ATLANTIS PRESS

Advances in Intelligent Systems Research, volume 134

2nd International Conference on Control, Automation, and Artificial Intelligence (CAAI 2017)

A Key Management Scheme Based on Outsourced Database

Yonghui Shi and Kunfu Wang

CHINA SHIPBUILDING IT CO., LTD. BEIJING CHINA

***Abstract*—Aiming at the security issues in outsourced databases about data confidentiality and integrity, this paper proposes a key management scheme which is safe and effective based on NTRU (Number Theory Research Unit) signature technology. Firstly, according to the structure of the data table, the attribute parameters are selected to generate the data encryption key, and then the encrypted data is encrypted; Secondly, the encrypted data and signature data are stored together to the outsourced database, and data encryption key is stored in the local security database which is protected by the system master key, and the master key and the signature private key are stored in the hardware security module; Finally, when the data is decrypted, the NTRU signature of the requested data is first verified, and the data can be decrypted if the authentication is passed, otherwise the decryption data is not needed. The security and performance analysis results show that the scheme is secure and effective, and the key management scheme is secure and stable under different encryption algorithms, on the other hand, confidentiality and integrity protection can be provided at the same time.**

Keywords-outsourced database; NTRU signature; database encryption; key management; attribute parameter

I. INTRODUCTION

In recent years, with the development of electronic commerce and expand the scale of data, Hacigumus [1] et al. first proposed outsourced database (Database as a server DAS) conceptual model. In outsourced database, enterprise or organization will be entrusted to database business service provider ODB (Database server provider) to manage and maintain the database, this approach not only save management costs, but also can share data, but on the other hand, it is a new challenge to ensure the security of database. The traditional database security mechanism includes user identification, identity authentication, access control, audit mechanism, backup and recovery. But they cannot completely guarantee the security of the database. Database encryption technology can protect data, however, in the database encryption, key management is a problem to be solved, and a good key management scheme can greatly improve the efficiency and safety of outsourced database.

In order to solve the problem of key management in encrypted database, some solutions are proposed. In 1981, Davida [2] et al. put forward the key management technology based on the sub key, which opened precedent on the field and the encryption of the database system, but its security needs to be further verified. Sun [3] proposed a key management scheme based on the Chinese remainder theorem, this method makes use of identity authentication mechanism, and it improves the security and trust between users, DBA (Database administrator) and data. In addition, according to key management in the cloud database, Cui [4] et al. proposed a scheme in which each user only needs to master a static key without needing to pay attention to whether the data is re-encrypted and whether the authority is changed or not, this method reduces the key management work to a certain extent. Aiming at the security problem of outsourced database under cloud computing, Weis [5] et al. proposed a scheme for user to store data key in the local security database LSDB (Local Secure Database), and to transmit the key information through the secure channel during the encryption and decryption, so as to ensure the security of the data key. However, Vimercati [6] et al. use two level encryption scheme to store key management system to cloud, although the scheme is effective, it has not yet been applied to database encryption. In the aspect of key sharing, on the basis of certificate-less signature authentication mechanism, Cheng Fangquan [7] proposed a trusted user key sharing protocol in the trusted database environment, which was proved to be effective against all kinds of attacks in certificate-less security model. In order to improve the query efficiency, Li Jin [8] first proposed a thought, which fused Huffman encoding, Bloom filters and traditional encryption algorithms, this method can improve the query efficiency and reduce the storage cost.

With the growing demand for outsourced database, it is necessary to protect the confidentiality and integrity of outsourced database, so we must combine database encryption and integrity detection technology to ensure the security of database. We used watermarking technology [9] and digital signature [10], probability theory [11] and verify the data structure of technology [12, 13] to solve the integrity problems, however, it is necessary to design a reasonable and effective key management scheme to improve the security of the whole system. In this paper, on the basis of database encryption and NTRU signature, a secure and efficient key management scheme is proposed, which can make use of the structural characteristics of the data table and the attribute parameters to generate different data keys, and perform the NTRU signature for the encrypted data. The proposed scheme not only can generate data keys, but also can ensure the confidentiality and integrity of database.

II. KEY MANAGEMENT SCHEME DESIGN

The key management scheme proposed in this paper is as follows:

a) User needs to generate the dataKey before encrypting the data. Firstly, the attribute parameter is selected according to

structure of the database table and the data key is then generated using attribute parameters;

b) The encrypted data is then NTRU signed, and the cipher text and signature data are encapsulated into an outsourced database;

c) The data key must be stored in the local security database LSDB and stored by the system master key MK (Master Key);

d) The private key SK and master key MK, which are generated in the NTRU signature, are saved to the hardware security module HSM.

The key management scheme is shown in Figure 1. Among them, d_{ij} represents the plaintext data item in column i of line j in the data table, M represents the plaintext data, C represents the cipher text data, Ap represents the attribute parameter, PK represents the public key of the NTRU signature, and SD represents the signature data.

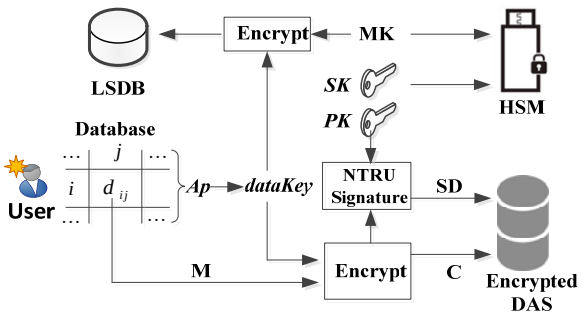

FIGURE I. THE SCHEME OF KEY MANAGEMENT

A Key Generation

Outsourcing database encryption not only need to consider the database encryption granularity and data encryption algorithm, but also need to resolve the problem of generating data key in different granularities, and key function need to have characteristics of high safety and fast operation etc. It has the following security requirements for the key generation algorithm F :

a) F Algorithm need to try to choose a cryptographic one-way function;

b) The data key must be generated using table key T_K ;

c) Different encryption keys are generated for different data, and the same probability is very small;

d) The other data key may be derived from a data key.

In accordance with requirements of the appeal, each database table in this scheme has a table key T_K , each row and column of the data table has the attribute parameters, in which R_i is used to express the parameters of line i , and C_j is used to represent the parameters of the first j column. The following is the study of data key generation method based on field level, record level and data item level encryption.

1) Field level encryption

When the field of data is encrypted, it is necessary to generate both the field key K_{column} and the field key flag M_{column}, which correspond to each other, and they cannot be deduced from each other. $M_{column} = H(T\_K, C_j)$, $K_{column} = subStr(K_j, dkLen)$, $K_j = H(T\_K, C_j \oplus salt_j) \oplus salt_j$,Where K_j represents the initial field key, $salt$ is not repeated salt value, $subStr(K_j, dkLen)$ said the output length field key function of $dkLen$, which mainly is the interception of K_j , and less than the number of characters need to add.

2) Record level encryption

When the record of data is encrypted, it is necessary to generate both the record key K_{row} and the record key marker M_{row}, which correspond to each other, and they cannot be deduced from each other. $M_{row} = H(T\_K, R_i)$, $K_i = H(T\_K, R_i \oplus salt_i) \oplus salt_i$, $K_{row} = subStr(K_i, dkLen)$,Where K_j represents the initial record key, $salt$ is not repeated salt value, $subStr(K_j, dkLen)$ said the output length record key function of $dkLen$, which mainly is the interception of K_j , and less than the number of characters need to add.

3) Data item level encryption

Data item encryption is the smallest granularity level database encryption, a large number of data items and solutions meet a large amount of resources, so the key which need to generate data items should be fast and high security. The data item key flag calculation function is $M_{daraitem} = H(T\_K, C_j, R_i)$, and the data item key function is $K_{ij} = H(T\_K, R_i \oplus C_j \oplus salt_{ij}) \oplus salt_{ij}$, $K_{dataitem} = subStr(K_{ij}, dkLen)$.

B Data Encryption and NTRU Signature

When the data is encrypted in a database, the selection of encryption algorithm is mainly based on the block cipher, the main reason is that the encryption and decryption efficiency is high. The form of encryption is denoted as $E_{dataKey}(m) = c$ E, in which $dataKey$ D represents the data key, m represents the plaintext, and c represents the cipher text data. When the data encryption is completed, the encryption algorithm needs to carry on the NTRU signature to the cipher text data c , the signature steps are as follows:

Step1: The signature uses the message digest MD algorithm to encrypt the encrypted data c $MD(c) = c'$;

Step2: The signature algorithm determines three integer parameters of the NTRU algorithms (N, p, q), then it randomly selects two polynomials f and g, and $\gcd(f, pq) = 1$, and calculates the multiplicative inverse Fp and Fq of the f module p and q, namely $f \cdot Fp = 1 \bmod p$, $f \cdot Fq = 1 \bmod q$,

where $\cdot$ represents the convolution, then the public key of the signature algorithm is $h = p \cdot Fp \cdot g (\bmod q)$, the private key is (f, Fp);

Step3: The signature algorithm will be encoded c', and then converted into a polynomial, and randomly selected polynomial r, and $r \in D_r$, D_r is the value space, and then it calculates the $SD = (r \cdot h + c') \bmod q$, where SD represents the signature message.

Step4: The cipher text data c and NTRU signature message SD are stored in the outsourced database, and the data key *dataKey* and NTRU private key (f, Fp) are stored in the local security database, and their table structures are shown in Table 1 and table 2.

TABLE I THE STRUCTURE OF DAS TABLE

id	NTRUSign	CM	C	Changed
1	sd_1	cm_1	c_1	Y/N
2	sd_2	cm_2	c_2	Y/N
…	…	…	…	….
n	sd_n	cm_n	c_n	Y/N

TABLE II THE STRUCTURE OF LSDB TABLE

Id	CM	NTRUf	NRUFp	decription
1	cm_1	f_1	Fp_1	Text1
2	cm_2	f_2	Fp_2	Text2
…	…	…	…	…
n	cm_n	f_n	Fp_n	Textn

Among them, the *NTRUSign* in Table 1 represents the NTRU signature data, CM represents the cipher text data flag, and C represents the cipher text data. The $NTRUf$ and $NTRUFp$ in Table 2 represent the private key f and Fp, which are generated in the NTRU signature process. The local security database LSDB is saved by the system master password MK, and the system master key MK is saved to HSM.

C Verify Signature and Data Decryption

When user needs to extract data from an outsourced database, we must first verify the integrity of the data, if the validation does not pass, the encrypted data in outsourced databases is usurped or that is incomplete in the transmission process, then you don't need to decrypt the data. However, if the verification passes, you need to take out the system master key MK from the HSM to decrypt the local security database LSDB, and retrieve the data decryption key, then you can decrypt the data. The schematic diagram of verification and decryption is shown in figure 2.

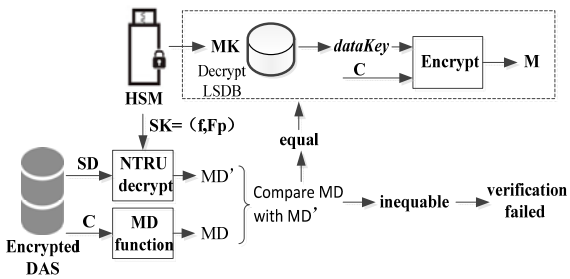

FIGURE II. VALIDATION DATA AND DECRYPT DATA

Step1: User needs to take out the encrypted data C and the corresponding signature SD from outsourced database;

Step2: User needs to take out the NTRU private key (f, Fp) from the local security database to decrypt the signature data SD, then it will get the original message digest MD', the calculation process are as follows:

$$a = f \cdot SD (\bmod q)$$
$$b = a (\bmod p)$$
$$MD = Fp \cdot b (\bmod p)$$

Among them, a and b represent the results of the intermediate calculation, and then the new message digest MD is calculated by the message digest algorithm.

Step3: We compare the numerical value of MD with MD', if they are not equal, the signature verification fails, and then a data state *changed* is returned to the outsourced database, and you don't need to decrypt the data. If $MD = MD'$, you need to take out the system master key MK from LSDB to decrypt the LSDB, and we retrieve the corresponding data key *dataKey* and then decrypt the data.

D Key Replacement

1) Replacement of master key MK

When the local security database is threatened, the system master key can be replaced. The master key system MK is stored in HSM, when user needs to change the system master key, we need to use MK to decrypt data in the LSDB, that is $D_{MK}(c) = m$. Among them, c represents the encrypted data key stored in LSDB, m represents the decrypted data key. Then we use the new master key MK' to encrypt LSDB, that is $E_{MK'}(m) = c'$, and then the main key MK' can be safely stored in HSM.

2) NTRU signature key replacement

Because the signature data is stored in outsourced database, it may be subject to external attacks. Therefore, it is necessary to modify the signature data periodically, so we need to replace the NTRU key. Suppose that we now need a summary of the signature information for MD, the NTRU algorithm generates a common parameter NPQH, the private key pair that it generates is FFP, and then the signed data is SD. If user wants to modify the data which has been signed, there will be some steps:

Step1: User randomly selects two polynomials f' and g', and Fp' in $f' \cdot Fp' = 1 \bmod p$ and Fq' in $f' \cdot Fq' = 1 \bmod q$ are recalculated, then we compute the signature public key $h' = p \cdot Fp \cdot g (\bmod q)$;

Step2: We take out the NTRU signature private key pair (f, Fp) which had been stored into LSDB, then the signature data SD is extracted from the outsourced database, and the following calculations are performed:

$$y_1 = f \cdot SD (\bmod q), \quad y_2 = a (\bmod p)$$
$$MD = Fp \cdot y_2 (\bmod p)$$

Step3: We first calculate the MD, then randomly select polynomial r', $r' \in D_{r'}$ and then calculate the $SD' = (r' \cdot h + MD) \bmod q$;

Step4: SD' will be saved to the outsourced data, and key pair (f', Fp') will be saved to LSDB.

3) Data key replacement

When the cipher text data in outsourced database has been attacked. First of all, user can change the key data; secondly, user needs to modify the attribute parameters which include row and column parameters, however modify one parameter that attribute parameters will change; finally, in order to improve the safety, we need to replace the two parameters (R_i, C_j) and random salt. If the user selects the granularity of encryption as a data item, then $K_{ij} = H(T\_K, R_i \oplus C_j \oplus salt_{ij}) \oplus salt_{ij}$, and the data item key will change. After the change, $K_{ij}' = H(T\_K, R_i' \oplus C_j' \oplus salt_{ij}') \oplus salt_{ij}'$, at this point will be returned to the third section of data encryption and NTRU signature phase; finally, the new data should be saved to the database.

III. Security Analysis of the Scheme

A Attacks Against DAS

1) Cipher text only attack and Cipher text statistics attack

When intruder wants to attack the cipher text data, that is, the cipher text only attack, and at this time, the attacker can only be intercepted from the cipher text, the purpose is to try to get the corresponding plaintext or key. In the implementation process of the scheme, we choose different encryption algorithms, such as AES, 3DES, IEDA and so on, because these algorithms are more robust against attack. The attack of the cipher text data by an intruder is actually an attack on the above algorithms, so this scheme has certain security in the design. In the cipher text statistical attack, the attacker tries to decipher the plaintext based on statistical information from the same cipher text in the database. In this scheme, the data key generated in the data encryption phase will not be the same, and suppose that user selects the data item to encrypt, at this time $K_j = H(T\_K, C_j \oplus salt_j) \oplus salt_j$, $K_{column} = subStr(K_j, dkLen)$ the randomness of attribute parameter C_j and salt $salt_{ij}$ increase the security of the key, So the key will not be the same and the key generation function can meet the security requirements, in addition, $subStr()$ only has the function of intercepting or adding K_j, So we just need to prove that K_j meets the requirements.

Proof: $K_j = H(T\_K, C_j \oplus salt_j) \oplus salt_j$ It can be seen that the key generation function satisfies the security requirements.

Different fields have different attribute parameters CJ, now assume that the two attributes in the same table are CP and CQ, and random salt values are $salt_p$, $salt_q$ and $C_p \neq C_q$, $salt_p \neq salt_q$ $C_p \oplus salt_p = C_q \oplus salt_q$ may occur, that is, $H(T\_K, C_p \oplus salt_p) = H(T\_K, C_q \oplus salt_q)$ at this time there is $salt_p \neq salt_q$, so $K_p \neq K_q$, this function satisfies the safety requirements.

Because each attribute parameters corresponding to fields are not the same, and the random salt value is not the same, so it is not possible to derive another field key from a field key. To sum up, the field key generation function is to meet the security requirements and the statistical attack is not likely to succeed.

2) Attack NTRU signature

In this scheme, the NTRU algorithm is used to encrypt and decrypt the message so as to achieve the purpose of signature, an attack on the NTRU signature can be converted to an attack on the NTRU algorithm. The security of NTRU algorithm is like finding a very short vector in a very large dimension lattice. The algorithm has the ability to fight against quantum computing attacks, while the RSA and ECC algorithm is unable to resist quantum computing, and in the same security conditions, NTRU algorithm is faster than other public key cryptosystems. Up to now, there are a lot of scholars and researchers to discuss the security of NTRU algorithm, but there is no one way to decipher the NTRU cryptosystem. From the current research results, NTRU algorithm based on the difficult problem is safe. The following research is focused on the parameter setting of the system and the selection of the appropriate filling scheme. In 2000 Jaulmes et al. [14] demonstrated that the choice of padding scheme can effectively prevent cipher text attacks. The results of Hoffstein [15] et al. showed that set the correct parameters can reduce the failure rate of data encryption and decryption, even close to 0, then you can put any attacks on NTRU for forwarding to solve difficult problems in grid.

B Attacks Against LSDB

1) Social engineering attacks

The local security database LSDB is stored in the local which user can directly manage the database, and its data flow will not go outside the network, otherwise Wires hark and sniffer capture tool in the network will not produce results. However, attacks against LSDB that can be done by means of social engineering, when an intruder has access to the data in the LSDB, the intruder will encounter a cipher text only attack for DAS attacks. Therefore, it is still difficult to access the plaintext data. So to some extent, LSDB based social

engineering attacks are also difficult to succeed.

2) HSM security attacks

User stores the private key of the signature and the system master key into the HSM. Therefore, it is necessary to consider the security of the HSM, it is mainly through the hardware module to protect the security of key storage, such as SmartCard, SaftNet and other hardware security module. Today, the security scheme of pure hardware has more advantages than the software program. The hardware based security solutions use devices which was designed to meet specific security requirements. Because the hardware is usually equipped with security key, and attacker cannot read or change the key from the outside. In addition, the HSM usually has a built-in protection mechanism that protects them against environmental and physical attacks.

IV. THE PERFORMANCE ANALYSIS OF THE SCHEME

A Comparison of Key Generation Algorithms

There are many ways to solve the problem of key generation in the key management scheme of encrypted database. We often use the key generation based on chaos theory, and based on BBS (Blum Blum Shub) key generation, or key generation based on noise source technology, etc., these methods can meet the needs of different users, and their commonality is the use of randomness to produce the same probability of characters or numbers, etc. This scheme mainly uses the structural characteristics of the data table to construct the attribute parameters, and then generates the data key from the attribute parameters. Among them, the one-dimensional Logistic chaotic discrete model is defined as: $x_{n+1} = \mu * x_n(1 - x_n)$ The sequence generated by $\mu \in [0,4]$, $x_n \in (0,1)$ $n = 1,2,3,\cdots$ and mapping is determined by n, x_0 and μ . And $\mu = 3.58$ or so, the one-dimensional Logistic has chaos. When the key is generated, we need to set up the initial n, the initial value x_n and μ , and chaotic sequence generated by Logistic chaotic discrete model $X = \{x_{n+1}, x_{n+2}, x_{n+3}, \cdots\}$. According to the data encryption algorithm, we select a certain length of the sequence, and the corresponding coding, and then select some of the sequence as a password. We first select two prime p and q in BBS and $p \equiv 3(\bmod 4)$ $q \equiv 3(\bmod 4)$ $n = pq$. We randomly select the integer s , $\gcd(s,n) = 1$ and BBS polynomial is $x_0 = s^2 \bmod n$, $x_i = (x_{i-1})^2 \bmod n$ and then we use $b_i = x_i \bmod n$ to indicate the password. In the literature [15], the key generating function is $k_{ij} = k_{1i} \oplus k_{2j}$, among them $k_{1i} = E(TK, R_i)$, $k_{2j} = E(TK, C_j)$ TK is the table key, and E is a block encryption algorithm. The following is a comparison of the methods based on BBS, chaos theory, the proposed [15] and the time of key generation in this scheme, as shown in figure 3.

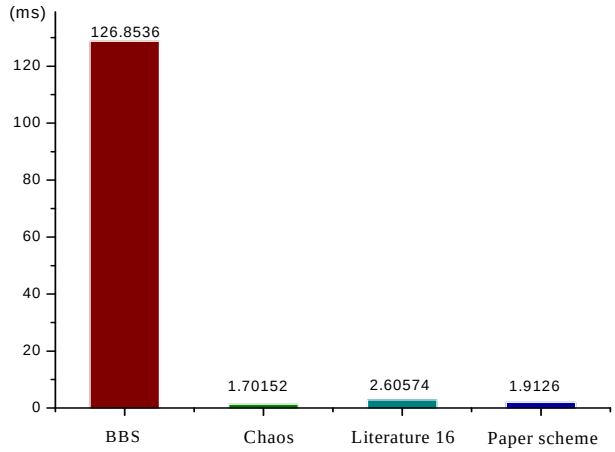

FIGURE III. COMPARE THE TIME OF KEY GENERATION

Through the analysis of these key generation algorithm, the key generation algorithm based on BBS consumes more time. The main reason is that the algorithm requires large prime P and Q, they are randomly generated in the implementation process, and they need to be judged whether they are prime numbers and whether they are prime. The key generation algorithm based on chaos has only iterative operation, so the speed is the fastest. In the literature [16], the key generation time depends on the operation speed of the block encryption algorithm, and the AES algorithm is selected in the experiment. In the experiment, the key structure of the key generation algorithm based on wonton is simple: 28AAAA2A2AA88AA22A2A88A88A8AA8A8, and other key structures generated by the algorithm is more complex, such as: DD360A2D2C961EE48465F673035CD67F, in contrast, the program has certain advantages.

B Signature Algorithm Comparison

Nowadays, the signature algorithm is more. In this scheme, the commonly used RSA512 signature, ELGamal signature and SM2 signature are compared with the time and verification time of the NTRU (N=107) signature of this scheme, as shown in figure 4.

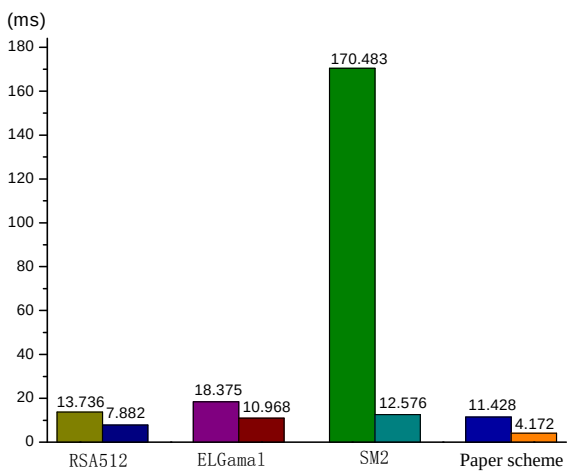

FIGURE IV. COMPARE DIGITAL SIGNATURE AND VERIFICATION

In the experiment, the signature data is the same, the factors that affect the efficiency of the signature, as well as the efficiency of the algorithm itself, the experimental environment, the realization of the platform, etc. The elliptic curve public key cryptography (SM2) takes more time in the process of signature and verification, while RSA512, ELGamal and NTRU consume less time, and the NTRU signature and verification efficiency is the best.

V. SUMMARY

In order to solve the problem of confidentiality and integrity of outsourced database. In this paper, we propose a secure and efficient key management scheme based on database encryption and NTRU signature. In this scheme, the NTRU signature is carried out after data encryption, and then stored in the outsourced database, and the data integrity is verified firstly when the data is decrypted, and that is to say, the validity of the NTRU signature is verified. The paper also analyzes the security of the scheme, and proves that it can resist the attack. The efficiency of the key generation algorithm and NTRU signature is compared by experiments. The experimental results show that the proposed scheme is efficient and secure.

REFERENCES

[1] Hacigumus H , Iyer B R, Mehrotra S .Providing database as a service[C] Proc of ICDE .Washington :IEEE C omputer Society , 2002 :29-40

[2] MI Sarfraz , M Nabeel , J Cao , E Bertino.DBMask: Fine-Grained Access Control on Encrypted Relational Databases[A].CODASPY '15 Proceedings of the 5th ACM Conference on Data and Application Security and Privacy[C].ACM New York, NY, USA ©2015.P.1-11.

[3] Sun X H. A Secure Scheme of Key Management for Database Encryption [M]// Advances in Technology and Management. Springer Berlin Heidelberg, 2012:315-319.

[4] Zongmin C, Lifen Z, Guangyong G, Caixue Z,Anyuan D. Efficient Key Management of Data Owner for Cloud Scenarios[J]. Journal of Computational Information Systems.(2015) 7693–7700

[5] Weis J, Alves-Foss J. Securing Database as a Service: Issues and Compromises [J]. IEEE Security & Privacy Magazine, 2011, 9(6):49-55.

[6] Vimercati S D C D, Foresti S, Jajodia S, et al. Encryption policies for regulating access to outsourced data[J]. Acm Transactions on Database Systems, 2010, 35(2):78-78.

[7] CHENG Fangquan, PENG Zhiyong, SONG Wei, et al. Certificateless authentication for trusted key sharing in trusted database. Journal of Frontiers of Computer Science and Technology, 2010, 4(9): 791-802.

[8] LI Jin-Guo,TIAN Xiu-Xia,ZHOU Ao-Ying. Privacy preserving fuzzy keyword research in Database as a Service paradigm [J]. Chinese Journal of Computers, 2016(2).

[9] Zhu Qin, Luo Yishu, Le Jiajin. Protecting right of Outsourced Database using digital watermark [J]. Journal of Computer Research and Development, 2006, 43(z3):212-218.

[10] Narasimha M , Tsudik G .DSAC: Integrit y of out sourced databases with signature aggregat ion and chaining [C] Proc of the ACM Conf on Information and Knowledge Management.New York :ACM , 2005 :235-236

[11] Xie Min, Wang Haixun , Yin Jian , et al.Integrity auditing of outsourced data [C] Proc of VLDB 2007 .New York :ACM ,2007 :782-793

[12] Xian Hequn,Feng Dengguo. An integrity checking scheme in Outsourced Database [J]. Journal of Computer Research and Development, 2010, 47(6):1107-1115.

[13] ZHAO Chun-hong, LIU Guo-hua, WANG Ning,et al. Text data integrity detection scheme in Outsourced Database Model[J]. Journal of Chinese Computer System, 2010, 31(9):1790-1796.

[14] Jaulmes E, Joux A. A CHOSEN CIPHERTEXT ATTACK ON NTRU [J]. Crypto ’00 Lncs, 2000.

[15] Hoffstein J, Howgrave-Graham N, Pipher J, et al. Practical Lattice-Based Cryptography: NTRUEncrypt and NTRUSign[M]// The LLL Algorithm. Springer Berlin Heidelberg, 2009:349-390.

[16] Yu Xiangxuan,Ni Xiaojun.The key management of an encrypted DBMS[J]. J.Huazhong Univ. of Sci . & Tech.1995,23(7):52-55.