Intelligent Encrypted Storage Method for Medical Health Database Based on Internet of Things

Qing-bang Zeng[1](✉) and Wen-da Xie[2]

[1] College of Information Engineering, Jiangmen Polytechnic, Jiangmen 529000, China
honda455@163.com

[2] Computer Engineering Technical College (Artificial Intelligence College), Guangdong Polytechnic of Science and Technology, Zhuhai 519090, China

Abstract. By analyzing the process of encrypting and decrypting, the traditional method of encrypting and storing medical health database is designed, and the data in database is transformed slowly, which leads to high cost and long time of encrypting. The intelligent encryption of medical health database is divided into three parts: encryption algorithm, decryption algorithm and matching algorithm, and the key family, key cycle and key management mode of the three parts are determined to design the intelligent encryption algorithm for medical health database; the relationship between the encryption and decryption process and user read-write operation is analyzed to design the database encryption and decryption processing mode and transform database data; the cloud computing technology in the Internet of Things technology is adopted to design the database encryption scheme based on the Internet of Things technology to realize the intelligent encryption algorithm for medical health database. The experimental results show that the size of encrypted and unencrypted data is 7% and 4%, and the average ratio of key management time to data processing time is 0.274% and 0.298%, respectively, in one operation cycle, with less space overhead, faster database encryption speed and better encryption performance.

Keywords: Internet of Things · Medical treatment · Health data · Intelligent encryption · Storage method

1 Introduction

With the rapid development of the Internet of things, cloud computing has become the main supporting technology of the Internet of things due to its strong computing power, convenient access, low cost and other advantages, which is the development direction and research focus of the IT field at present [1]. At present, cloud computing has entered a period of rapid development, and various big data applications based on cloud computing are gradually enriched. Users' awareness of adopting cloud services to process various computing, storage and other tasks has increased. Many governments, educational and

S. Wang et al. (Eds.): IoTCare 2021, LNICST 415, pp. 216–232, 2022.
https://doi.org/10.1007/978-3-030-94182-6_16

medical institutions and large enterprise groups take the initiative to consider integrating their business into the cloud. In the era of rapid expansion of information, cloud storage in the storage and management of large data applications has a very obvious advantage.

Medical cloud combines cloud computing, cloud storage and big data processing technology to improve the construction of hospital intelligent information technology to a higher level. First of all, cloud storage and cloud computing are introduced into the construction of medical cloud. Through unified deployment and centralized management, the investment of hardware equipment for hospital information management is reduced [2], the utilization ratio of storage resources is improved, and the operation and maintenance costs of medical information system are reduced. Secondly, due to the popularization of cloud computing in health care system and major hospitals, a large number of clinical data, patient information and so on are rapidly expanding to form medical big data. Finally, medical big data itself has high application and research value, so it is necessary to extract valuable and meaningful data from hospital information platform data to construct medical big database.

Traditional personal computer and server architecture can not meet the requirements of medical big data processing [3]. Cloud storage technology provides an efficient management mode for medical big data processing. In the application of medical cloud system, users are most concerned about the data security, especially the sensitive medical information data. In medical cloud system, in order to protect the data security in sensitive medical database, the common method is to encrypt the data in the database and store the encrypted medical data to cloud server. However, the data encrypted by traditional encryption algorithm has lost a lot of data processing functions. Therefore, how to encrypt the data of medical database efficiently is one of the urgent problems in the application of medical cloud.

In reference [4], an intelligent database encryption method based on AES algorithm is proposed. The method comprises the following steps: constructing a security key for encrypting sensitive database information, adopting an arithmetic coding design scheme to carry out adaptive feature classification and vector quantization coding design of sensitive database storage information, carrying out cyclic code reconstruction and block encryption design of sensitive database storage data encryption under advanced encryption standard protocol (AES), constructing encryption and decryption keys, and carrying out feature block recombination on plaintext decoded by intelligent encryption cyclic code of sensitive database, The random sampling method is used to reconstruct the sample space of the information stored in sensitive database, and the multi-layer encryption design of highly sensitive database storage is realized by combining the key negotiation and elliptic linear coding method. In the literature [5], a decentralized blockchain information management scheme is proposed to realize the safe storage of medical data. The scheme adopts improved PBFT consensus algorithm and optimized Hash encryption algorithm, and stores medical data safely and effectively in distributed database, ensuring the integrity and traceability of medical data. At the same time, a brand-new data interaction system is designed to prevent the direct interaction between the third party and the database, so as to prevent the untrustworthy third party from maliciously destroying the medical data and ensure the security of the data. Finally, the access control and Lucene retrieval mechanism are used to protect patients' privacy and realize fast retrieval of medical data. Experimental analysis shows that compared with algorithms such as Proof of Work, POW) and Delegated Proof of Stake, DPOS), the improved PBFT consensus algorithm provides better stability and throughput for medical blockchain system. Compared with the common database interaction, the data interaction system effectively prevents the direct operation of the database, and has better security and tamper resistance.

The most common method of database encryption is based on a single record. The basic idea of this method is that each record is encrypted into ciphertext and stored in the database file, and the smallest granularity of decryption is generally records. Each foreign large-scale database manufacturers have more or less provided the solution of database encryption, partly solved the problem of encrypting confidential data. However, due to various limitations, the solutions provided by various database vendors can not achieve the effect of multi-user sharing key and encrypting data, and it is very troublesome to migrate ciphertext data between different DBMSs. Because the application of database and computer network in our country is later than abroad for many years on the whole, so the need of database encryption software in our country is very urgent. Through the use of peripheral encryption software, the data in database is encrypted, and a database encryption system can be designed. The encrypted tables and fields can be chosen by users freely, and the encryption process can be transparent to the application.

Therefore, this paper proposes to use the Internet of Things technology to design an intelligent encryption storage method for medical and health database. Through collusion mode, the key family, key period and key management mode of intelligent encryption algorithm of medical database are determined. According to the relationship between encryption and user operation, the database encryption and decryption processing mode is designed, the conversion of database data is realized, and the database encryption scheme using Internet of Things technology is realized. It can provide theoretical support for the safe storage of medical data, ensure the secrecy of medical data, and has important practical significance for the safety of patient privacy.

2 Intelligent Encrypted Storage Method for Medical Health Database Based on Internet of Things

2.1 Design Intelligent Encryption Algorithm for Health Care Database

The design of the health database intelligent encryption algorithm, which is divided into encryption algorithm, decryption algorithm and matching algorithm in three parts. Among them, encryption algorithm is to encrypt plaintext into ciphertext, according to the type of plaintext is divided into ordinary string encryption and pattern string encryption algorithm; decryption algorithm is to restore the ciphertext data in the database to plaintext data [6], and can be based on the check algorithm to infer whether ciphertext is tampered with; matching algorithm is to match the characters in the pattern string with the characters in the ordinary string stored in the database one by one, so as to find out ciphertext data containing keywords. In addition, we also need to determine the key family, key cycle and key management.

Encryption Algorithm

The design of intelligent encryption algorithms for health care databases will use a collusion pattern, which often contains ordinary characters and special characters, such as letters a, b, c, d, and special characters such as *, /, and ?, +, {, }, etc. At the same time, numbers can be either ordinary characters or special characters. A special character in a pattern string has a special meaning, which generally indicates that the preceding ordinary character occurs more than once, such as a indicates that a occurs one or more times. For a pattern string can not be simply encrypted into ciphertext, because the pattern string of special characters encrypted into ciphertext, will lead to special characters lost its meaning, then the server's regular parsing engine can not parse the meaning of the expression [7]. For example, if the pattern string a is encrypted into ciphertext, the server cannot know what it means, but only encrypts a, and if the + does not, the corresponding ciphertext is repeated one or more times for the server. So, for ordinary characters can use encryption technology, to hide its specific meaning. However, special characters should retain their specific meaning and can contain ordinary characters that are encrypted or not encrypted using encryption algorithms. The result is an ambiguity in the ciphertext string. For example, suppose the ciphertext character a is encrypted and becomes a sign, then the ciphertext string ++ can mean either the ciphertext string aa or the ciphertext string a +, whereas aa and a + have completely different meanings in a regular expression.To solve this problem, a flag flag is introduced to distinguish between a character that is a plain ciphertext character data or an unencrypted special character data. The form of the flag prefix code is combined with each ciphertext character, meaning that each clear text character in the pattern string is processed (encrypted or reserved) and the flag bit is added before that character. The length of flag is one byte (8 bits).

After the introduction of flag bit and special character encoding, the design of mode string encryption is as follows: mode string encryption granularity is a single character, for the processing of a single character, first determine its character category, according to the character category set flag value [8]. The characters shall be encrypted. If they are ordinary characters, they shall be encrypted into ciphertext characters by AES algorithm. If they are special characters (key characters in regular expression), they shall be encoded according to the encoding method shown in Table 1.

Table 1. Special character encoding table

Character	Code	Character	Code
ˆ	0x0000	*	0x0001
$	0x0002	#	0x0003
+	0x0004	?	0x0005
{	0x0006	}	0x0007
[	0x0008	]	0x0009
.	0x000A	,	0x000B

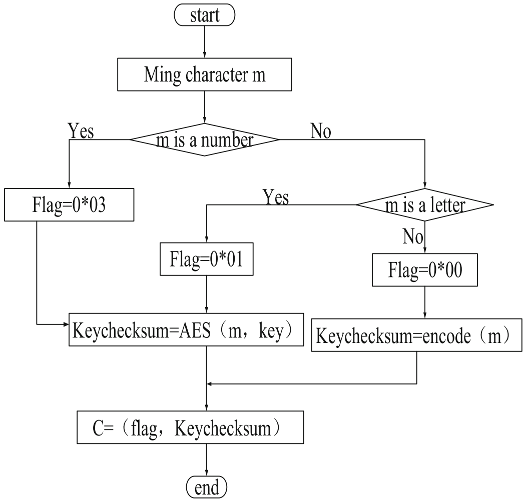


Fig. 1. Flow chart of modal string encryption algorithm

The encrypted characters are called trap gates, which are recorded as keychekcsum, and the flag flag is then stitched into the trap keychecksum to form the final ciphertext data, as shown in Fig. 1. The ciphertext pattern string can be obtained by processing each character in the pattern string, and then the ciphertext data satisfying the regular expression can be queried by ciphertext retrieval algorithm.

Decryption Algorithm

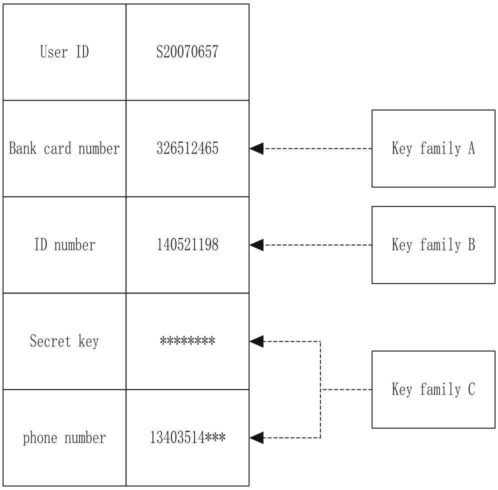
Decryption of ciphertext data is the inverse operation of encryption algorithm. The ciphertext character is stored as ({c} checksum), so decrypting c gets the corresponding plaintext character. Firstly, C is decrypted by AES algorithm. The plain-text character m can then be calculated by the exclusive OR operation of the random numbers r and c produced by the random number generator [9]. In order to verify whether the decrypted data has been tampered with, the AES algorithm can be used to encrypt m to get the trap keychecksum. Through the irreversible hash algorithm MD5, c and key are calculated to get the check code. Finally, the comparison between cs and ckecksum is equal. If not, the ciphertext data c is tampered with. Because the key in the pattern string is different from the key in the normal string, an attacker cannot change the value of checksum even if the normal string key is compromised.

Matching Algorithm

The matching granularity of ciphertext matching algorithm is single character, and the receiving object is ciphertext pattern string. Each ciphertext character in a ciphertext pattern string is composed of ({flag} keychecksum). First, the meaning of the ciphertext pattern string is parsed by analyzing whether each keychecksum is an ordinary character encrypted or a special character encoded using flag flags [10]. Then ciphertext matching algorithm is used to match ciphertext characters one by one. For ciphertext matching algorithms, the storage format of ciphertext characters is ({o} oheoksum) and the character format of ciphertext pattern strings is ({flag} keychecksum), so the value of ({c} keychecksum) is computed as checksum ‘through the irreversible hash algorithm MD5, comparing whether checksum is equal to a checksum, and if equality proves that the two ciphertext characters correspond to the same plaintext character, matching can proceed downward until a destination string that satisfies the meaning of the pattern string is found.

Key Family

Key family A is used to encrypt ID number and Key family B is used to encrypt hospital information. The key family B is used to encrypt the same group of data [11]. Each key in the family of keys has its own specific encryption object, but at the same time, only one key should be used for encryption, while the others may be used to decrypt data, and some keys may have expired, as shown in Fig. 2.



User ID	S20070657
Bank card number	326512465
ID number	140521198
Secret key	********
phone number	13403514***

Fig. 2. Key family example

Consider assigning a key family to each column in the database that needs to be encrypted. If there are three columns in a table that need to be encrypted, three key families should be set for encryption [12]. As an example of the key family shown in Fig. 2, it is assumed that ID number, bank card number, key and mobile phone number are the data that need to be encrypted and protected. Key family A encrypts bank card number, Key family B encrypts ID number, Key family C encrypts key and mobile phone number.

However, if each key family encrypts one column, all keys are fetched for decryption when a row of data is read, which degrades system performance. If only one key family is used for all columns, the impact on system performance can be reduced to some extent, although the extent depends on how well the key engine handles the request. In fact, the performance gains from encrypting multiple columns with the same key family are limited, so it is not worth sacrificing security.

Key Cycle

Key cycle, that is, the key from the creation, waiting, enabled, expired, deactivated until the destruction of the entire process. One key cannot be used all the time, because the more and longer data encrypted with the same key, the more likely an attacker is to crack the key and the less secure it is, so the key needs to be changed periodically [13]. During the replacement process, all data encrypted with the old key needs to be decrypted, and then encrypted with the new key, the old key will be discarded, and will be destroyed later. The whole process of the key cycle is shown in Fig. 3.

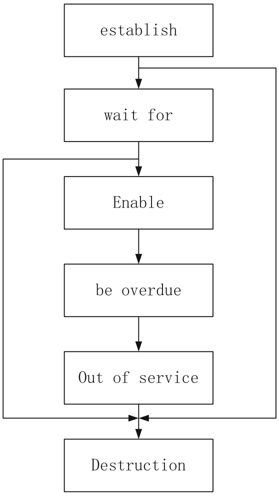


Fig. 3. Key cycle

A key is created as required, and is assigned a key family and activation time. Before the activation time arrives, it is in a waiting state, when the activation time is reached, it is enabled, active, and is used by the encryption system to encrypt and decrypt data until it is disabled by the new key and destroyed by the system after a certain time. Keys that are expired will never be used again. Deactivation key should be deleted as soon as possible [14], even if the key is no longer used, there are security risks, a little improper handling, it may cause data leakage.

When the old key is replaced by the new key, all data encrypted by the expired key is decrypted and then encrypted with an active key. When all the data is encrypted, the expired key is marked as inactive and may be deleted very quickly [15]. There is also a risk that a large amount of data can be decrypted and encrypted in a very short time. Key replacement is a very resource-consuming process. Frequent key changes may result in errors and outweigh the benefits. Therefore, the time for key replacement should be reasonably set according to the actual needs of the data system.

Key Management Method

Key management plays an important role in database encryption technology. Even if the encryption intensity is high and the key algorithm is complex, once the key is compromised, the data in the database is not safe and the encryption is meaningless. Key distribution management is a difficult problem, which involves the whole process of key creation, wait, enable, expire, deactivate and destroy.

There are two kinds of keys in database encryption system: user key and data key. User key is mainly used to query and modify data. A user may provide his personal information (user name, secret key, etc.) to the database management system, and after the verification and confirmation of the legitimate authority of the user, the user may conduct corresponding operations. Depending on the permissions, database users are divided into ordinary users and database administrators. Ordinary users may have different permissions, such as VIP users who generally have more permissions than free users. Database administrator has the most authority to deal with both plaintext data and ciphertext data. For the sake of security, there may be many administrators managing a database. Even so, administrators are often a hidden danger of database security that can not be ignored.

Data keys fall into two categories: master keys and working keys. The master key is used for encryption and is changed periodically. The security of the master key is very important to the whole database system, because once it is compromised, all the data will be in an insecure state. Work keys are used to encrypt tables or data items, and each table or data item requires a different work key.

Keys are often generated randomly through mathematical functions, there is no law, so it is impossible to predict the statistical analysis of the key attacks is impossible, so the key itself without security risks. The key generating function shall meet the following conditions: the probability of the same key of different data items is extremely low; even if some information of the data items is obtained, it is impossible to infer the information of other data items from ciphertext; the key of one data item cannot infer other keys; and the key generating speed shall be as fast as possible.

The key storage is divided into: the master key storage, the work key storage, user key storage. The storage of the master key is suggested to use some hardware encryption devices with high encryption strength and inviolability. If it is implemented by software, the master key generator should be kept out of all hidden troubles, and the administrator who manages the master key should be absolutely trusted. The master key should be changed periodically for security reasons.

The working key is usually encrypted by the master key and stored in the database data dictionary in the form of ciphertext. Because of its large number, it is generally not allowed to modify the operation directly.

Generally, the user key is stored in the database table, the user will save a copy, so in addition to ensuring database security, the user should also properly keep the key, it is best to use key protection card to access the database.

2.2 Database Encryption, Decryption and Conversion

In addition to the encryption algorithm, the encryption and decryption of the method is the basis of the whole operation process throughout the method. It includes both the encryption and decryption of user information and the encryption and decryption of data keys.

The decryption process is similar to encryption except that the parameters of the Cipher object are set slightly differently. It is important to note that the encryption and decryption processes cannot be performed simultaneously. If you want to do both encryption and decryption, you need two instances of the Cipher object. In fact, we can find that the whole system's encryption/decryption process is closely related to the user's read-write operations. The relationship between the encryption and decryption processes and user read-write operations is shown in Fig. 4.

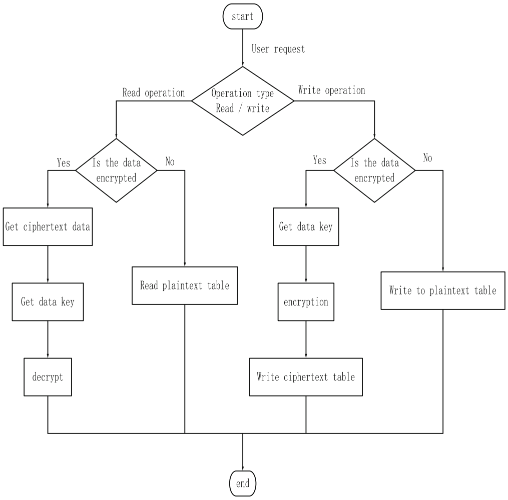


Fig. 4. Relationship between encryption and decryption and user read and write operations

As you can see in Fig. 4, when a user reads, the system determines whether the pre-read data is encrypted. If not encrypted, the system directly reads the plaintext data and returns. If it has been encrypted, the system will obtain the relevant data keys for decryption. When a user writes, the system also determines whether the pre-written data needs to be encrypted. If no encryption is required, the system writes the plaintext data directly and returns. If encrypted storage is required, the system will obtain the relevant data keys after encrypting the plaintext data.

In this study, in order to facilitate users to observe the changes of data before and after encryption, binary data is converted into hexadecimal string and stored in the database. Here is mainly for the character type of data encryption. The data conversion code is contained in the Utils class, which is used to convert byte arrays to hex strings. It contains two methods, {by} tes? The HexString () method is used to convert a byte array to a hexadecimal string. Another hex String2Bytes () method, which is inversely related to the above method, converts a hexadecimal string to a byte array.

Because the computer directly processes the binary data the speed is quicker, therefore in the actual application, directly uses the binary form to encrypt the ciphertext data carries on the storage. Also, binary data saves more storage space than encoded hexadecimal data.

2.3 Intelligent Encrypted Storage of Medical Health Database Based on Internet of Things Technology

The cloud computing technology of Internet of things technology is adopted to realize the intelligent encryption algorithm of medical health database. The process is as follows: Firstly, the data users in cloud computing database are partitioned by sharding functions, and the data is distributed to different processors by the master node as the dispatcher. Each Mapper completes a portion of the final result. The Reduce function is responsible for consolidating some of the results of all Mappers, as shown in Fig. 5.

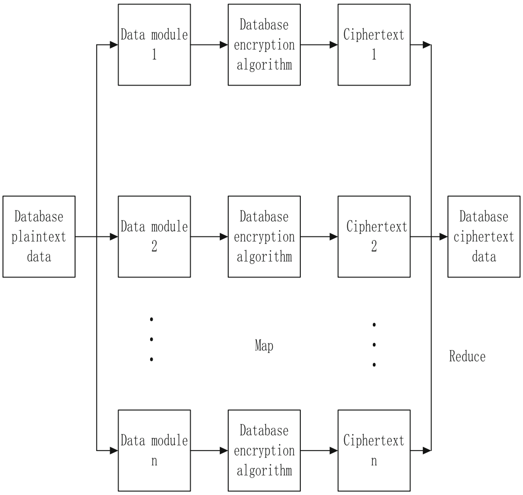


Fig. 5. Database encryption scheme based on Internet of Things

As can be seen from the analysis of Fig. 5, the implementation of intelligent encryption algorithm for health care database based on Internet of Things technology mainly includes collusion mode encryption stage and parallel encryption stage, which are described below:

Fragment function to realize the encryption stage (serial encryption stage): the initial file is divided into n fragments through one-time fragmentation, in which the file size is represented as L, the i fragments as i and the i fragment database data size is represented as L_i. In order to decide whether the file reading process is over or not, we use POS to represent the location of the real-time processed data in the initial file, and use i to store the data after the slicing through filebuffer, and i sliced data filebuffer to store the value.

The encryption results of the serial encryption phase are processed in parallel, and the Map function realizes the encryption phase as (the parallel encryption phase): any Map node is encrypted in its entirety, resulting in any block of ciphertext. Set the input of the Map function as key/value pair, the offset of the block data in the initial data is key, and the clear text representation of the block data is value.

The Reduce function implements the encryption phase as (parallel encryption phase): after the Map phase encryption processing is completed, the key value pairs of all nodes are counted and sorted by the value of the key as a weight, and the value is pieced together to get the final file.

A sorting algorithm is used to process the file assembled from the above values, and the encryption complexity calculation formula is obtained as follows:

$$O(t) = s \tag{1}$$

In this formula, t represents time, O represents encryption complexity, and s represents the encryption process. At this point, assume that the constant of the encryption process is α and t_0, and at $t > t_0$, there are $O \leq \alpha s$, and assume that the complexity associated with sorting in the Reduce function process is $m\ x$ operations, and $\alpha > 0$, then use expression (2) to minimize the encryption complexity:

$$O_{\min} = \min \alpha ksO(t) \tag{2}$$

In the formula, k represents the offset of the chunked data from the initial data.

Based on the above contents, the algorithm, data encryption, decryption and conversion process, as well as the two calculation formulas are substituted into the database encryption scheme diagram based on the Internet of Things technology shown in Fig. 5 to realize the intelligent encrypted storage of medical health database.

3 Experiment and Analysis

Two groups of traditional encrypted storage methods were selected, and the health database was chosen as the experimental object to verify the intelligent encrypted storage method of health database based on Internet of Things.

3.1 Experimental Preparation

According to the operating characteristics of the encryption storage algorithm selected in this experiment, three sets of encryption storage methods are set up on the computer, as shown in Table 2.

Table 2. Running environment

The environment	Configure the	Specification
Software	Operating system	Unbutu
	Operating system Type	The 64 - bit
	Agent server open source software	SNProxy
	Network card	Gigabit network
	Network card transmission rate within LAN	1Gbps
	Database	MYSQL 5.0.41
	Auxiliary Function Writing Language	Javascript
	Test the script	Python
	IDE	MyEclipse 7.5
Hardware	Server server	MongoDB
	Ser ver Version	MongoDB3.0.2
	Proxy server	SNProxy
	CPU of physical machines	Intel i5 7500
	CPU Memory	8GB
	CPU main frequency	2.9 GHz
	Mechanical hard disk capacity	500GB

Based on the running environment of encrypted storage method shown in Table 2, the encryption and decryption algorithms of the three encrypted storage methods are stored in the SNPROXY proxy server.

During the test, the client first connects to the MongoDB server via the SNProxy proxy server to perform the query and record the output. The client then connects directly to the MongoDB server to perform the query and record the output. Compare the two query results before and after, observe whether the first query results are clear and correct, and whether the second query results are ciphertext data. This verifies the performance of three cryptographic storage methods.

Through the method of comparison test, the time consumed by various basic operations (adding, deleting, modifying, querying) of MongoDB server database without using SNProxy proxy server and using SNProxy proxy server is measured respectively. The time is the time interval between the client initiates the request and receives the reply, so as to determine whether the response time of the whole system can be severely slowed down when the system performs the encryption and decryption calculation.

In the design of SNProxy proxy server, the key management mechanism is also introduced, and the additional time cost is introduced in the generation of encrypted sub-key and the management of online and offline user information.

3.2 Experimental Results

Results of the First Set of Experiments

According to the experimental process of this experimental design, three groups of encrypted storage methods are tested respectively. In the functional test process of this design, when inserting, key-value query, aggregation, range, fuzzy query, update, delete and other operations are carried out, the function test result is shown in Table 3.

Table 3. Functional test results table of three cryptographic storage methods

Method	Type of operation	Implementation results	Output results	
			SNProxy	MongoDB
Research method	Insert operation	Correct	–	–
	Key Value Query	Correct	Plain text	Secret text
	Aggregation operations	Correct	Plain text	Secret text
	Scope operations	Correct	Plain text	Secret text
	Fuzzy query	Correct	Plain text	Secret text
	Update operations	Correct	–	–
	Delete operations	Correct	–	–
Traditional method 1	Insert operation	Correct	–	–
	Key Value Query	Correct	Plain text	Secret text
	Aggregation operations	Error	Plain text	Plain text
	Scope operations	Correct	Plain text	Secret text
	Fuzzy query	Error	Plain text	Secret text
	Update operations	Correct	–	–
	Delete operations	Error	Plain text	Secret text
Traditional Method 2	Insert operation	Error	Plain text	Secret text
	Key Value Query	Correct	Plain text	Secret text

(*continued*)

Table 3. (*continued*)

Method	Type of operation	Implementation results	Output results	
			SNProxy	MongoDB
	Aggregation operations	Error	Plain text	Secret text
	Scope operations	Correct	Plain text	Secret text
	Fuzzy query	Correct	Plain text	Secret text
	Update operations	Correct	–	–
	Delete operations	Correct	–	–

As can be seen from Table 3, in the process of testing the three groups of encrypted storage methods designed in this experiment, the output results of the traditional method 1 are all clear text data when the client uses SNProxy for query, but in the process of deletion, errors of encrypted storage operation occur, which leads to multiple clear text in the output results when the client directly connects to the MongoDB database for query again; the traditional method 2 uses SNProxy for query, the output results are all clear text data when the client directly connects to the MongoDB database for query again, but in the process of insertion, errors of encrypted storage operation occur, which leads to the existence of clear text in the output results when the client directly connects to the MongoDB database for query again; the research method uses SNProxy for query and the query results are correct, and when the client directly connects to the MongoDB database for query again, the output results are all clear text data. This means that SNProxy encrypts the data and enables the MongoDB database to perform a query on the ciphertext. Obviously, this research method has the better encryption performance.

Second Set of Experimental Results

All the experimental data selected in this experiment are normalized, and the experimental data size is controlled to one unit. The data size of the three methods after encryption is calculated, and the ratio of the data size before encryption to the data size before encryption is compared. The data size ratio before and after encryption is compared. The experimental results are shown in Fig. 6.

It can be seen from Fig. 6 that the data size ratio of the medical health database data encrypted and stored by traditional method 2 is the largest before and after encryption, which indicates that the data of the medical health database encrypted by traditional method 2 increases, resulting in large space overhead; The ratio of data size before and after encryption in traditional method 1 is 3% smaller than that in traditional method 2, which indicates that the data of medical health database encrypted by traditional method 1 increases and the space overhead is smaller than that of traditional method 2. The data size ratio before and after encryption is the smallest, which is 7% and 4% smaller

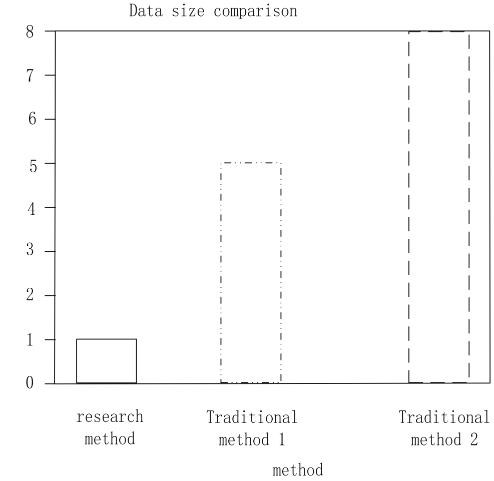


Fig. 6. Data size ratio before and after encryption

than that of the two traditional methods, indicating that the data of the medical health database encrypted by the research method increases and the space cost is smaller than that of the two traditional methods. Therefore, this research method has better encryption performance.

The Third Group of Experimental Results

Change the number of experimental data selected in the experiment, let three groups of encryption storage methods manage the data keys of medical health database in the same cycle, and process the data of medical health database, calculate the ratio of time

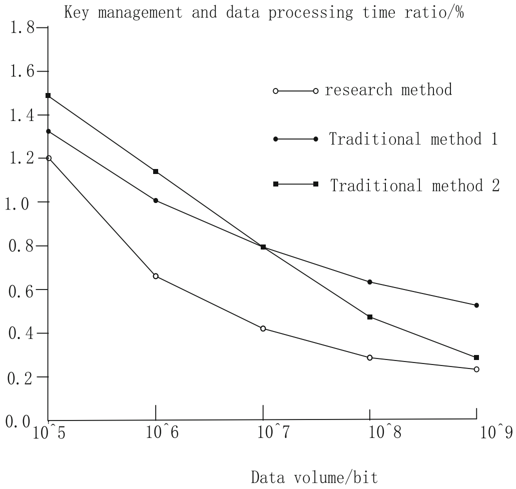


Fig. 7. Key management to data processing time ratio

consumed by key management and data processing in the same cycle, and compare the time ratio of key management and data processing in the same cycle. The experimental results are shown in Fig. 7.

As you can see in Fig. 7, the amount of time spent in key management is doubly reduced as the amount of data in the database increases by an order of magnitude, depending on the number of documents in the database. However, the average ratio of key management time to data processing time of traditional method 2 in one operation cycle is 0.832%, the average ratio of key management time to data processing time of traditional method 1 in one operation cycle is 0.856%, and the average ratio of key management time to data processing time of research method in one operation cycle is 0.558%. Compared with the two traditional methods, the average ratios of key management time to data processing time are 0.274% and 0.298%, respectively, so this method has better encryption performance.

According to the above three experimental results, this research method can meet the needs of data security and encrypted storage, and make the basic operation of database better supported. It has less space overhead and faster database encryption speed.This is because the method studied in this paper decomposes the intelligent encryption algorithm of medical database into key family, key period and key management mode with the help of Internet of Things technology, manages the key by using ciphertext matching algorithm, and increases the steps of data conversion, thus improving the security protection performance of database encryption and decryption.

4 Closing Remarks

To sum up, the research on the intelligent encrypted storage method of medical health database based on the Internet of Things makes full use of the technology of the Internet of Things to enhance the encryption effect of the encrypted medical health database, so that the time expenditure and data expansion rate of the encrypted medical health database are in an acceptable range and the cost of application system access is reduced. However, there are still some deficiencies in this study, in the future research, we still need to study the key management scheme to protect the data of online users of medical health database.

References

1. Shuaihang, Y., Jinrong, H.: Application of data encryption technology in computer network data security. J. Yanan Univ. (Natural Science Edition), **40**(1), 78–82 (2021)
2. Linxin, W., Shigang, L., Shulin, L., et al.: De privacy and data encryption of smart grid big data based on deep learning. Electronic Design Eng. **29**(3), 175–178,183 (2021)
3. Yiming, W.: Analysis of key matrix encryption method under one-way HASH function. Wuxian Hulian Keji **18**(1), 69–70 (2021)
4. Yahong, L.: Research on intelligent encryption method of sensitive database based on aes algorithm. J. Nanyang Inst. Technol. **011**(002), 31–35 (2019)
5. Hui, W.A.N.G., Yuxiang, L.I.U., Shunxiang, C.A.O., et al.: Medical data storage mechanism integrating blockchain technology. Comput. Sci. **4**, 285–291 (2020)

6. Wenbo, X.I.E., Yongzhuang, W.E.I., Zhenghong, L.I.U.: Parallel implementation and analysis of SKINNY encryption algorithm using CUDA. J. Comput. Appl. **41**(4), 1136–1141 (2021)
7. Minmin, X.I.E., Yong, W.A.N.G., Lin, Z.H.O.U.: IEC 61850 Communication message encryption algorithm based on SM2-SM3. Autom. Panorama **38**(1), 108–112 (2021)
8. Hailing, L.: Analysis and discussion of data encryption technology in computer network communication security. Telecom Power Technol. **37**(4), 222–223 (2020)
9. Zijie, Z.: The role and application of data encryption technology in computer network communication security. Telecom Power Technol. **37**(3), 205–206 (2020)
10. Guoqing, Z.: Data encryption algorithm. Comput. Eng. Softw. **41**(10), 129–131 (2020)
11. Xia, W.U.: Research on data storage encryption technology under the background of Big Data. China Comput. Commun. **32**(24), 136–138 (2020)
12. Junli, Y.A.N.G.: Research on user privacy data encryption protection system based on Big Data analysis. Microcomput. Appl. **36**(8), 65–67 (2020)
13. Liu, S., Liu, G., Zhou, H.: A robust parallel object tracking method for illumination variations. Mob. Netw. Appl. **24**(1), 5–17 (2018). https://doi.org/10.1007/s11036-018-1134-8
14. Shuai, L.I.U., Weiling, B.A.I., Gaocheng, L.I.U., et al.: Parallel fractal compression method for big video data. Complexity **2018**, 1–16 (2018)
15. Shuai, L., Tenghui, H., Jianhua, D.: A survey of CRF algorithm based knowledge extraction of elementary mathematics in Chinese. Mob. Netw. Appl. 1–13 (2021)