management. The main steps include key design, distribution, storage, use, change, and extinction.

Data encryption can protect the database to not being stolen by criminals and then cause significant loss. Encryption algorithms and key management are indispensable elements in an encryption system. The primary data encryption process converts the original plaintext data set files into unreadable codes of a certain length (ciphertext) after being processed by some algorithm [6]. The original content can be displayed only after entering the corresponding key.

The effect of database encryption is also closely related to key management. Accordingly, the key management mechanism is set, which mainly includes:

1) The length and content of the key must meet certain security levels.

2) The key must be replicable and match the corresponding encrypted data.

3) The key distribution process must be secure and confidential, and the key content should be replaced and reclaimed regularly.

Assume that the user information central authority management center of the encrypted database server is TA, and assigns the corresponding global user identity digital signature string to each user, denoted as GID,Assign global identifiers to each level of authority center AA,denoted as AID.During system initialization, the recursive depth of the user key structure is given through the central authority management center, denoted as depth. Then the maximum value of the multi-level permission center can be obtained by combining the corresponding relationship between all levels of permission center and user attribute subset [7].

Take $depth = 2$ as an example, select parameters α and $\{\beta_1, \beta_2\}$,the system public key PK_0 and master key MK_0 are expressed as follows:

$$PK_0 = \begin{Bmatrix} G_0, g : h_1 = g^{\beta_1}, f_1 = g^{\beta_1^{-1}}, \\ h_2 = g^{\beta_2}, f_2 = g^{\beta_2^{-1}}, e(g,g)^{\alpha} \end{Bmatrix} \tag{1}$$

$$MK_0 = \left\{ \beta_1, \beta_2, g^{\alpha} \right\} \tag{2}$$

Among them, G_0 is a bilinear group, g is the generator of G_0, e is A mapping on G_0. h_1, h_2, f_1 and f_2 are the Multiplication cyclic group of G_0.

The master key of the first-level permission center AA and lower-level permission center AA_{k+1} is:

$$MK = \left\{ \Lambda, D = g^{\frac{\alpha + r}{\beta_1}}, D_{i,j} = g^{r_{i,j}} \cdot H\left(a_{i,j}\right)^{r_{i,j}} \right\},$$
$$D'_{i,j} = g^{r_{i,j}}, E_i = g^{\frac{r + r_i}{\beta_i}} \tag{3}$$

$$MK_{k+1} = \left\{ \hat{\Lambda}, \hat{D} = D \cdot f_1^{\hat{r}}, \hat{D}_{i,j} = D_{i,j} \cdot g = g^{\hat{r}_i} \cdot H\left(\hat{a}_{i,j}\right)^{\hat{r}_{i,j}} \right\},$$
$$\hat{D}'_{i,j} = D'_{i,j} \cdot g^{\hat{r}_{i,j}}, \hat{E}_i = E_i \cdot f_2^{\hat{r}+\hat{r}_i} \tag{4}$$

Among them, Λ and $\hat{\Lambda}$ respectively represent the attribute set of the corresponding level permission center. r and $\hat{r}$ are respectively the representatives of the attribute sets Λ and

$\hat{\Lambda}$ randomly selected by the superior center. r_i and $\hat{r}_i$ represent the attributes A_i and $\hat{A}_i$ of the layer i of the corresponding center respectively, $r_{i,j}$ and $\hat{r}_{i,j}$ represent the attributes $a_{i,j}$ and $\hat{a}_{i,j}$ of the layer i of the corresponding center respectively, $0 \le i \le n$, $1 \le j \le m$; E_i and $\hat{E}_i$ are used to decrypt the corresponding central transformation node for cross-set matching of attributes; D, $D'_{i,j}$ and $\hat{D}$, $\hat{D}'_{i,j}$ are the identification identifiers of the corresponding central node before and after transformation respectively.

Assuming that the user information is managed directly by the user management system authority center at layer l of the server, all attributes A_u of any user are managed jointly by K authority centers on the authority management chain. The set of attributes of the user in the k-th($k \le K$) permission center $Au^{(k)} = \{Au_0^{(k)}, Au_1^{(k)}, \cdots, Au_n^{(k)}\}$,The authority center uses pseudo random function PSK, combined with GID and AID_k, to generate the private key component of the corresponding layer for the user $Au_{(k)} = PSK(u)$. The authority management center AA_k randomly selects $Ru^{(k)}$ to represent $Au^{(k)}$, $ru_i^{(k)}$ represent $Au_i^{(k)} \in Au^k$, $au_{i,j}^{(k)}$ represent $au_{i,j}^{(k)} \in Au_i^{(k)}$,Finally, the key of the user in the corresponding authority center is generated

$$\beta_{k,1} = \frac{\beta_1}{\alpha + r + \hat{r}} \tag{5}$$

$$\beta_{k,2} = \frac{\beta_2}{r + r_i + \hat{r} + \hat{r}_i} \tag{6}$$

Among them, $\beta_{k,1}$ and $\beta_{k,2}$ represent two groups of local master keys in the corresponding permission center. Finally, the total key expression of the user is as follows

$$SK_u = \left\{ \left\{ SK_u^{(k)} \right\}_{k=1}^{K}, D_{user} = g^{\alpha + \sum_w^W \alpha u_{(w)} \Big/ \sum_w^W \beta_{w,1}} \right\} \tag{7}$$

In this formula, D_{user} is the total user decryption key issued by the central authority for decryption.

3.2 Heterogeneous database encryption algorithm

The plaintext data should be encrypted for the data owner before being uploaded to the database server. According to the user access authority rules set by the WEB server authority management center and the comprehensive attribute set managed by each permission center, the user access policies are divided into several sub-policies. Set the number of the sub-policies to W , and each sub-policies corresponds to each authority center.

Assume that the policy tree set composed of all access sub-policies is $\{\mathrm{T}^{(\mathrm{w})}\}_{\mathrm{w}=1}^{\mathrm{W}}$,During data encryption, the data owner needs to select any random number θ first. Then, the ciphertext description after M conversion is

$$\tilde{C} = M \cdot e(g,g)^{a\theta} \tag{8}$$

Assume that the access strategy tree on the permission center AA_w is $T^{(w)}$, starting from the root node R of the strategy tree, each node $x^{(w)}$ after that has a corresponding polynomial q_x .For the non-leaf node in the strategy tree, the order d_x of the corresponding polynomial is 1 less than the change limit value k_x of node X, and the corresponding description is $d_x = k_x - 1$. For leaf nodes, the order of corresponding polynomial is 0. For any node $x^{(w)}$ other than the access policy root node, $q_x(O) = q_{parent(x)}(index(x))$ is satisfied, and other values of q_x can be randomly selected. For the root node of the access policy tree, if $q_R(O) = \theta$ is satisfied, the remaining items are randomly selected and the Lagrange polynomial is used to determine the threshold polynomial [8]. $Y^{(w)}$ and $X^{(w)}$ respectively represent the set of all leaf nodes $y^{(w)}$ and all non-leaf nodes $x^{(w)}$ in the strategy tree. Similarly, a similar encryption process is performed in the remaining $W-1$ rights management centers, and the ciphertext and access policy of plaintext data M is finally obtained.

$$C = \left\{ \left\{ T^{(w)} \right\}_{w=1}^{W}, \tilde{C} = M \cdot e\left(g,g\right)^{\alpha\theta}, \left\{ CT^{(w)} \right\}_{w=1}^{W} \right\} \tag{9}$$

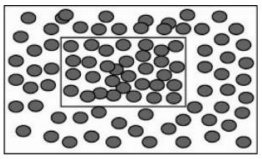
The ciphertext can be decrypted through encryption reverse operation, converted into plaintext again, and then fed back through the server-client for the encrypted heterogeneous database content. In obtaining database content and decrypting ciphertext, according to the differences in

user levels, the corresponding database content obtaining permission and operation permission is set through the User information permission management center of the WEB server, realizing multi-level encryption and management of the heterogeneous database.

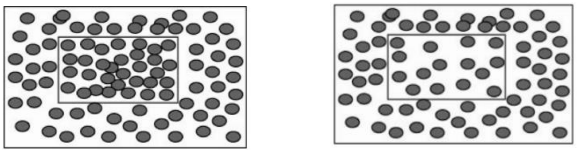
4. Results analysis

In order to verify the performance of the proposed heterogeneous database encryption algorithm based on the B/S structure, a simulation is needed.

The algorithm in this paper and the database encryption algorithm of association operation are used for experimental comparison, and the loss of encrypted data is used to measure the data encryption effect. The comparison results are shown in Figure 2



(a) Original database



(b)algorithm in this paper (c) algorithm based on association operation

Figure 2 Loss of encrypted data with different algorithms

According to Figure 3, in encrypting the same database, the algorithm in this paper not only plays a particular encryption role in the database but also ensures no loss of data in the encrypted database. However, when using the Associative operation database encryption algorithm encrypts the database, the data loss is significant in the encryption process, indicating that the algorithm in this paper can effectively reduce the data loss in the cloud storage platform.

Take data queries and data updates as examples. The experimental results are shown in Figure 4. BS represents the proposed algorithm; MA represents the database encryption algorithm based on association operation; MB represents the database encryption algorithm based on Arnold transform. The processing time is calculated as follows.

$$T_c = ent(T_f / N_c) \tag{10}$$

T_f represents the real signal characteristic quantity, and N_c represents the query and update traffic.

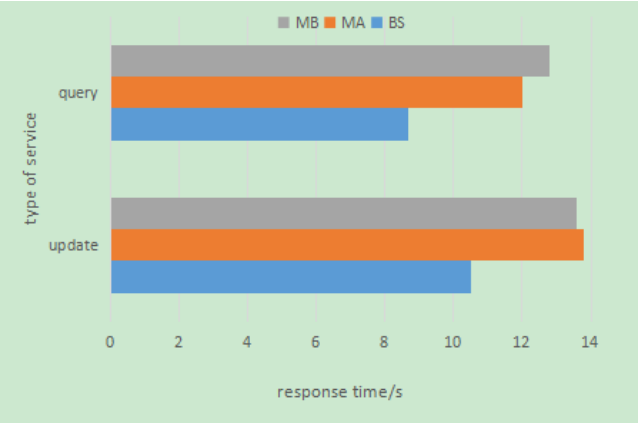


Figure 3 Business processing time of the three algorithms

According to the figure above, for data query and data update, among the three algorithms, the business response time of the proposed algorithm is shorter, indicating that the proposed algorithm has higher operating efficiency and its performance is superior to the other two algorithms compared in the experiment.

5. Conclusion

Encryption protects information security by hiding plaintext information and making it unreadable when special information is incomplete. Aiming at the problem of a long time of heterogeneous database encryption caused by uneven key correspondence between users and authority center in the traditional method, a heterogeneous database encryption algorithm based on B/S structure is proposed. The results show that the proposed algorithm can effectively complete the encryption and decryption process of the heterogeneous database with high efficiency and good security protection.

References

[1] Li Aining, Ji Qingchang. Encryption and Optimization of Database Access Information Transmission [J], Computer Simulation. 2018, 35(2): 135-138. (in chinese).

[2] Ma Ruchao, Zhao Liang. Data encryption technology for coal mine safety monitoring system[J], Industry and Mine Automation, 2017,43(2):15-18.

[3] Sun Yanjun, Yang Geng, Shi Jingqi. User - definable order - preserving encoding with low adjustment ratio [J], Computer Engineering and Applications,2018,54(9):67-74.

[4] Li Zichen, Yang Wei, Yang Yatao. Design of Homomorphic Cloud Platform Based on Onion Encryption Model, Computer Engineering[J],2018,44(8):30-35.

[5] Yan Xixi, Hu Qianwei, Yang Yongli, Key management schemes based on access control and Chinese remainder theorem in database [J], Computer Engineering & Science, 2017,39(8): 1457-1464.

[6] Jiang Bingcheng, He Qian, Chen Yiting. Cloud database oriented attribute based encryption and query translation middleware [J], Journal of Computer Applications,2018,38(8):146-152.
[7] Jiang Xue, Tan Wentao, Ma Shilong. Health management system of mine main ventilator based on B/S structure [J], Coal Technology, 2018, 37(11):252-253.
[8] Yang Haowei, Chen Gouxi, Han Tong. Scope of application of homomorphic encryption algorithm and improvement of efficiency and application [J], Computer Engineering and Design, 2017, 38(2):318-322.