RESEARCH Open Access

Feasibility study on security deduplication of medical cloud privacy data

Huiqi Zhao[1,2], Lexia Wang[1], Yinglong Wang[2*], Minglei Shu[2] and Jimin Liu[1]

Abstract

In view of the problem of resource waste caused by the repeated storage of medical data, our scheme proposes that convergent encryption is applied to the medical cloud, and the hash value of the convergent key is used as a repeat detection label and Bloom filter search to achieve the efficiency of the medical cloud, and the practicality of medical data is improved by introducing fuzzy keywords searching, ensuring the safe storage and correctness of medical data through local validation. At the same time, based on the actual application of different medical data, we use the authorization access method of identity token to realize the multi role access of medical cloud.

Keywords: Medical cloud, Convergence encryption, Fuzzy keyword search, Deduplication, Identity token, Message authentication code(MAC)

1 Introduction

In order to protect the security of these medical privacy data, the traditional scheme puts forward the attribute-based access control [1], which is a more flexible access control mechanism; has the advantages of strong scalability and fine granularity access; and is suitable for complex and changeable cloud environment. However, there are some problems in the traditional attribute-based access control mechanism, such as low encryption efficiency, lack of continuity, and low efficiency of strategy retrieval. At the same time, because of the flexibility of attribute access control model, the strategy conflict detection and resolution in the model is also an urgent problem. Jia-shun [2] introduces the concept of using the obligation in the control model based on the traditional access control, adds the state mechanism, realizes the continuity and the state relativity of the decision, and has the characteristic of dynamic multiple authorization. At the same time, the concept of linked list and balanced binary tree is introduced, and two kinds of static conflict detection methods based on conflict detection list and conflict detection tree are proposed, which transforms the conflict detection problem into the data structure problem of query overlapping path. However, the scheme only considers the security privacy protection mechanism of access control, ignoring the importance of encryption for the security protection of medical data. Then, in the access control scheme which is based on the PBAC access control model and the ABE encryption technology, a data access control method based on attributed-based encryption is adopted in this paper [3], and the attribute encryption technology is applied to the access control process. On the one hand, the security of medical data is realized in the access control layer; on the other hand, the security protection [4] of medical data is realized on the encryption level. Based on the application of medical cloud, this paper puts forward a research on data protection of medical cloud system based on attributed-based encryption [5], using attributed-based encryption to restrict the access of doctors, so as to ensure the security and privacy of medical data. In addition, the decision access structure and multithreading implementation are adopted to improve the efficiency problem. Later, the research on data sharing scheme based on privacy protection in medical cloud [6] adopts an improved aggregation key encryption algorithm (Improved Key-Aggregate Encryption; IKAE) to control the user's access rights. The authorization center can complete the access authorization for the user by simply sending a constant

* Correspondence: wangylscsc@126.com
[2]Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, Shandong, China
Full list of author information is available at the end of the article

size aggregation key to the authorized user. Compared with attributed-based encryption, the access efficiency is greatly improved [7] and the complexity of key management is greatly reduced. But with the massive increase in medical data and the increasing number of duplicate medical data, we have to consider applying ridding unbalance to the medical cloud to take full advantage of the storage space of the medical cloud and reduce the transmission bandwidth of unnecessary medical data. However, it is very difficult for the medical data to be repeatedly detected on the medical cloud after the encryption method is encrypted locally. This is because the uniqueness of the local key causes the same plaintext data to produce different ciphertext. However, if all senders use the same key to ensure the consistency of the ciphertext to facilitate the medical cloud ridding unbalance, it will certainly bring some security threats [8].

The contradiction between the encryption of medical data and ridding unbalance cloud data always exists before the convergence encryption [9]. The so-called convergence encryption is to define the hash value of the medical data file as the key of the file symmetric encryption. This allows the same file to remain the same encryption key for each user with a different private key, thus ensuring that the encrypted file is kept the same as the encrypted file, thus resolving the contradiction between encryption and ridding unbalance. But the simple convergence encryption algorithm can easily encounter a violent crack and guess an attack and so on. To optimize the performance, the file label is no longer defined as the hash value of the data encryption ciphertext, but the hash value of the convergence key, which is proposed in the randomized convergent encryption scheme (Randomized Convergent Encryption; RCE) [10]. This greatly saves the tag generation overhead. However, this also poses a new threat to the data security of the medical cloud-replica forgery attack, and the integrity of the file is not guaranteed. To protect the integrity and privacy of medical data on unreliable medical clouds, encryption prior to data upload and validation of data returned by the cloud server is particularly important. Moreover, even if the encryption is applied before the data is uploaded, it does not mean that the data we store in the medical cloud will not be maliciously altered. In view of the cloud server's unreliable and inquisitive service attitude, the cloud server may be, to save computational overhead and network bandwidth, only in partial operation, returning a part of the results, or the cloud server may maliciously delete and modify some files, plus malicious adversary attacks, causing users to receive unreliable data files. Therefore, it is necessary to verify the integrity of the results returned by the cloud server.

The necessary encryption is required before data is uploaded, making efficient retrieval of ciphertext data a challenging problem. A common solution is to search for encryption [11], which allows the server to retrieve encrypted data from the user and the server will not get any plaintext messages. The first symmetric searchable encryption scheme is the SWP scheme proposed by Song et al. In this scheme, the server can scan and compare ciphertext words and ciphertext files to confirm whether the key words are in the file, and even count the number of keyword occurrences. However, with the increasing of data in the database, the search cost is increasing linearly, resulting in low efficiency. Later, the method proposed by Goh used the Bloom filter to construct an index for each file, which greatly reduced our search cost for ciphertext files.

Summarizing the above problems, aiming at the massive data duplication caused by the medical data mismanagement in medical cloud, which leads to the waste of storage resources and bandwidth resources, we apply the convergent encryption to the medical cloud and use the hash value of the convergent key as the label of duplicate detection to save the cost of generation. In the first scenario, the Bloom filter is combined to improve the efficiency of PHR accurate search for medical data. In the second scenario, we introduce a private cloud to store the secret key, which improves the security of the key, while the private cloud shares some local computing work, further saving local resources. Taking into account the actual situation, in the medical system, the patient goes to the hospital to see a doctor; in the process, doctors generally with medical instrumentation results, combined with the doctor observation inquiries and causative agent complained, get the patient's illness set description. These symptom sets describing the symptoms of the disease often contain more vague, inaccurate digital and text medical information, which is the final formation of electronic medical records, in order to accumulate medical diagnostic knowledge. When we search the cloud for medical data such as electronic medical records, we allow users to perform all possible fuzzy searches [12, 13] with errors in search, rather than having accurate keywords to carry out relevant searches, so that medical data such as electronic medical records can be used to maximize medical value. Finally, understanding the importance of medical data correctness, for the cloud server is not fully trusted, we find a solution of medical data download to the local after the integrity of the validation, to ensure that the use of medical data is as correct as possible, to most likely reduce the loss of data error.

2 Preliminary knowledge

This section introduces the symbolic instruction and key techniques and algorithms introduced in this paper (Tables 1,2, 3, 4, 5, 6, 7, 8, 9, and 10).

2.1 Symbol description

Symbol	Definition
P	PHR, a simple representation of a document
E	EHR, a simple representation of a document
C_p	PHR, encrypted files for documents
C_E	EHR, encrypted files for documents
T_{W_i}	The trap gate computed by the keyword w_i
PID	ID of PHR document
EID	ID of EHR document
$C_p = \{C_{p_1}, C_{p_2}, C_{p_3}, \ldots C_{p_N}\}$	N collections of PHR-encrypted documents
$C_E = \{C_{E_1}, C_{E_2}, C_E, \ldots C_{E_N}\}$	N collections of EHR-encrypted documents
$t(0 \le t \le T)$	Timer, starting from 0 times to T
Key	Identity token for non-data owner
C_{k_p}	Convergence key for PHR-encrypted document
C_{k_E}	Convergence key for EHR-encrypted document

2.2 Technical introduction

This paper briefly introduces the key technologies and algorithms in the proposed scheme based on medical environment.

2.2.1 Convergence encryption

Convergent encryption is a deterministic encryption algorithm, which is essentially a special symmetric encryption scheme, the special point is that the encryption key is not a user's private decision, but rather is determined by the text to be encrypted, using the hash value of the text as the encryption key, thus guaranteeing that, although in different user encryption, as long as the same text is to be encrypted, the same cipher can be obtained. Based on this feature, the symmetric encryption method of convergent encryption is widely used in the heavy scheme of all kinds of encrypted data.

A basic convergence encryption scheme consists of the following four algorithms:

Table 1 Algorithm of generation Keys

Algorithm 1: Convergence key generation algorithm
1. Enter the text M to be encrypted
2. Calculation key: *CE. KeyGen(M) = > K*
3. Output key K

Table 2 Algorithm of encryption

Algorithm 2: Convergent encryption algorithm
1. Enter the convergence key K and the text to be encrypted M
2. Computational ciphertext based on cryptographic algorithms *C: CE.Enc(K, M)= > C*
3. Output ciphertext C

2.2.2 Bloom filter

Bloom filter [14, 15] is a kind of a high spatial efficiency random data structure, which uses a bit array to represent a set, consisting of a long bit array and a series of random mapping functions.

The Bloom filter can quickly determine whether an element is in a set or is an array of m bits, as each position is initially initialized to 0, using R-independent hash functions $h_1, \ldots, h_r$, among them, $h_i : \{0, 1\}^* \rightarrow [1, m]$, $i \in [1, r]$. For a collection of n elements, $\boldsymbol{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots \boldsymbol{S}_n\}$,

where each of the elements in S is mapped by the r hash function to $h_1(s)$,$h_2(s)$, …, $h_r(s)$ and replace the position of the array $h_1(s)$,$h_2(s)$, …, $h_r(s)$ with 1. To find out if an element a belongs to S, just take a into the R hash function to get $h_1(a)$,$h_2(a)$, …$h_r(a)$. Then check $h_1(a)$,$h_2(a)$, …$h_r(a)$ whether the position is 1 in the m-bit array, and if so, prove $a \in S$; otherwise, $a \notin S$.

Of course, because of the hash collision, each bit of the array could be caused by several hash functions. Therefore, the Bloom filter in this kind of judgment has an error rate.

2.2.3 Edit distance [16]

Edit distance is a measure of string similarity. The editing distance between two strings refers to the smallest number of edits required to convert from one to another. The editing operations here include character insertion, character substitution, and character deletion, where substitution converts one character to another, deleting removes a character from a word, and inserting inserts a character into a word. Based on the above three basic operations, we can define the following set: For a given keyword W, $S(W, S)$ indicates that all keyword W remains within the edit distance d of the keyword set, that is, each keyword in the collection satisfies $ED(W, W^{'}) \le d$ (d is an integer).

2.2.4 Hash function [17]

A hash function, also called a hash function or a hash function, is a public function that maps any long

Table 3 Algorithm of decryption

Algorithm 3: Decryption algorithm
1. Input convergence key and ciphertext C
2. Execution algorithm: *CE.Dec(K, C)= > M*
3. Output: Clear text M

Table 4 Algorithm of generation tag

Algorithm 4: Tag generation algorithm
1. Enter the convergence key *K*
2. Execution algorithm: *CE.TagGen(K)=* > *Tag*
3. Output: Label *Tag*

message to a shorter, fixed-length value as a qualifier, called a hash value, hash code, or message digest. A hash value is a function of all bits in a message, thus providing an error detection capability that alters the hash value by altering any bit or bits in the message. The definition of the hash function is given below:

Function $h = H(M)$ satisfies:

(1) The length of the input M is arbitrary, and the length of the output $H(M)$ is fixed;
(2) The forward calculation is easy, that is, given any M, it is easy to calculate the $H(M)$, inverse calculation difficulty, that is, give an $H(M)$ value, find the input M, so that $h = H(M)$ which is not feasible in the calculation;
(3) Given an input M, finding $M^{'}(M^{'} \neq M)$ makes $H(M) = H(M^{'})$ which is infeasible to compute, and finding any two different inputs $M_1$and M_2 make $H(M_1) = H(\ M_2)$ which is computationally infeasible.

Such a function is called a hash function.

Hash function is mainly used for digital signature, message integrity detection, message origin authentication detection, and so on.

2.2.5 Message Authentication Code [18]

The Message Authentication Code (MAC) is a short message that validates the message and provides integrity and authenticity to the message. In our scheme, we apply the message authentication code to realize the integrity of the cloud medical data. It is easy to construct a hash function; by entering a user's private key and an arbitrary length of the file, you can get a MAC. A hash is obtained by the downloader based on its own private key and the received file, and if the hash is equal to the MAC of the file, the integrity and authenticity of the file is guaranteed. Otherwise, it indicates that the file has been modified.

Table 5 List of identity tokens

Identity tokens	T (timer)
Key_1	t_1
Key_2	t_2

2.2.6 Ownership certificate [19–23]

For the first time, Halevi and others introduce the concept of provable ownership (Proof of Ownership; PoW). This technique is used to provide data privacy and confidentiality in the context of a user-side-heavy scenario. In other words, users can prove to the server that they really own a file without having to upload the entire file. The essence of PoW and PDP is to achieve data integrity verification, where the difference is that the PDP is the cloud server to the user to prove the integrity of the file, and PoW is the user to the cloud server to prove the integrity of the file (the authenticator and the authenticator identity interchange).

3 System models

In program I, a medical cloud system [24] consists of three types of entities: medical cloud servers, patient roles, doctors, or medical institutions.

In program II, a medical cloud system consisting of five entities of medical cloud servers, private cloud servers, medical personnel, patient roles, and other authorized units is studied.

A medical cloud server entity is responsible for receiving the medical data uploaded by the private cloud or other role entities, storing various medical data documents and checking to ensure that the storage is unique.

Patient role: In program I, the patient belongs to the data owner, has the direct operation authority to the data, through the medical cloud's after the examination, is responsible for uploads the local medical data, and protects the medical cloud data's uniqueness. At the same time, the patient through the keyword search can get the data on the cloud. In program II, the patient no longer belongs to the data owner and cannot upload the medical data, even though the data originates from the patient; although the patient cannot directly access the data which searches the cloud, the patient may access it through the medical personnel who carries the keyword fuzzy search to carry on the data understanding [25].

A doctor or medical institution is not the owner of the medical data, but in the case of a certain demand, the doctor or the medical institution may apply to the data owner (patient) for authorization to obtain the identity token so that it can be uploaded through the patient and the keyword search.

The private cloud server, save the user's private key and a public key, is the place for encryption, trap calculation and file tag calculation.

Other authorized units: In program II, it does not belong to the owner of the data, but after obtaining the identity token under the authorization of the medical staff (data owner), the keyword search can be conducted through the medical personnel.

Table 6 Total time overhead for files without duplicates

File size/KB	64	128	256	512	1 K	2 K	4 K	8 K	16 K	32 K
Baseline/s	0.501	0.602	0.603	1.001	1.203	3.102	4.500	9.001	20.551	53.104
BF/s	0.495	0.585	0.602	0.6	0.855	0.995	1.187	3.064	7.552	19.895

4 Keyword search scheme based on multi role access for secure duplication in medical cloud

In our program, we set up the medical cloud system based on two different medical data of PHR and EHR according to different actual situations [26–28].

4.1 PHR [29] medical cloud [30] application system

Basic thought: patients have stored n initial medical data $PHR = \{P_1, P_2, P_3, \cdots, P_N\}$ in the medical cloud, wherein each file has a corresponding file label. When a patient uploads his or her own, he or she needs to define a set of keywords for each one, forming a different set of keywords on the medical cloud. The patient uses the local key to encrypt the data and then uploads the medical cloud. Later, users continue to upload a medical data, where he needs to check first, as there are backups that do not have to be repeated for upload, otherwise, defined as the initial upload.

4.1.1 System initialization

Step 1: On the client side, or the patient end, the convergence encryption algorithm is used to converge the local n PHR document: (1) To execute the convergence key generation algorithm, the convergence key generation algorithm used in this scheme is $K_P = H0(P)$, where the patient enters the PHR document to be encrypted, according to the algorithm that can obtain each PHR document convergence key, as k_p. (2) In the implementation of the encryption algorithm $C_P = Enc_{CE}(k_P, P)$, the patient input convergence key is k_p which is the PHR document to be encrypted, and the output of the encrypted PHR document is C_P (3) For each copy to be uploaded, the encrypted PHR document performs the two steps above to get $C_P = \{C_{P_1}, C_{P_2}, C_{P_3}, \cdots, C_{P_N}\}$.

Step 2: The patient indexed these n local encrypted documents:

(1) The patient input the above to upload the encryption PHR. The document keyword is set as W , the client system carries on the trap gate T_W, calculates $T_{w_i} = f(sk, w_i)$, enters the user private key and the key word, and calculates the trap gate under the above algorithm effect.

(2) The patient will calculate the T_W inserted to the Bloom filter as an index and the document identity PID with the Index binding:

Define r hash function $h_1(PID, T_w)$, $h_2(PID, T_w)$, $\cdots$, $h_r(PID, T_w)$, enter the trap gate, and under the action of N hash function get the code word (position code) of the trap gate in $y_1 = h_1(PID, T_w), \cdots, y_r = h_r = (PID, T_w)$, place the r position above 1.

Step 3: PHR file label page calculation: According to the algorithm Page = $H(k_P)$, with the convergence key as input, hash operation is used to get the document label Page. Each document to be uploaded has its own label, which corresponds to each one.

Step 4: Encryption Convergence key: $C_{kp} = Enc(sk, k_p)$

Step 5: Upload the encrypted N PHR files, the encryption of the Convergence key, index and file tags to the medical cloud server.

4.1.2 Uploading of PHR documents based on patient roles

Based on the patient role, they will have ownership of their private PHR document, that is, the patient is unique owner of the PHR document, who can determine what information is stored in the PHR document and decide who can access that data.

First, when a patient is unaware of a medical cloud store and wants to store its PHR in a medical cloud, the medical system will repeat the test to determine whether to perform PHR.

The patient uploads PHR file tag Page, which is intended to interact with the medical cloud to repeat the check:

Step 1: PHR Document label calculation formula: $Page = H(k_P)$ (where the convergence key is generated in the same way as above)

Step 2: The patient uploads Page medical cloud: the medical cloud compares the upload tag Page

Table 7 The time cost of ownership proof based on BF

File size/KB	64	128	256	512	1 K	2 K	4 K	8 K	16 K	32 K
BF/s	0.213	0.201	0.441	0.445	0.500	0.892	1.117	2.865	6.558	14.556

Table 8 Time cost of cryptographic convergence secret key

Repetition rate/%	0	25	50	75
Baseline/KB	80	80	80	80
BF/KB	80	60	40	20

Table 10 Construction of index

Keyword quantity	1000	2000	3000	4000	5000	6000
Fuzzy/ms	6.721	8.899	10.006	14.622	16.665	20.007
General/ms	4.001	4.998	5.668	9.362	11.411	14.996

whether the cloud server already exists to determine whether the file already exists.

Step 3: If the comparison exists, the same file already exists on the medical cloud server. However, there is a possibility of a hash collision based on the patient's Page calculation, and in order to prevent illegal users from exploiting this vulnerability to gain access to the document, the patient needs to further document ownership to prove that he or she owns the *PHR* document. If the proof does exist, the patient is allowed to assign pointers directly to access the same *PHR* document on the medical cloud server without having to repeat the upload.

Step 4: Otherwise, the patient realizes the encryption upload and index construction operation of the new PHR document.

(1) Encryption upload stage: The same as the system initialization phase, using the convergence key to encrypt the new upload *PHR*, calculate $k_P = H0(P)$and $C_P = Enc_{CE}(k_P, P)$, while calculating the file label Page = $H(k_P)$ and encryption of the convergence key $C_{k_P} = Enc(sk, k_P)$.The encrypted *PHR* document, the encrypted convergence key, and the label are uploaded after the calculation is completed.

(2) Indexing stage: According to the uploaded *PHR* keyword *W* and the patient's unique private key *sk* calculates the trap gate $T_{w_i} = f(sk, w_i)$, then inserts the trap door into the Bloom filter to form the updated index and binds it to the *PID*. As with the initialization phase of the index creation, this is no longer detailed here.

4.1.3 Uploading of PHR documents based on series roles such as medical institutions

Doctors or medical institutions essentially do not have information ownership for a PHR document. However, in reality, sometimes doctors or medical institutions also need access to the patient's PHR document, so there is a situation in which a patient's PHR document can be accessed after the consent authorization is granted.

The authorization of a patient's role to a physician or a medical institution consists of two parts: the first part is a shared hash function; the second part is the identity token key that the patient assigns to the doctor or the medical institution with the third party identification effect, and the patient himself keeps a list of identity tokens. Each identity token in the list is equipped with a timer t ($0 \leq t \leq \mathrm{T}$), which is automatically cleared in the list when the timer t number reaches a certain size of T.

First, the doctor or the medical establishment calculates the $k_P = H0(P)$, $Page = H(k_P)$to get the label Page of the document to be uploaded locally based on the hash function obtained by the shared patient. Upload tag Page to the medical cloud server and compare to find out if such a label exists to determine whether the uploaded file has been duplicated in the medical cloud.

If present, it indicates that the same file already exists on the medical cloud server. As with patients, doctors or medical institutions need to further document ownership to prove that they do own the *PHR* document. If it proves to be true, the medical cloud returns to a physician or medical institution, and then the doctor or medical institution uploads the identity token Key, and the medical cloud assigns a time-sensitive pointer to the same *PHR* document on the medical cloud server after identifying the identity token Key exists in its own saved list of Identity tokens.

Otherwise, the doctor or medical institution realizes the encryption uploading and indexing of the new PHR documents.

(1) Encrypted upload: The same as the patient role, using the convergence key to encrypt the new upload PHR, calculate $k_P = H0(P)$and $C_P = Enc_{CE}(k_P, P)$;

(2) Index construction: After a physician or medical institution's identity token is present at the patient's end, the physician or medical institution will input in the PHR the uploaded keyword *W* through the patient's unique private key sk for $T_{w_i} = f(sk, w_i)$, and then return to the doctor or medical institution, and then, like the patient, use the *R* hash function $h_1(PID, T_w)$, $h_2(PID, T_w)$, , $h_r(PID, T_w)$ to get the position code in the Prum filter and place each position in 1 to form an index.
The authentication request is shown in Fig. 2 and the de-reupload process is shown in Fig 3.

Table 9 Space overhead of cryptographic convergence secret key

Repetition rate/%	0	25	50	75
Baseline/KB	80	80	80	80
BF/KB	80	60	40	20

4.1.4 Keyword search for patients

Step 1: The patient provides the private key with the keyword to compute the trap gate T_W—as a search request to submit the medical cloud server;

Step 2: The medical cloud server will receive the trap gate T_W as the input calculation result of the $h_1(PID, T_w)$, $h_2(PID, T_w)$, , $h_r(PID, T_w)$ R hash function as the position code in the Prum filter. The medical cloud calculates the position code to find the corresponding position of the server-side Prum filter. According to the position code, if the search result is all 1, then it proves that the cloud *PID* contains the keyword and returns the patient PID; otherwise, the search fails.

4.1.5 A keyword search for the role of a physician or medical institution

After obtaining a patient's search authorization, the physician or medical institution will be allowed to temporarily pass the patient's trap calculation to submit the medical cloud server as a search request. The search process is ditto.

Step 1: First, the doctor or medical institution obtains the identity token Key with the third party identity effect after obtaining the patient's search authorization.

Step 2: If a physician or medical institution wishes to search for a keyword, it should first submit the keyword and all its identity tokens Key to the medical cloud system.

Step 3: The medical cloud system will search the keyword and identity token Key to the patient end, the patient end after verifying identity token will carry out the trap door T_W calculation: $T_{W_i} = f(sk, w_i)$, and then the trap door back to the doctor or medical institutions.

Step 4: Next, as with the patient search, the doctor or medical institution submits the trap door to the medical cloud server first.

Step 5: The medical cloud server after taking over to the trap door, the same as the patient search for the operation of R hash function to get the location code, detects whether the layout of the corresponding position in the filter is all 1; this is no longer explained in detail.

4.1.6 Download and integrity verification of PHR documents

When the patient or the doctor or the medical institution will upload the medical cloud server download request and *PID*, the medical cloud to the requester authentication confirm that it is eligible to download the PHR document.

After the authentication passed, the medical cloud issued $C_P = Enc_{CE}(k_P, P)$ and $C_{kp} = Enc(sk, k_p)$ to the requester local.

1. If the patient then uses the local key to decrypt, the convergence key k_P is obtained and if the convergence key is used to decrypt, the plaintext *PHR* is obtained.
2. If the doctor or medical institution will submit the identity token k_P and $C_P = Enc_{CE}(k_P, P)$ as before, $C_{kp} = Enc(sk, k_p)$ to the patient end; after the authentication, the patient will use the private key to decrypt $C_{kp}$to obtain the convergence key k_P, and then use the convergence key k_P decryption eventually to get a clear PHR document and return it to the doctor or medical institutions local.

Finally, the local computation decrypts the plaintext *PHR* which is the hash value of *H(P)* and compares whether it is equal to decrypt the obtained convergence key k_P

(1) Equal, the certificate received the correct and complete *PHR* document;
(2) Otherwise, it is proved that the received PHR is an error *PHR* that has been maliciously changed.
The keyword search process is shown in Fig. 4.

4.2 EHR [31] medical cloud application system

4.2.1 System initialization

The model uses the "medical staff—private cloud (holding private key)—Medical cloud storage" role chain, where the medical personnel no longer holds the private key, in turn to private cloud security preservation.

The medical cloud stores an index of n-Encrypted *EHR* documents with medical personnel and a different set of keywords for these *EHR* documents:

(1) Encrypting *EHR* documents: On the client side, that is, the medical personnel to upload the *EHR* document to the private cloud, the private cloud performs the convergence encryption algorithm to the uploaded n *EHR* document to converge the encryption: Executing the convergence key generation algorithm, this scheme uses the convergence key generation algorithm as $k_E = H(H0(E), K)$. The *EHR* document and private cloud encryption public key are encrypted as input of the algorithm, and the convergence key $k_E = H(H0(E), K)$ of each EHR document can be obtained according to the algorithm.
② Executing the cryptographic algorithm $C_E = Enc_{CE}(k_E, E)$, the convergence key k_E and the EHR document to be encrypted is entered and outputs the $C_E = Enc_{CE}(k_E, E)$ of the encrypted EHRdocument.
③ It is $C_E = \{C_{E_1}, C_{E_2}, C_{E_3}, \cdots, C_{E_N}\}$ to perform

both of these steps for each of the encrypted EHR documents to be uploaded.

(2) Index construction: ① the medical personnel should first set the editing distance d, and then upload the keyword W and EID to the private cloud; ② private cloud first calculates the keyword fuzzy set W = $\{w_1, w_2, w_3, \quad, w_i\}$ according to the medical personnel preset editing distance and keyword; Then, the private key is used to compute the trap gate $T_W = \{T_{W_1}, T_{W_2}, T_{W_3}, \cdots, T_{W_i}\}$ for each keyword in the fuzzy set, and the EID for each keyword is $T_{w_i} = f(sk, w_i)$ according to $\mathrm{Enc}(\mathrm{sk}, EID_{w_i})$. Finally, the computed trap is correlated with the encrypted EID according to the corresponding keywords to get the index table $\{T_{W_i}, \mathrm{Enc}(\mathrm{sk}, EID_{W_i})\}$.

(3) Private cloud using the user's private key SK encryption convergence key: $C_{k_E} = Enc(k_E, sk)$

(4) Private cloud uses the convergence key to compute the label of the EHR Document: $Page = H(H(k_E), K)$

(5) Private cloud upload a series of encrypted file $C_E = \{C_{E_1}, C_{E_2}, C_{E_3}, \cdots, C_{E_N}\}$, each file label PAGE and its respective computed index table $\{T_{W_i}, Enc(sk, EID_{W_i})\}$ and encrypt the convergence key C_{k_E} to the medical cloud.

Private cloud holds the key pair (sk, K) and the convergence key k_E of the medical personnel.

4.2.2 Re-uploading of EHR files

The medical personnel uploads the EHRto the private cloud, through the private cloud computing after the convergence key k_E (the computation way ditto), then takes the input and unifies the user to encrypt the public key to carry on the hash operation that obtains the file the label $Page = H(H(k_E), K)$, in order to carry on the examination. In turn, the private cloud uploads the tag page to the medical cloud for repeated checks: if there is a duplication of the label comparisons, the certificate of ownership of the EHR is first shown and then assigned to the medical staff by a pointer to an *EHR* with the same label as the medical cloud;

Medical personnel will complete no repeat upload: As with system initialization, just complete the encryption of the EHR document, index creation, encryption of the convergence key, tag calculation, and the above four upload medical cloud.

4.2.3 Key words for medical personnel on electronic medical records EHR fuzzy search

Step 1: Medical personnel input search keyword W to private cloud and preset edit distance d;

Step 2: The private cloud computes the fuzzy set $W = \{W_1, W_2, W_3, \ldots, W_i\}$ using the keyword W and d, and then $T_{w_i} = f(sk, w_i)$ the fuzzy set to the $T_W = \{T_{W_1}, T_{W_2}, T_{W_3}, \ldots T_{W_i}\}$ and the calculated trap is uploaded to the medical cloud as a search request.

Step 3: The medical cloud carries on the comparison between the trap gate and the index, determining if there is a matching *EHR* document: If there is a match, return the corresponding encrypted EID in the Index Table $\{T_{W_i}, Enc(sk, EID_{W_i})\} Enc(sk, EID_{w_i})$, send it to the private cloud using the private key of the medical personnel sending the request to decrypt, and get the EID identifier, search success, otherwise, return search failed.

4.2.4 Patients fuzzy search for the key words of electronic medical records

Although the electronic medical record is of the patient's hospital diagnosis and treatment situation and so on records, the patient does not have the ownership to the electronic medical record. However, as long as the patient provides identification to the medical personnel, searching for patients with electronic medical records can be through the medical staff with the medical cloud keyword. The fuzzy searching of medical staff for the key words of the medical cloud is the same as part 4.2.3 aboved.

4.2.5 The keyword fuzzy search of electronic medical records by other authorized units

In some cases, the transfer of electronic medical records between different hospitals or between different departments of the hospital is not optional, and each call should be legally authorized. Therefore, the other authorized units on the electronic medical records of the keyword fuzzy search in the medical cloud should be allowed.

Step 1: First, the other departments to the ownership of medical personnel departments to apply for authorization, after the successful authorization, medical personnel departments to the application department issued a certain timeliness of identity token key.

At the same time, the medical personnel department will itself save a list of identity tokens, each identity token in the list is equipped with a time $t(0 \leq t \geq T)$, when the timer t number reached a certain size T will automatically clear in the list.

Step 2: Other units authorized to search the medical cloud for keyword ambiguity, need to submit their identity token key and search the keyword W to the ownership of the medical personnel system client.

Step 3: The medical personnel system client first detects whether the identity token key is legally authorized by itself (that is, whether it is in its own list of Identity tokens), and if so, can upload the keyword W to the private cloud.

Step 4: The private cloud uses the submitted keyword W and the preset edit distance d to compute the fuzzy set to get the $W = \{w_1, w_2, w_3, \ldots w_i\}$, then to the fuzzy set to carry on the trap gate computation $T_{w_i} = f(sk, w_i)$ obtains the $Tw = \{w_1, w_2, w_3, \ldots w_i\}$, calculates the trap gate namely as the search request uploads to the medical cloud.

Step 5: The medical cloud carries on the comparison between the trap gate and the index, determine if there is a matching *EHR* document: If there is a match, return the corresponding encrypted *EID* in the Index Table $\{T_{W_i}, Enc(sk, EID_{W_i})\}$ $Enc(sk, EID_{W_i})$, sent to the private cloud using the private key of the medical personnel sending the request to decrypt, get *EID* identifier, search success; return search failed.

4.2.6 EHR document download and integrity verification

Medical personnel, patients, or other authorized units send a download request to the medical cloud and download an *EID* of *EHR* for the enterprise.

The medical cloud first verifies the eligibility of the patient and confirms that it is eligible to download the *EHR* document;

After the authentication is passed, the medical cloud sends $C_E = Enc_{CE}(k_E, E)$and $C_{KE} = Enc(sk, k_E)$ to medical personnel, patients, or other authorized units locally, and medical personnel, patients, or other authorized units are decrypted by the private key of the medical personnel on the cloud to obtain a convergence key k_E(under the premise that the patient or other authorized entity has been authenticated),

Use the convergence key to decrypt the plaintext *EHR*.

Finally, for medical personnel, patients, or other authorized units, the hash value of the plaintext *EHR* obtained by the local computing decryption gets the message authentication code *(MAC)*= $H(E)$to determine if the convergence key k_E is equal: if equal, then the medical personnel, patients or other authorized units receive the correct and complete *EHR* document, to certify that an *EHR* received by a medical person, patient, or other authorized entity is a malformed EHR that has been maliciously altered.

5 The experimental method

5.1 Experimental target

1. Data confidentiality: We mainly consider the confidentiality of the data from three aspects: first, when a multiple identity role accesses a medical cloud system, the non-data owner is not able to access the data without authorization, and the second is that the attacker cannot acquire the secret data illegally by cracking the acquisition of the convergence key. Third, an attacker with partial file information is still unable to obtain access to a file for legal access.
2. Data integrity: Download to local medical data before use to be clear whether there is integrity and to ensure that the actual use of medical data is correct and complete.

5.2 Experimental steps

We proposed solutions to the above, in order to reflect the efficiency of our scheme and practical, we truly simulate large medical data simulation experiments, namely in the different phases of the scheme, random selection of $N(< = 1500)$ different sizes ($< = 32$ MB) of medical documents, full extraction M ($< = 6000$) different keywords. We use the C language on the Linux operating system to complete multiple experiments at different stages. Here, we compared the Baseline scenario mentioned in literature [20, 21] with a series of comparisons based on the Bloom filter we are going to adopt.

Experiment 1: in the experiment, we selected a large number of files between 64 KB and 32 MB.

From the diagram, the BF scheme efficiency is better than the baseline scheme, and especially when the file is larger, the advantage of BF is more obvious. When the file is first stored, because the server needs to initialize the baseline filter, it occupies a large amount of time overhead, so at this time, the BF scheme is less obvious than the efficiency of the scheme. But when the file is duplicated, the server needs to prove the ownership of the client, and the BF scheme has obvious advantages in efficiency.

Experiment 2: the introduction of BF has a great advantage in the encryption key. Because our solution client ultimately needs to upload the ciphertext of the key that is not a duplicate file, the small key space decreases linearly with the file repetition rate.

Experiment 3: compared with ordinary search, the time consumption of fuzzy search index construction is high, as shown in the figure below:

With the increase of the number of files, index build time also increased, as shown, in the process of time consumption, where fuzzy search is always higher than that of an ordinary keyword search; however, as the file size increases at the same time, the time consumption basically remains the same, and this shows that in the fuzzy search, the greater the number of files, the lesser the search to have disadvantages.

6 Results and discussion

From the analysis of statistical results, it is concluded that the introduction of BF, especially in the case of duplication of files, makes the efficiency of the file retrieval significantly improved, and also has great advantages in the encryption key, but the fuzzy search used in our scheme does require a greater price for the more accurate search in the index construction (Figs. 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10).

6.1 Risk analysis

In our program implementation, there must be a variety of potential risks that affect the outcome we expect. In the previous experiments on the use of wireless technology to obtain various data on physical health information, we expect all data to be true and reliable, but the error is inevitable. In the experiment, we carried out statistical analysis on the test data and selected the confidence interval reasonably according to the 95% confidence level, and then analyzed the absolute risk and relative risk. In different roles, for example, on a visit to medical data, based on an artificial operation, may lead to a wrong operation, and there is the possibility of human error to tamper with medical data and the data of the system to avoid the absolute risk. However, such operational risk is trivial compared with our expectations of the medical system (i.e., relative risk).

6.2 Discussion

Even though we have tried to ensure the performance of the entire medical cloud system in all aspects, there are still some unavoidable problems in our proposed scheme. Below, we briefly discuss some issues from the following three perspectives.

First, in the third party, when the non-data owner wants to access medical data of the medical cloud, the third needs to submit identification and access to the owner's system, but here, we have no better mechanism to ensure that the submitted proof is legal and cannot be forged. Secondly, we do not provide a better mechanism to ensure the security of the token list held by the owner of medical data. In other words, the illegal visitor is likely to modify the identity list by the intrusion owner system and save the identity information of the illegal visitors directly in the list, so that the illegal visitors can achieve legal access. In addition, for the cost of medical system, public cloud and private cloud will increase investment in cost. The building of a private cloud, the structure, and good planning are all a great problem in the process of practice.

7 Security analysis

7.1 Confidentiality of data

In the case of a medical cloud system with multiple identity roles, no authorized non-data owners will be

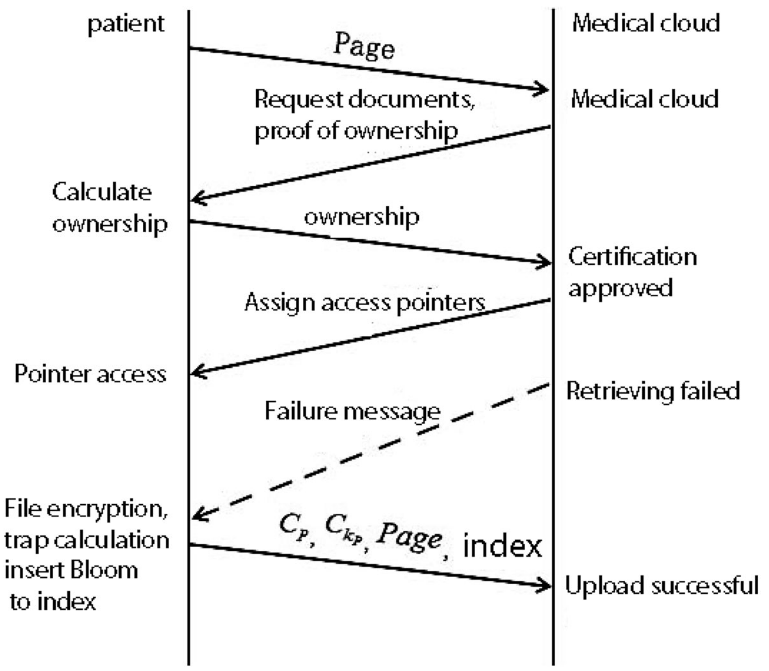

Fig. 1 The patient's re-uploading

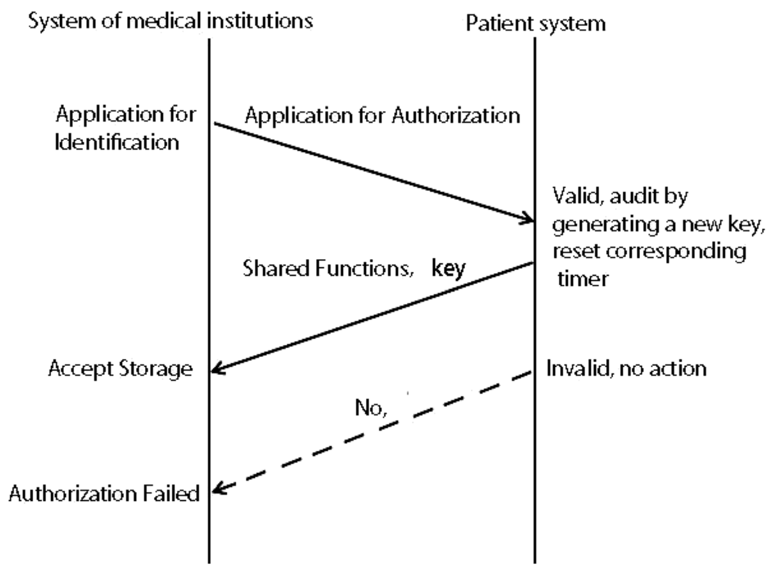
System of medical institutions
Patient system
Application for Identification
Application for Authorization
Valid, audit by generating a new key, reset corresponding timer
Shared Functions, key
Accept Storage
Invalid, no action
No,
Authorization Failed

Fig. 2 Request authorization process

able to perform the search for heavy detection or accurate or fuzzy keywords. For the access role for non-data owners to interact with and access the medical cloud data, the access request must be submitted to the data owner system prior to this, and the data owner will then be given the validity of an identity token by interacting with the legality of its identity. Visitors can access the medical cloud with this identity token.

However, visitors are not allowed to make permanent visits at once, because their identity tokens are valid for a period of time, which reduces the likelihood of identity tokens being stolen and provides secure access. In addition, it is impossible to get the plaintext information through the decryption of the convergence key, especially in our second scheme, in which an attacker may obtain a convergence key in two aspects. The first is that it is extremely difficult for an attacker to obtain a

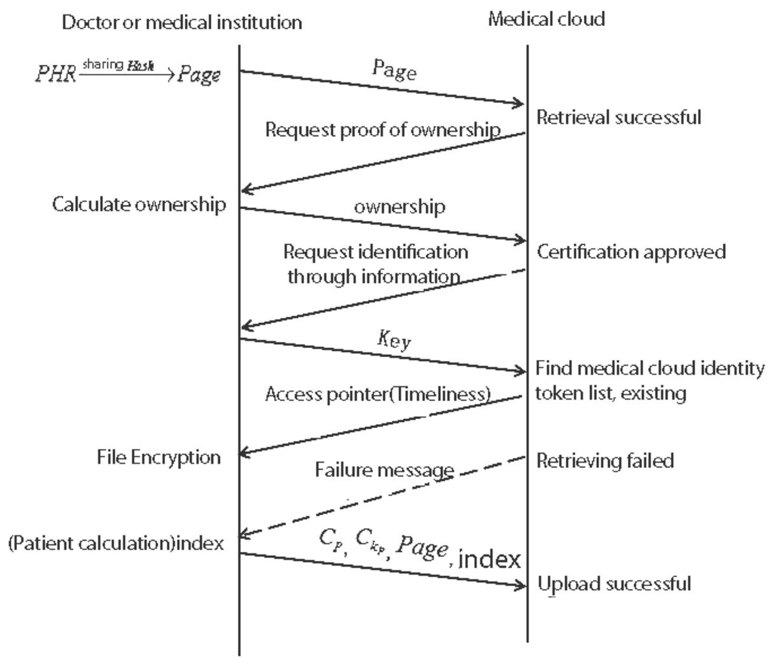
Doctor or medical institution
Medical cloud
$PHR \xrightarrow{\text{sharing Hash}} Page$
$Page$
Retrieval successful
Request proof of ownership
Calculate ownership
ownership
Certification approved
Request identification through information
Key
Find medical cloud identity token list, existing
Access pointer(Timeliness)
File Encryption
Failure message
Retrieving failed
(Patient calculation)index
C_P, C_{k_P}, $Page$, index
Upload successful

Fig. 3 A doctor or medical institution to upload

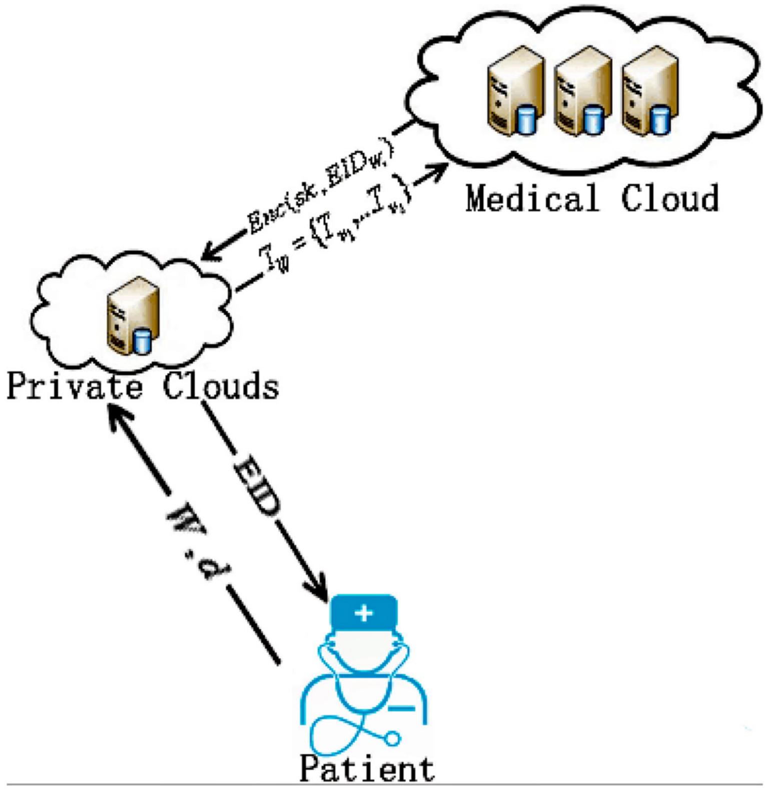

Fig. 4 The keyword fuzzy search of medical personnel

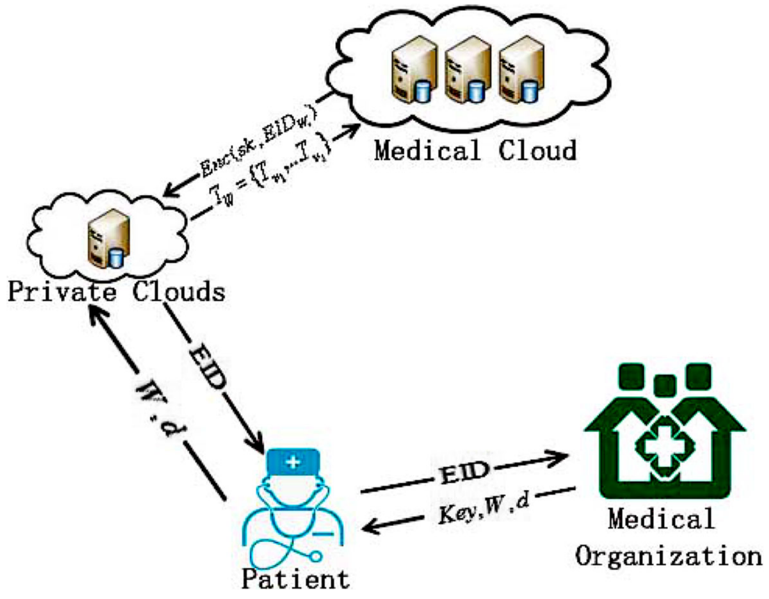

Fig. 5 A keyword blur search for other authorities

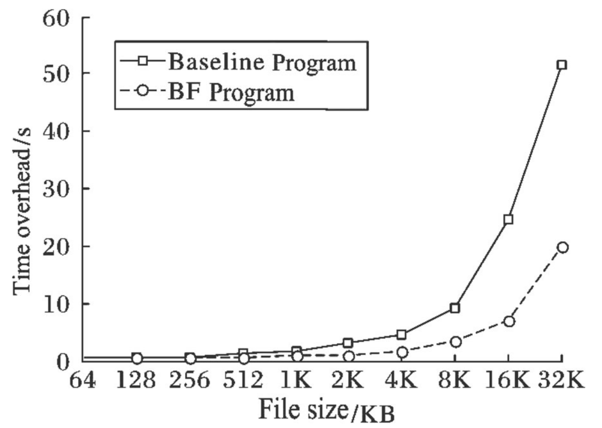

Fig. 6 Total time overhead for files without duplicates. —□—:It indicates the total time overhead of the baseline scheme without duplication of files. - -○- -:It indicates the total time overhead of the BF scheme without duplication of files

user's private key and decrypt it directly to obtain a convergence key. Because the private key protection is more rigorous, and in the second scenario, we put the private key more closely on the private cloud, so we think that it is impossible to crack the convergence key by obtaining the private key. The second aspect is that the attacker uses a hash collision to directly brute force the convergence key. First of all, considering the unknowns of the hash function plus the possibility of a function collision, we think this is a very low probability method. Moreover, medical data based on extraordinarily complex data types in the absence of clear text, the likelihood of a hash collision is almost non-existent, and in the second scenario, our convergence key is generated by two hash functions, which makes it possible, regardless of whether or not some of the plaintext information is available. It is not feasible to directly guess the convergence key. A third type of attacker is an illegal visitor who has some plaintext information and wants to access the entire information of the file by obtaining an access pointer.

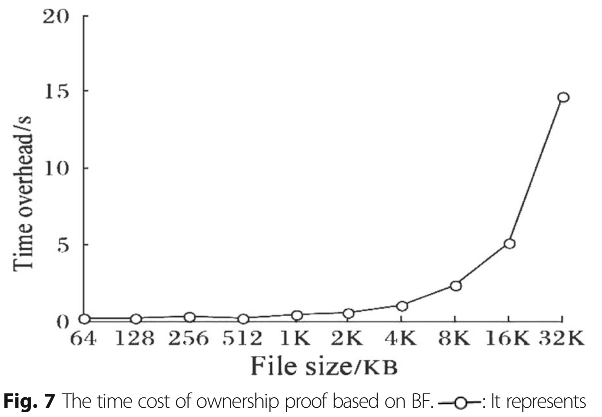

Fig. 7 The time cost of ownership proof based on BF. —○—: It represents the time cost of ownership proof based on BF

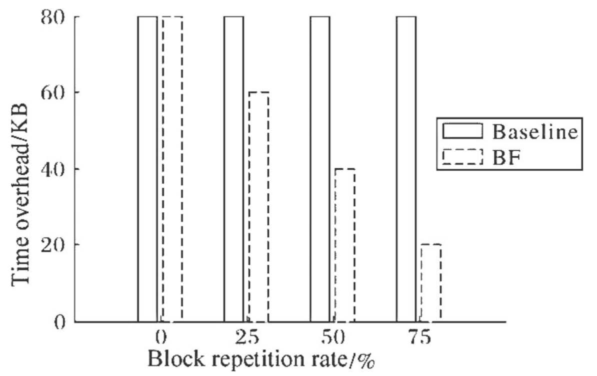

Fig. 8 Time cost of cryptographic convergence secret key. ▭: It represents the time overhead of the baseline scheme to encrypt the key. ⬚: It represents the time overhead of the BF scheme to encrypt the key

In this case, the attacker would have to meet the following conditions:

1. Having the label of the document
2. Through the Proof of Ownership Agreement (PoW) detection

The second condition will fail at a high probability when the attacker has only part of the information in the file (in the first scenario, the Prum filter is set to a reasonable size), and the POW protocol requests a sufficient amount of data information. However, we also consider that if the attacker owns most of the information in the file, it would be highly possible to verify through the POW protocol. However, when a user owns most of the information on a file, he can be identified as a legitimate user.

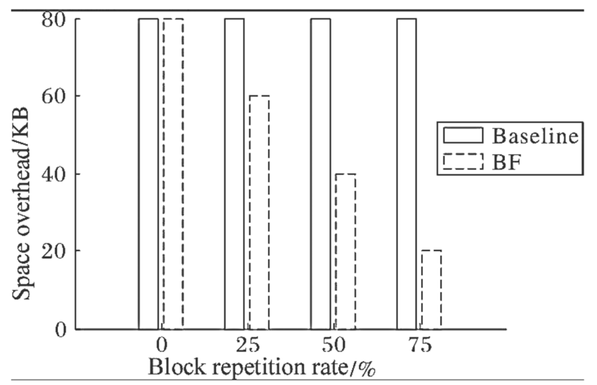

Fig. 9 Space overhead of cryptographic convergence secret key. ▭: It represents the space cost of baseline scheme encryption and convergence key. ⬚: It represents the space cost of BF scheme encryption and convergence key

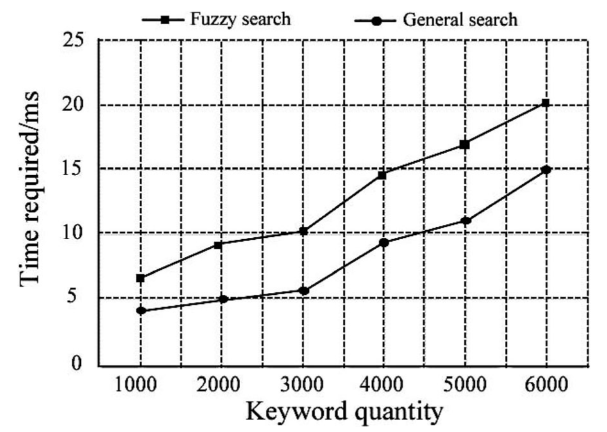

Fig. 10 Construction of index. ━■━: It represents the time overhead of index establishment when fuzzy search is performed. ━●━: It represents the time overhead of index establishment when general search is performed

7.2 Data integrity

After the medical data is downloaded to the local, the message verification code is computed, comparing with the data downloaded Convergence key, considering that the convergence key has also been tampered with, the attacker will have to have the data owner's private key, because the possibility is very low, so the use of the integrity of the message Verification code verification method is more reliable.

8 Conclusions

In spite of the above problems, there is still a lot of value in our scheme. In the rapid development of the medical industry information age, the growing mass of medical data to urge the birth and application of medical cloud. The protection of the privacy of medical data has also become a major focus of the medical cloud. In the two schemes of this paper, we introduce Bloom filter and private cloud respectively to improve the efficiency and safety of this structure in medical cloud. Then, according to the specific data type of medical cloud and the actual situation of the owner and user of the document, the method of identity token is used to realize the multiple role uploading and keyword accurate or fuzzy search safely, so that the structure application in medical cloud is more operable and closer to the actual situation.

At the same time, the paper proposes the structure combination of document ownership and authentication to avoid the illegal users of confidential data caused by the leak. Finally, using the message verification code to test the integrity of the downloaded documents, the users can avoid using the wrong medical data blindly and have unpredictable and serious consequences.

Acknowledgements
This work was supported by the Natural Science Foundation of Shandong Province under Grant No.ZR2017MF029.

Authors' contributions
HZ and LW contributed to the conception and algorithm design of the study. YW and MS contributed to the acquisition of simulation. HZ and JL contributed to the analysis of simulation data and approved the final manuscript. All authors read and approved the final manuscript.

Authors' information
Huiqi Zhao received his B. Sc and M. Sc degree from Shandong University of science and technology, China, in 2003 and 2009, respectively. He is currently working toward his Ph. D. degree in Shandong University of science and technology. He is an Ph. D. and lecturer in Department of information engineering, Shandong University of science and technology, Taian, Shandong Province and Qilu University of Technology (Shandong Academy of Sciences), Shandong Computer Science Center (National Supercomputer Center in Jinan), Shandong Provincial Key Laboratory of Computer Networks, Shandong Jinan. His research interests are computer networks, wireless body area networks and healthcare applications, and data privacy protection.
Lexia Wang is a college student in Department of information engineering, Shandong University of science and technology, Taian, Shandong Province. Her research interests are network and information security.
Yinglong Wang is a doctor and researcher in Qilu University of Technology (Shandong Academy of Sciences), Shandong Computer Science Center (National Supercomputer Center in Jinan), Shandong Provincial Key Laboratory of Computer Networks, Jinan, Shandong Province. His research interests are information security, computer forensics and wireless sensor network technology.
Minglei Shu is a doctor and associate researcher in Qilu University of Technology (Shandong Academy of Sciences), Shandong Computer Science Center (National Supercomputer Center in Jinan), Shandong Provincial Key Laboratory of Computer Networks, Jinan, Shandong Province. His research interests are wireless sensor network, wireless body domain network and cloud health.
Jimin Liu received his Ph.D. degree in Shandong University of science and technology. He is an Ph. D. and lecturer in Department of information engineering, Shandong University of Science and Technology, Taian.
His research interests are computer networks and cloud computing

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1]Department of Information Engineering, Shandong University of Science and Technology, Taian, China. [2]Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, Shandong, China.

Received: 2 May 2018 Accepted: 26 June 2018
Published online: 27 July 2018

References
1. Liu Xueyan, *Research on data access control method based on attribute encryption* (Lanzhou University of Technology, 2016). http://kns.cnki.net/kns/detail/detail.aspx?FileName=1017802099.nh&DbName=CDFD2017
2. W Jianfeng, *Research on fuzzy searchable encryption scheme in cloud computing* (Xi'an University of electronic science and technology, 2013), http://kns.cnki.net/kns/detail/detail.aspx?FileName=1013304858.nh&DbName=CMFD2013
3. Z Jiashun, Z Yongxie, G Yan, Research on ABAC model based on usage control in cloud environment. Comput Appl Res **31**(12), 3692–3694+3699 (2014)
4. S Jin-shu, Il Apao, Wang, Yipin S, H Qiaolin, Attribute base encryption mechanism. Software J. **6**, (2011)
5. F Anmin, S Jianye, S Ying, L Yu, The client-side security scheme for ciphertext data in cloud storage. Electronics **45**(12), 2863–2872 (2017)

6. W Chenfeng, *Research on data protection of medical cloud system based on attribute encryption* (Xi'an University of electronic science and technology, 2016) http://kns.cnki.net/kns/detail/detail.aspx?FileName=1016214547.nh&DbName=CMFD2017
7. H Nana, *Research on data sharing scheme based on privacy protection in medical cloud* (Xi'an University of electronic science and technology, 2015) http://kns.cnki.net/kns/detail/detail.aspx?FileName=1016248002.nh&DbName=CMFD2017
8. W Jianfeng, *Research on efficient retrieval and security audit technology of outsourced data in cloud environment* (Xi'an University of electronic science and technology, 2016) http://kns.cnki.net/kns/detail/detail.aspx?FileName=1016245829.nh&DbName=CDFD2017
9. Z Yukun, F Dan, X Wen, F Min, A convergent encryption strategy based on two-time hashing for data-weight. Comput Eng Sci **38**(09), 1755–1762 (2016)
10. J Li. Research on key issues of data outsourcing security in cloud computing. Nankai University, 2014
11. HG Wang, KF Chen, BD Qin, et al., Randomized convergent encryption in standard model via UCES//Con Ference on Computer and applications. IEEE, 298–302 (2017)
12. S Zhirong, X Wei, S Jiwu, Research and development of searchable encryption mechanism. Software J **25**(04), 880–895 (2014)
13. Y Xiaolong, *Research on fuzzy keyword searchable encryption scheme in cloud computing* (Chongqing University, 2016) http://kns.cnki.net/kns/detail/detail.aspx?FileName=1016908011.nh&DbName=CMFD2017
14. Yin Qinqin. Hybrid cloud storage based on Bloom filter security to go heavy scheme. Computer Engineering and Application: 1–9[2018–02-20]. http://kns.cnki.net/kcms/detail/11.2127.Tp.20170810.0852.008.html. Accessed Aug 2017
15. M Yang, Research on the sharing of medical archives information in the cloud ERA. China Manag Inf **19**(03), 204 (2016)
16. L Shi, C Xiaofeng, L Yupeng, Cloud platform history electronic medical record case similarity fuzzy inference research. Electron Technol Softw Eng **10**, 29 (2015)
17. H Xiao, The rights and obligations of hospitals and patients with respect to medical records. Chinese Med Inf lead **19**, 15 (2002)
18. Z Jing, Z Yuanyan, T Lunyu, Study on the security mechanism of medical cloud platform. Comput CD-ROM Software Appl **18**(01), 17+19 (2015)
19. J Li, X Chen, F Xhafa, L Barolli, Secure deduplication storage systems supporting keyword SEARCH. J Comput Syst Sci. **81**(8), 1532–1541 (2015)
20. S HALEVI, D HARNIK, B PINKAS, et al., *Proofs of ownership in remote storage systems//Proceedings of the 18th ACM Conference on Computer and Communications Security* (ACM, New York, 2011), pp. 491–500
21. L González-Manzano, A Orfila, An efficient confidentiality-preserving proof of ownership for deduplication. J Netw Comput Appl **50**, 49–59 (2015). https://doi.org/10.1016/j.jnca.2014.12.004
22. J Xiong, Y Zhang, F Li, S Li, J Ren, Z Yao, Data security in cloud environment to restudy progress. J Commun. **37**(11), 169–180 (2016)
23. J Wang, Z Zhao, Z Xu, H Zhang, L Li, Y Guo, I-sieve: An inline high performance deduplication system used in cloud storage. Tsinghua Sci Technol. **20**(01), 17–27 (2015)
24. R Zhu, L Qin, J Zhou, H Zheng, Using multi-threads to hide deduplication I/O latency with low synchronization overhead. J Cent South Univ. **20**(06), 1582–1591 (2013)
25. L Zhenhua, K Yaqian, L Chen, F Yaqing, Hybrid cloud approach for block-level deduplication and searchable encryption in large universe. J China Univ Posts Telecommun **24**(05), 23–34 (2017)
26. Zhibo Li. Deduplication of files in cloud storage based on differential Bloom filter. The Institute of Electrical and Electronics Engineers、IEEE Beijing Section. Proceedings of 2016 IEEE 7th International Conference on Software Engineering and Service Science (ICSESS 2016). The Institute of Electrical and Electronics Engineers, IEEE Beijing Section: 2016:4. http://kns.cnki.net/kns/detail/detail.aspx?FileName=IEEE201608003026&DbName=IPFD2017
27. M Jianting, *A Deduplication-based Data Archiving System. IACSIT*, Proceedings of 2012 international conference on image, Vision and Computing(ICIVC 2012).IACSIT (2012), p. 5
28. L Sandrine, Multiple protective pathways against reperfusion injury: a SAFE path without Aktion? J Mol Cell Cardiol. **46**(5), 607–609 (2009)
29. MC Talley Kristine, F Wyman Jean, R Gross Cynthia, Psychometric properties of the activities-specific balance confidence scale and the survey of activities and fear of falling in older women. Am Geriatr Soc J **56**(2), 328–333 (2008)
30. T Debbie, SAFE discharge for infants with high-risk home environments. Adv Neonatal Care **7**(4), 167 (2007)
31. C Bridget, Making Wisconsin kids SAFE KIDS. Wis Med J. **104**(1), 15 (2005)