Design of Encryption System for Urban Cultural Heritage Protection Database Based on Blockchain Technology

Cai-ting Peng, He-qing Zhang(✉), and Yi-lin Feng

Management (Tourism), School of Guangzhou University, Guangzhou 510006, China

Abstract. Aiming at the problems of long data encryption time and low encryption efficiency in the traditional urban cultural heritage protection database encryption system, an urban cultural heritage protection database encryption system based on blockchain technology is designed. In terms of hardware design, database encryption services were added, and database encryption system hardware was deployed. In terms of software design, based on hardware deployment, design the system application architecture; design system interfaces from the perspectives of encryption and decryption; generate and manage system encryption keys; use blockchain technology to reduce encryption principles. Language composition, adding symbols and terminology, data encryption structure, designing a mixed encryption and decryption flowchart, and realizing the encryption of the urban cultural heritage protection database. The experimental results show that under different tuple pairs, the average time of Setup data encryption is 38 s and 41 s less, and the average dynamic update time consumption is less than 5.77 s, 10.48 s. Under the condition of different number of keywords, Setup data encryption The average time is shorter by 34 s and 96 s, and the average dynamic update time is shorter by 45.19 s and 98.72 s. Has a high encryption database efficiency.

Keywords: Blockchain technology · Urban culture · Heritage protection · Database encryption

1 Introduction

Since the reform and opening up, China's economy has shown an unprecedented high-speed development trend, and the urban population has become larger and larger, followed by "old city reconstruction", real estate development, infrastructure reconstruction, etc., which has led to the serious destruction of many historical and cultural heritage. At the same time, with the improvement of people's material living standards, people's eager desire for cultural needs has transformed into a growing nostalgia complex. It is this complex that directly stimulates the economy that leads to the phenomenon of "antique" replacing the real ancient. City is not only the carrier of history and culture, but also the cultural landscape of social economy. To maintain the continuity of urban landscape, to protect the local characteristics of local architecture and to retain the memory of street space are the needs of the development of modern human civilization [1].

S. Liu and X. Ma (Eds.): ADHIP 2021, LNICST 416, pp. 627–644, 2022.
https://doi.org/10.1007/978-3-030-94551-0_49

Nowadays, social network technology has developed rapidly and has been widely used in all walks of life. Everyone obtains the information he needs on the Internet, and information has gradually become an important resource in the online world. Based on this, the protection method of urban cultural heritage data has also changed from traditional paper, photo and other protection methods to network technology protection, which stores all urban cultural heritage data in the heritage database [2]. However, with the continuous development of the network, various network attacks have also occurred. Therefore, the security of the urban cultural heritage protection database is highly valued by people. The core of network security is database security, so the discussion on database security is of great significance.

In the database management system, the administrator has great authority, which can not only manage what the system wants to do, but also query any information he wants to know. As long as the database administrator wants to know the user's information, he can do it. In addition to hacker attacks, part of cybercrime comes from the inside of the system. Since internal attacks on the database are simpler than hackers, in order to prevent insiders from unauthorised querying of data to view the urban cultural heritage protection data, and record them, Under this kind of situation, the information leaked out makes the protection of urban cultural heritage more difficult [3]. In order to avoid this problem, the urban cultural heritage protection data can be encrypted. After the information is encrypted, the urban cultural heritage protection data can be decrypted by the private key of the relevant personnel. In this way, even if the management personnel or hackers obtain the information, they can not decrypt the data and make the act of leaking the urban cultural heritage protection data, In this way, the confidentiality of urban cultural heritage protection data is guaranteed.

At home and abroad, database encryption technologies such as time control encryption technology, timed release proxy re-encryption scheme, specific time encryption, TSE scheme, TRE scheme, information security timed release secret sharing scheme, and access control model have been developed to solve malicious intrusion by hackers, Database security issues such as stealing and data leakage [4]. On the basis of domestic and foreign research, literature [5] chooses the way of encrypting the data in the database, so that important data information is stored in the form of ciphertext in the database. Even if the database is attacked, it can effectively ensure that the data will not be leaked, so as to ensure the security of the data in the urban cultural heritage protection database. Literature [6] uses a symmetric encryption algorithm to improve the efficiency of multi-keyword encryption, combines blockchain technology to solve the problem of dishonest search in cloud servers, and uses a linear index structure to achieve multi-keyword search while improving search efficiency. Blockchain is proposed. A searchable encryption scheme that supports multiple keywords on the Internet to ensure the security of data in the urban cultural heritage protection database.

Based on the above research, this research uses blockchain technology to encrypt the data in the urban cultural heritage protection database, through the client PC, based on the browser, remote configuration management system hardware, and from the encryption and decryption two In terms of designing the system interface, ensuring the security of the database encryption system through the system encryption key, using blockchain technology to realize the encryption of the urban cultural heritage protection database,

enhancing the encryption performance of the urban cultural heritage protection database encryption system, and proposing a blockchain-based technology The design of the encryption system of the urban cultural heritage protection database.

2 Deploy the Hardware of the Database Encryption System for the Protection of Urban Cultural Heritage

In this study, considering the data of urban cultural heritage protection and the demand for database encryption system, the database encryption service is added on the basis of the hardware of database encryption system determined by predecessors. Through the PC of the client, based on the browser, the database encryption system hardware is remotely configured and managed, as shown in Fig. 1.

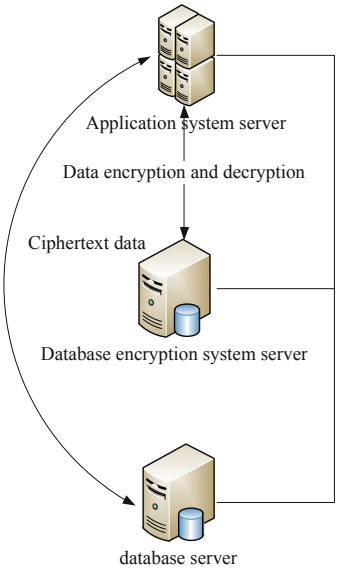

Fig. 1. Database encryption system deployment

The database encryption system deployment method shown in Fig. 1, the selected database encryption server is a high-performance dedicated all-in-one server, and the key management system of the data encryption system, the main control service system and the management control platform are deployed inside.

In Fig. 1, the database encryption system is located between the application system and the application system database. The specific implementation of data encryption and decryption is carried out between the application system server and the database encryption system server, completely outside the application system database, so it is transparent to the application system database, and will not affect the original architecture and performance of the application system.

3 Software Design of the Encryption System for Urban Cultural Heritage Protection Database Based on Blockchain Technology

3.1 Design System Architecture

Based on the deployment mode of the database encryption system shown in Fig. 1, the application architecture of the designed database encryption system is shown in Fig. 2.

As shown in Fig. 2, the system application architecture diagram includes three parts: user's database application system, database encryption server and user's application system database.

Among them, the user's application system is the same as an ordinary database application system. It can be a client in a C/S structure, a browser in a B/S structure, or an independent database application. It is just a client who deploys a database encryption server. interface. The database encryption server is the core part of the whole database encryption system. It provides security services to the user application system, receives encryption and decryption requests from the user application system, and performs encryption and decryption of database information, key management, etc. The application system database can be all kinds of relational data, such as Oracle, DB2, mysql, MS SQL server, etc. the database encryption server is transparent to the application system database.

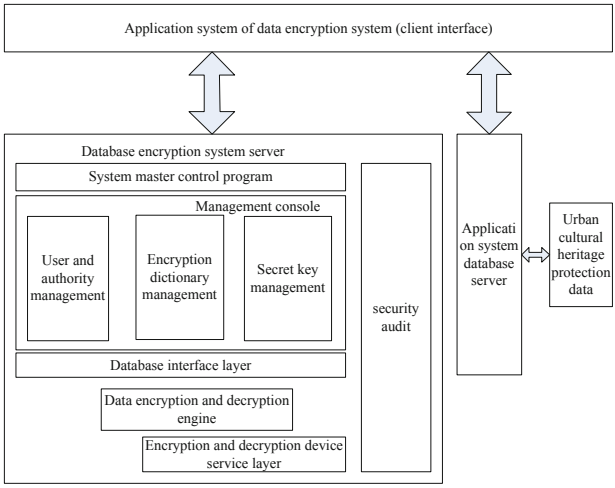

Fig. 2. System application architecture diagram

The data encryption and decryption processing is completed between the application system and the database encryption server, and the database encryption server is not directly related to the application system database. In this way, it is compatible with the application system database to the greatest extent, and does not need to change the user's logic of operating the original database. At the same time, for application developers, they can use various ways of operating the application system database (such as JDBC, ODBC, hibernate, special database operation language).

From a security perspective, the database security architecture of this study uses a four-level security layer to achieve the security of the entire system, namely the outer layer, the middle layer, the inner layer and the analysis layer. The function of the outer layer is to verify the legitimacy of the user's identity. Only having a legal login account (together with password) can access the database encryption server and use the database encryption server for encryption and decryption; The middle layer is a restrictive management security measure based on RBAC policy. Its function is to verify the user's rights and ensure that legitimate users can only encrypt and decrypt the data within their rights. The inner layer is a data encryption layer that includes a data encryption and decryption engine and key management. This security measure converts the original data into an encrypted form that is not directly readable, and the confidentiality, tamper-proof and reliability of the data are guaranteed. The analysis layer is a security analysis layer including security audit control. Since it is impossible to build data that is not attacked, the security system must provide a mechanism to detect illegal access to the database. In the four security layers, the outer layer, middle layer and inner layer provide active protection for database, while the analysis layer provides passive protection for database.

3.2 System Interface Design

When providing the interface to the application system, we can consider using static global class to save some global setting information and authentication status of users, and consider caching mechanism and timing mechanism to ensure the efficiency of authentication and verification. In addition, the connection pool can be used to improve the efficiency of the system, and the reserved interface can be designed for intelligent automatic processing of user SQL requests.

Encrypted Interface

Parameters to be provided by the application system ("database name, table name, field name", "data to be encrypted"); The interface first determines whether it has been authenticated. If the system automatically obtains the hidden parameters (user ID, password), it connects to the server for identity authentication. After the authentication is passed, the system needs to set the authentication ID [7]. If the authentication is passed, the encryption process with the server is automatically completed, and the encryption result is obtained and returned to the caller. The interface description is shown in Table 1.

Table 1. Data encryption interface description

Serial number	Encrypted representation	Paraphrase
1	Prototype	String db-Encrypt (string strKeyHash, byte[] byPlainText)
2	Description	Encrypt the data and return the encrypted data after Base64 encoding

(*continued*)

Table 1. (*continued*)

Serial number	Encrypted representation	Paraphrase
3	Parameter	StrKeyHash [in]: Data belongs to library name, table name, field name byPlainText [in]: Plaintext data (hexadecimal string)
4	Return value	base64Encrypted data after encoding: success Null: fail

As shown in the data encryption interface description in Table 1, the data encryption interface access system process is as follows: 1. Import the user ID and encrypted password, and configure the file by the client; 2. The background automatically establishes a connection with the server and completes the identity Authentication; 3. Determine whether the plaintext data is greater than the maximum number of bytes configured in the system;4. When the plaintext data is less than the maximum number of bytes configured by the system, the function is used to encrypt the data. When the plaintext data is greater than the maximum number of bytes allowed, the user is prompted to convert the packet and encrypt it in sections; 5. Automatically release the connection and clear the authentication mark after completion.

Decryption Interface

The parameters (database name, table name, field name, data to be solved) that the application program needs to provide; The interface first determines whether it has been authenticated. If not, the system automatically submits the implicit parameters (user ID, password) to the server for authentication. After the authentication is passed, the system needs to set the authentication ID [8]. If the authentication has been passed, the decryption process with the server is automatically completed, and the decryption result is obtained and returned to the caller. The interface description is shown in Table 2.

Table 2. Description of data decryption interface

Serial number	Encrypted representation	Paraphrase
1	Prototype	Byte[] db_Decrypt (string strKeyHash, string strEncryptData)
2	Description	Decrypt the data and return the decrypted data (plaintext)
3	Parameter	strTablename [in]: Data belongs to library name, table name, field name strEncryptData [in]: Ciphertext data (base64 encoding)
4	Return value	Data decryption: successful Null: fail

As shown in Table 2, the process of data decryption interface accessing the system is as follows: 1. Import the user ID and encrypted password, and configure the file by the client; 2. The background automatically establishes the connection with the server and completes the identity authentication; 3. Determine whether the encrypted data is greater than the maximum number of bytes configured by the system; 4. When the data is less than the maximum number of bytes configured by the system, use this function to decrypt the data (automatically break the sub-packages of suitable size and return to the caller automatically Group package); 5. After completion, the connection will be automatically released and the certification mark will be cleared.

3.3 Encryption Key Management

The management of data encryption keys directly affects the security of the database encryption system, so key management occupies a very important position in the database encryption system. For this reason, the encryption key management module of the system that conforms to this design is studied.

Generate Encryption Key

To generate the encryption key of the urban cultural heritage protection database encryption system this time, a user-specific data will be selected to participate in the hash operation to generate the data encryption key, so this design uses the following method to generate the data encryption key:

$$D = H(u, p, s, r) \tag{1}$$

D represents the last data encryption key; H represents hash algorithm; u is the user name of the user; p is the user's private key; s is the name of the field to be encrypted; r represents the random number generated by the system for different table structures [9].

Using this data encryption key generation method can make the same user generate different keys for the same-named fields in different table structures, which increases the difficulty of key cracking.In addition, the security of this key generation method completely depends on the security of the user's password. Even if the attacker can get the user's database password, only the user's corresponding urban cultural heritage protection data will be affected. The attacker cannot get the encryption key of other users' corresponding urban cultural heritage protection data through the attacked user's encryption key.

Design Encryption Key Management

When the database administrator creates a table, it will store the field information that needs to be encrypted in the table in kmdb, including the encryption algorithm used by the field and a random number generated when the table is generated.When the user stores data in the table, DECO's encryption judgment module will determine that this operation needs to be encrypted based on the information in the KMDB. At this time, DECO calls the key management module to generate the final data encryption key based on the user's user name, password, field name, and a random number generated by the key management module.Then, when the user reads the data in the encrypted field,

DECC obtains the corresponding encryption algorithm and random number from kmdb through the same steps, generates the data encryption key according to the user name and password, decrypts the field, and finally returns the information to the user. The process is shown in Fig. 3.

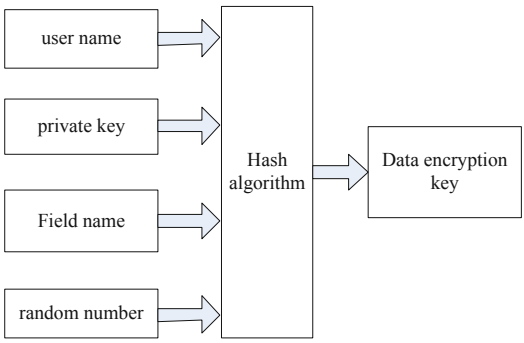

Fig. 3. Data encryption key generation process

3.4 Basic Representation of Data Encryption Based on Blockchain Technology

Determine the Basic Encryption Primitives of the System

Based on the encryption key generated by formula (1), the user's private key encryption scheme is composed of three polynomial time algorithms

$$S = (G, E, D) \tag{2}$$

(2) In the formula, S represents a time algorithm; G represents a conceptual algorithm, which requires a security parameter k and returns a key K; E also represents a probability algorithm, which requires a key K and a message m as parameters, and returns the ciphertext c; D represents a deterministic algorithm, enter the key K (if K is the key to generate c) and the cipher text c and return m [10].

Informally, if the output of the ciphertext does not reveal any part of the plaintext information to the adversary, the private key encryption scheme is secure against chosen plaintext attack. Let's suppose that a scheme outputs ciphertexts that are computationally indistinguishable, And from random to adversary, it can query Encryption o adaptively. It is said that this scheme is safe for random ciphertext under the selected plaintext attack. In addition, the pseudo-random function P is a polynomial-time computable function, and it cannot be distinguished from a random function in polynomial time with any probability.

Basic Symbols and Terms of the System

The set of all binary strings of length n is denoted as $\{0, 1\}^n$, and the set of all binary strings of finite length is denoted as $\{0, 1\}^*$. And $[n]$ represents the integer set $\{1, \cdots, n\}$, and $2^{[n]}$ represents the corresponding power set. And $x \leftarrow X$ indicates that the element x

is sampled from the distribution X, and $x \leftarrow A$ indicates that the output of the algorithm A is x. Given a sequence o of n elements, the i element is denoted as o_i or $o[i]$. If T is a set, then $\#T$ is its base. In addition, $\langle x, y \rangle$ or xy represents the concatenation of the character strings x and y.

d is the plaintext structure, while e is the ciphertext structure constructed by the client using K as the key. If the scheme is stateful, the setup program will also output state st. Then, the client sends the encrypted structure e to the untrusted server and keeps the state st and the key K private. Among them, the client can use supported query operations on e.

Data Structure

The design of the urban cultural heritage protection database encryption system is adjusted to the original data decryption structure. The blockchain will generate a block Merkle tree based on the on-chain data, and generate the block header during the consensus process, redesigned as Fig. 4 shows the block data structure.

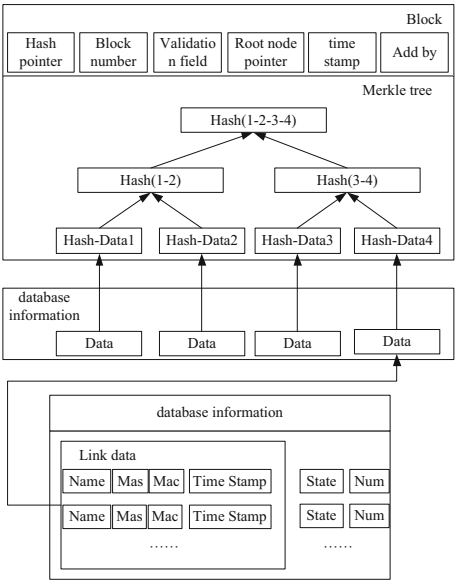

Fig. 4. Block data structure

In Fig. 4, the hash pointer is a pointer recorded with a hash value. Pointing to the predecessor block of the block, the entire block chain can be traversed through the hash pointer. The verification field of the predecessor block is stored. During the traversal process, the verification field of the block can be checked to determine whether the block is attacked.

The block number records the number of the block in the whole network.

The root node pointer is a pointer to the root node of the Merkle tree in this block, which is used to find the root node in the search process. Save the hash value of all root nodes.

The verification field is generated by hash pointer and root node pointer through specific hash function. In this way, all the blocks are concatenated with hash values from the beginning to the end.

The timestamp is used to record the adding time of the block, which is generated locally by each node, and small differences are allowed in the network. However, the timestamp of a block is not allowed to be earlier than the predecessor node.

The adder tag is used to record the adder of a block. When a node successfully applies to add a block, its public key is recorded on the block as a token.

The Merkle tree node is similar to the most basic tree structure. When the Merkle tree node points to the pointer of the child node, the pointer of the leaf node is empty. Used to store the hash value calculated by the child node or the content of the Mas field of the message, which is used to solidify the data.

3.5 System Encryption Process

The database encryption system designed this time will meet the following functions in actual use:

1. The storage period of the data in the database is relatively long. When using the encryption algorithm, the attacker should not be able to crack the algorithm in a short time and then attack the database.
2. After the data in the database is encrypted, the storage space it occupies cannot be clearly larger than that when it is not encrypted. The speed of data encryption and decryption must be fast enough to be transparent to the user, and the user should not be clearly aware of the encryption and decryption. The resulting delay operation.
3. The database encryption system has flexible authorization mechanism and reliable key management function.
4. After the database is encrypted, try not to affect the original functions, and at the same time, allow users to access the database at the level of their own authorization.

Based on this, a hybrid encryption algorithm is used to encrypt the urban cultural heritage protection database. The flow chart of hybrid encryption and decryption is shown in Fig. 5.

As shown in Fig. 5, the user encrypts the urban cultural heritage protection data to be stored with an encryption algorithm to generate the key K, and then obtains the public key K_1 generated by the database from the key server, Encrypt K with K_1 to get the encryption key K_2, and then the user will need to store the urban cultural heritage protection data, send the ciphertext and the encryption key K_2 to the database, The database uses its own private key to unlock the encryption key K_2 to get the encryption key K_1, and then uses K_1 to unlock the ciphertext of the stored urban cultural heritage protection data to get the plaintext.

4 System Test

Two groups of conventional database encryption systems were selected. By means of comparative test, the urban cultural heritage protection database was used as the experimental object of comparative test. The data was drawn into a chart by MATLAB software

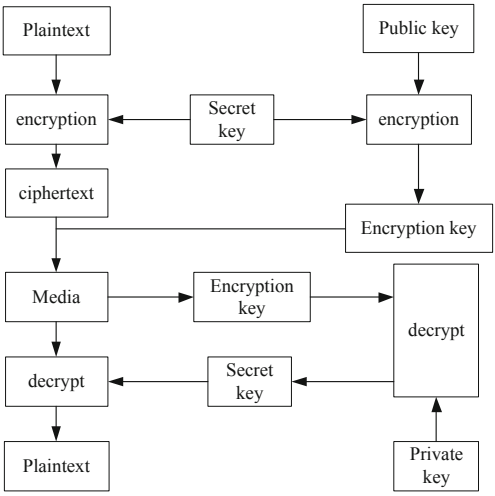

Fig. 5. Flow chart of hybrid encryption and decryption

to verify the encryption system of the research. Compare the data encryption time of three groups of database encryption systems for the same data.

4.1 Experimental Preparation

In order to avoid the failure of the three groups of system software and hardware selected in this experiment, which will affect the system test results, the various hardware circuits in the three groups of systems will be tested on the PCB board; for the system software, the build as shown in Fig. 6 System software testing environment, testing system software.

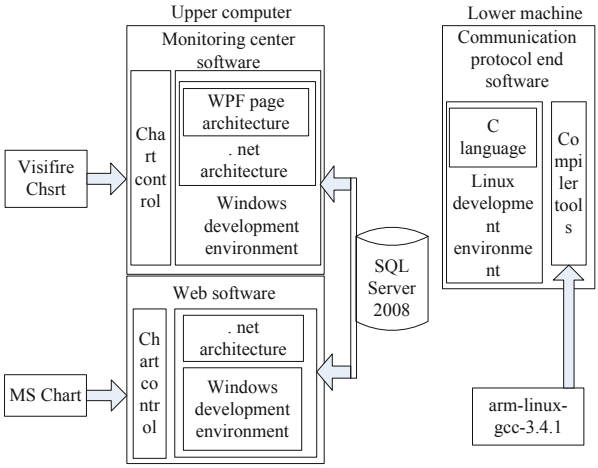

Fig. 6. System software testing environment

The system software and hardware testing process is as follows:

1. Using PCB circuit board, check whether there are circuit faults, short circuit, loose welding, open circuit, load, poor contact and other problems in the system circuit. 2. Check whether the three groups of system hardware equipment are complete and whether the detection equipment is in normal working condition.3. When there is no problem with the system hardware, the operating system and system communication protocol are embedded in the system software operating environment shown in Fig. 6. 4. Separately test all the drivers in the system, focusing on the system encryption program; 5. Use the database encryption module to control the system to encrypt the urban cultural heritage protection database, and check whether the encryption module is operating normally;6. When it is confirmed that the encryption module of the system is in normal operation, Download all the programs of the system to the corresponding chips of the system.

In this experiment, the system test hardware environment set up is shown in Fig. 7.

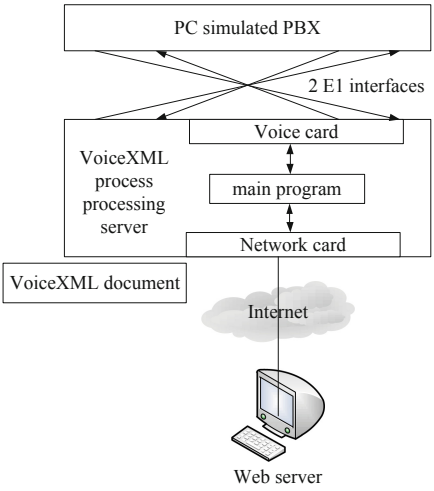

Fig. 7. System test environment

In this experiment, the selected urban cultural heritage protection database, in txt format, stores a total of about 500,000 urban cultural heritage data. Using each urban cultural heritage data as an edge, 36,692 points and 183,831 edges can be obtained. The data set size is about 1.4G.

According to the database selected for this experiment, the experimental data is divided as shown in Table 3, so that the number of (W, ID) pairs in the database is from $1 * 10^6 - 20 * 10^6$, and the setup time, search voucher generation time, query time and dynamic update time are tested in different numbers, Test each quantity 10 times, take the average value, and test a total of 10 different quantities.

Table 3. Pairs logarithm and number of keywords in the data set

Pairs (10^6)	Keyword	Pairs (10^6)	Keyword
1	1108	2	5680
3	7650	4	9320
5	18653	6	30492
7	51116	8	61255
9	72594	10	76594
11	91254	12	179830
13	210346	14	257364
15	307812	16	353155
17	395713	18	434379
19	496193	20	500000

In Table 3, pairs (106) represents tuple number pairs after database data partition; Keyword is the number of keywords. It is divided into 1–20 blocks according to the key value logarithm, and the key value logarithm of each block is $i * 10^6$, where i represents the block.This paper uses this data set as the algorithm verification data set, and has carried out the setup data encryption time, token formation time, query time, update time and other perspectives to verify the efficiency of the system encryption database.

4.2 Experimental Results

The First Set of Experimental Results

The setup data encryption time includes the structure of security index and data encryption operation. Experiments are carried out under the pairs logarithm and the number of keywords of the dataset shown in Table 3, in which the number of different inverted index key value pairs and the number of unique keywords are taken as variables respectively, Comparing the Setup data encryption time of the three schemes, the experimental results are shown in Table 3 and Table 4.

Table 4. Comparison of setup data encryption time under different tuple numbers (s)

Pairs (10^6)	Design system	Conventional system 1	Conventional system 2
1	5	43	46
2	10	48	51
3	13	51	54

(*continued*)

Table 4. (*continued*)

Pairs (10^6)	Design system	Conventional system 1	Conventional system 2
4	20	58	61
5	25	63	66
6	30	68	71
7	35	73	76
8	40	78	81
9	45	83	86
10	50	88	91
11	55	93	96
12	60	98	101
13	65	103	106
14	70	108	111
15	75	113	116
16	80	118	121
17	85	123	126
18	90	128	131
19	95	133	136
20	100	138	141

It can be seen from Table 4 that in different tuple number pairs of three groups of database encryption systems, the encryption time of setup data increases linearly with the increase of data volume, and basically maintains a positive relationship with non duplicate tuple data pairs. Among them, the conventional system 2 encrypts the tuple pair data in the urban cultural heritage protection database, and the average time of Setup data encryption is 93.4 s, which takes the longest time; the conventional system 1 encrypts the tuple pair data in the urban cultural heritage protection database, The average time of Setup data encryption is 90.4 s; The design system encrypts the tuple number pair data in the urban cultural heritage protection database. The average time of setup data encryption is 52.4 s, which is 38 s and 41 s less than that of the two conventional systems.

As can be seen from Table 5, the experimental comparison under different number of keywords shows that the data encryption time of setup also shows a linear growth with the increase of the amount of data, and basically maintains a positive relationship with the number of non duplicate keywords.Among them, the conventional system 2 encrypts the tuple pair data in the urban cultural heritage protection database, and the average time of Setup data encryption is 117 s, which takes the longest time; the conventional system 1 encrypts the tuple pair data in the urban cultural heritage protection database, Setup data encryption average time is 55 s; The design system encrypts the tuple number

Table 5. Comparison table of Setup data encryption time under different number of keywords (s)

Keyword	Design system	Conventional system 1	Conventional system 2
1108	2	36	98
5680	4	38	100
7650	6	40	102
9320	8	42	104
18653	10	44	106
30492	12	46	108
51116	14	48	110
61255	16	50	112
72594	18	52	114
76594	20	54	116
91254	22	56	118
179830	24	58	120
210346	26	60	122
257364	28	62	124
307812	30	64	126
353155	32	66	128
395713	34	68	130
434379	36	70	132
496193	38	72	134
500000	40	74	136

of data in the urban cultural heritage protection database. The average time of the setup data encryption is 21 s, which is 34 S and 96 s less than the average time required by the two groups of conventional systems.

It can be seen that in this design system, the time efficiency of index construction and data encryption module has been significantly improved compared with other solutions. The system designed this time has a high encryption database efficiency.

Results of the Second Group

The urban cultural heritage protection database encryption system, the stored data will also be adjusted with the discovery of cultural heritage, and there will be deletion changes, which involve the dynamic update operation of database encryption, so the elements shown in Table 3 Group data pairs and keywords, as dynamic update time verification data, Verify three groups of systems, and dynamically update the time consumption when encrypting the urban cultural heritage protection database. The experimental results are shown in Table 6 and Table 7.

Table 6. Comparison table of dynamic update time consumption under different tuple number pairs (s)

Pairs (10^6)	Design system	Conventional system 1	Conventional system 2
10	0.05	5.82	10.53
20	0.10	5.87	10.58
30	0.03	5.8	10.51
40	0.02	5.79	10.5
50	0.25	6.02	10.73
60	0.30	6.07	10.78
70	0.35	6.12	10.83
80	0.04	5.81	10.52
90	0.45	6.22	10.93
100	0.05	5.82	10.53
110	0.55	6.32	11.03
120	0.06	5.83	10.54
130	0.65	6.42	11.13
140	0.07	5.84	10.55
150	0.75	6.52	11.23
160	0.08	5.85	10.56
170	0.85	6.62	11.33
180	0.09	5.86	10.57
190	0.95	6.72	11.43
200	0.97	6.74	11.45

It can be seen from Table 6 that in different pairs of tuples, the dynamic update time of conventional system 2 fluctuates between 10–11 s, and its average dynamic update time consumption is 10.813 s, and the resulting dynamic update time consumption is relatively large.; Conventional system 1 dynamic update time consumption fluctuates between 4–7 s, its average dynamic update time consumption is 6.103 s, the dynamic update time consumption generated is too large; The dynamic update time consumption of the design system fluctuates between 0 and 1 s, and the average dynamic update time consumption is 0.333, which is 5.77 s and 10.48 s less than the two conventional systems respectively.

Table 7. Comparison of dynamic update time consumption under different number of keywords (s)

Keyword	Design system	Conventional system 1	Conventional system 2
1108	80.14	125.33	178.86
5680	98.79	143.98	197.51
7650	98.96	144.15	197.68
9320	99.99	145.18	198.71
18653	97.18	142.37	195.90
30492	96.47	141.66	195.19
51116	95.31	140.50	194.03
61255	97.81	143.00	196.53
72594	90.16	135.35	188.88
76594	95.41	140.60	194.13
91254	97.16	142.35	195.88
179830	85.41	130.60	184.13
210346	83.14	128.33	181.86
257364	89.16	134.35	187.88
307812	94.31	139.50	193.03
353155	96.71	141.90	195.43
395713	87.46	132.65	186.18
434379	85.31	130.50	184.03
496193	89.77	134.96	188.49
500000	94.56	139.75	193.28

It can be seen from Table 7 that the dynamic update time consumption of conventional system 2 fluctuates between 170 and 200 s, with an average dynamic update time consumption of 191.38 s, resulting in a large dynamic update time consumption; The dynamic update time consumption of conventional system 1 fluctuates between 120–150 s, and its average dynamic update time consumption is 137.85 s, and the generated dynamic update time consumption is too large;The time consumption of dynamic update of the design system is between 80 and 100 s, and the average dynamic update time consumption is 92.66, which is 45.19 s and 98.72 s less than the two groups of conventional systems.

It can be seen that the system designed this time encrypts the urban cultural heritage protection database when updating data, and still has a high encryption database efficiency.

5 Conclusion

In this research, an encryption system is designed on the server side, which is the periphery of the database, to ensure the security of the database and encrypt the important information in the database. Therefore, this design makes full use of blockchain technology to design a data encryption structure, and realizes the design of various functional modules of the encryption system through both system hardware and software, and completes the design of an urban cultural heritage protection database encryption system based on blockchain technology. Through experiments, the encryption efficiency of the system designed this time was verified, and the problems existing in the traditional system were solved.

Fund Projects. National Planning Office of Philosophy and Social Science Foundation of China: Research on Cultural Heritage Conservation and the Activation of South China Historical Trail. (NO:19FSHB007).

References

1. Shufen, N., Yaya, X., Pingping, Y., et al.: Cloud-assisted attribute-based searchable encryption scheme on blockchain. J. Comput. Res. Dev. **58**(4), 811–821 (2021)
2. Bingsheng, D.I.N.G.: Research on database encryption technology. J. BeiBu Gulf Univ. **35**(2), 46–51 (2020)
3. 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)
4. Liu, S., Li, Z., Zhang, Y., Cheng, X.: Introduction of key problems in long-distance learning and training. Mob. Networks Appl. **24**(1), 1–4 (2018). https://doi.org/10.1007/s11036-018-1136-6
5. Shuai Liu, Guanglu Sun, Weina Fu. e-Learning, e-Education, and Online Training, pp. 1–386. Springer, Cham. Doi:https://doi.org/10.1007/978-3-030-63952-5
6. Ma, Y.: Design of inter satellite communication network security encryption control system based on blockchain. Comput. Measur. Control **29**(3), 171–175 (2021)
7. Nie, M., Pang, X., Chen, W., et al.: Fair searchable encryption scheme based on ethereum blockchain. Comput. Eng. Appl. **56**(4) (2020)
8. Wei, Y.: Research on encryption optimization of data base access information transmission. Modern Ind. Econ. Informationization **9**(4), 65–66 (2019)
9. Liu, T., Chen, Q., Zhang, Y.: Research on data encryption technology for computer network security protection. Modern Inf. Technol. **3**(15), 153–154, 157 (2019)
10. Xianlong, T.: Design and implementation of encryption module for geographic information field acquisition system. Land Resour. Herald **16**(3), 31–33 (2019)