




Enabling Efficient Multi-keyword Search Over Fine-Grained Authorized Healthcare Blockchain System

Yicheng Ding, Wei Song^(✉) , and Yuan Shen

School of Computer Science, Wuhan University, Wuhan, Hubei, China
{2019282110239, songwei, yuanshen}@whu.edu.cn

Abstract. As a new emerging technology, blockchain is attracting the attention from academic and industry and has been widely exploited to build the large-scale data sharing and management systems, such as healthcare database or bank distributed database system. The health records contain a lot of sensitive information, so putting these health records into blockchain can solve the security and privacy issues while uploading them to an untrustworthy network. In a typical health record management system, there are escalating demands for users including the patients and the doctors to execute multi-keyword search over the huge scale of healthcare records. In the meantime, they can authorize some part of their personal treatments to others according to personalized needs of the patients. In literatures, there is not an existing blockchain solution can satisfy these two requirements at the same time. These issues become prominent since it's more inconvenient to adjust a blockchain-based system to support efficient multi-keyword search and fine-grained authorization comparing to traditional RDBMS. To overcome the two challenges, we propose a novel multi-keyword searching scheme by establishing a set of Bloom Filters within the health record blockchain system to accelerate the searching process on service provider (SP). Moreover, we reduce the overhead of key derivation by proposing a Healthcare Data Key Derivation Tree (HDKDT) stored locally on the user's side. Putting our proposed scheme on the medical blockchain can speed up the multi-keyword search [3, 12] processes and reduce the key storage space to certain extent. At the end of this article, we formally prove the security of the proposed scheme and implement a prototype system to evaluate its performance. The experimental results validate our proposed scheme in this paper is a secure and efficient approach for the health record management scenario.

Keywords: Multi-keyword search · Fine-grained authorization · Blockchain in healthcare

1 Introduction

It has been a decade since cryptocurrency such as Bitcoin and Ethereum has a great impact on trading and data sharing [11]. The underlying technology behind these networks is

called blockchain which is a distributed ledger. In blockchain, recording transactions and tracking assets don't rely on any intermediaries by building a chain of block. Combining with the consensus protocols such as proof of work (POW) [2] in Bitcoin and proof of stake in Ethereum, blockchain ensures all the loyal nodes within the peer-to-peer network to own the same copy of data. Blockchain is essentially an append-only data structure with ascending timestamp data that sorted by the timestamp and query data on top of it since it's immutable. Tampering data on blockchain needs to corrupt the entire system in a certain degree.

Based on the immutable nature of blockchain technology, it becomes the ultimate use to improve medical record management, accelerating clinical and biomedical research and advanced health record ledger in the field of healthcare. In the field of healthcare, the integrity and security of data such as the case history of patients are the first priority for medical institutions or hospitals. They don't want to share their own data with counterparts if the shared data can't guarantee to be authenticated. That's what blockchain does, it fills the gap of distrust among the patients and the medical institutions.

Thus, there have been a larger number of blockchain applications for the medical data management scenarios such as Doc.AI nowadays. For example, disease researchers are likely interested in a certain kind of diseases, which majority of them have some common features such as breast tumors usually have some symptoms like growing out a lump around the breast, so these researchers may search for {"tumor", "lump", "early"} on the blockchain to fetch the health records that are relevant to those keywords. On the other hand, sharing data among peers in the network requires a huge number of keys to maintain the confidentiality. Thus, we need an efficient multi-keyword algorithm and a fined-grained authorization to elegantly solve these problems.

However, there is not an existing blockchain solution can well support these two fundamental operations for the real-world healthcare data management system. To address this issue, we propose a novel multi-keyword searching scheme by establishing Bloom Filter [6] within the health record blockchain system. Moreover, we design a Healthcare Data Key Derivation Tree (HDKDT) to achieve the fine-grained authorization and reduce the overhead of key derivation. The main contributions of this paper can be summarized as follow:

- We propose a novel multi-keyword search scheme on the blockchain in healthcare system to optimize the performance of search request. It reduces the times of the interaction among the clients (patients) and servers (services provider) by completing search request from the clients within one send/receive and speeding up search process with a probabilistic called Bloom filter.
- We propose a key derivation management scheme called HDKDT to enable fine-grained authorization among patients and research institutions with reducing key storage space and shortening the key derivation path.
- We theoretically analyze the security of the proposed blockchain scheme, moreover, we evaluate the performance against the state-of-the-art solutions. The experimental results show that our model is an efficient and secure solution for the healthcare records management.

The rest of the paper is organized as follow, in the next section, we discuss the related work. And Sect. 3 introduces the system model of the medical chain network and explains the motivation of optimizing multi-keyword search and enabling fine-grained authorization. Then, we introduce several preliminaries in Sect. 4. Section 5 details our work which is followed by the security analysis in Sect. 6. Experimental parameters and results are reported in Sect. 7. Finally, we conclude our paper and provide directions of future work in Sect. 8.

2 Related Work

The main motivation of our work is to design a practical blockchain scheme for the medical data management. In this section, we review the relevant work for this scenario, including the multi-keyword search over encrypted data on blockchain and fine-grained authorization with Attribute-Based Encryption (ABE) [16] scheme.

2.1 Multi-keyword Search Over Encrypted Data on Blockchain

Recent years, many efforts have been done to enable multi-keyword search on blockchain. Shan et al. [4] have developed a fine-grained searching scheme that can fetch all the matched results from the SP within one round communication. They put some auxiliary spaces into both the client and the SP to speed up multi-keyword search on the blockchain. Zhang et al. [13] proposed a novel authenticated data structure (ADS) [9], called GEM²-tree, which is not only gas-efficient but also effective in supporting authenticated queries. To further reduce the ADS maintenance cost without sacrificing much query performance, Hu et al. [14] introduced a scheme by designing a new smart contract for a financially-fair search construction, in which every participant (especially in the multiuser setting) is treated equally and incentivized to conform to correct computations. Ruan et al. [15] proposed an ADS called LineageChain provides a skip list index designed for supporting efficient provenance query processing. Niu et al. [17] proposed a multi-keyword search scheme under the assumption of decisional bilinear Diffie-Hellman exponent (q -BDHE) and decisional Diffie-Hellman (DDH) in the selective security model.

Cheng et al. [5] builds index inside each block called intra-block index and the other among all the blocks called inter-block index. The intra-block index skips some of the mismatched attributes rapidly by creating an ADS that unions by every data set within the block. The inter-block index skips some of the mismatched block by using SkipList.

These schemes we described above are not likely fix into blockchain in healthcare system to support multi-keyword search in a large scale since none of them can strictly reduce the time complexity with multi-keyword search. Thus, we propose a novel multi-keyword search scheme based on a set of Bloom filter [8] that can always guarantee to acquire relatively good performance on the worst case time scenario.

2.2 Authorization with Attribute-Based Encryption Scheme

Vasilios et al. [18] proposed a new type of OAuth 2.0 token backed by a distributed ledger. Their construction is secure, and it supports proof-of-possession, auditing, and accountability. Zhang et al. [19] discussed the potentials of the proposed approach to effectively

address certain vulnerabilities in current OAuth-like authorization and authentication services with tolerable performance. Rashid et al. [20] proposed a multi-layer security network model for IoT network based on blockchain technology for authentication and authorization purpose within each cluster handed by each Cluster Head. Yamauchi et al. [21] pointed out a new issue by abusing published information with relation to recipients on blockchain. Then, they proposed a solution to their discovered issue. Widick et al. [22] proposed a novel authentication and authorization framework based on blockchain technologies to control access to the resources of an IoT device.

However, all the schemes above are restricted by the key storage space for user to manage their own keys for authorization. It is not practical enough for the medical data sharing scenario. Thus, we extend GLHG [1] to enable a key derivation path in a graph for the data owner to calculate derived key while the data owner only needs to hold the source key at all time, which cost relatively small space to implement this scheme for the clients.

3 Problem Definition

3.1 System Model

Figure 1 shows the system model in this paper which is the primary structure of healthcare system. There are two entities in our system model, light node only stores the block header while full node stores a full copy of the entire medical chain.

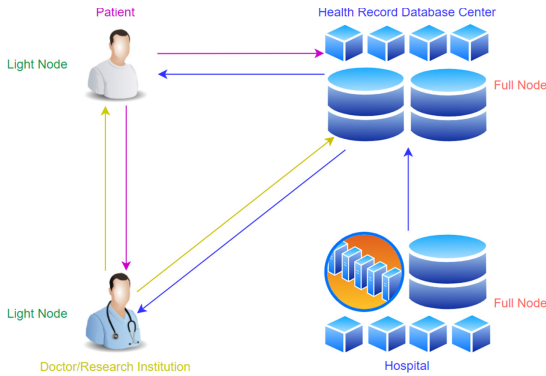


Fig. 1. A Blockchain Network in healthcare

More specifically, the clients, patients, Doctor or research institutions represented as light node, the service providers, healthcare database center and hospitals represented as full node. The light node can send their request to the SP such as health record database center and receive the corresponding result set. The light node such as the patients can also agree to authorize some of their health records by sending corresponding derived key to doctors or research institutions if they want to trade for their data. Once these doctors or institutions get the key, they are allowed to search and look up the health

records of these patients from the SP. On the other hand, Full node such as hospitals pack up a number of health records of different patients from the mining pool and create a new block after these patients commit to pay the gas after every loyal node in the network have verified and accepted this new block (reach consensus).

3.2 Design Goals

In the business related to the healthcare, the patients can be seen as the original owners of their own health records. It's somehow a certain kind of assets which they are eligible to acquire a full control over these pieces of data even if they don't realize about this. To let them be aware of this situation, companies like **Medicalchain** start to build their own blockchain for storing and securing their electronic health records. Legally, the patients can sell all or some of their data to other research institutions, medical big data companies and healthcare sectors by authorizing them to be one of their authorized users and get paid through the medical chain. And the authorized users such as the doctors can perform multi-keyword search on the chain to look up the some of the owner's health records. The design goals of our paper include the multi-keyword search and the fine-grained authorization over medical blockchain.

Definition 1: Multi-keyword Search Over Blockchain: Given a client C and a set of keywords $K = \{k_1, k_2 \dots k_n\}$, C asks the SP to retrieve the health records according to K . Based on the protocol, the SP executes the multi-keyword search for C over the blockchain and returns the result $R = \{r_1, r_2 \dots r_m\}$. For each $r_i \in R$, r_i contains all the keywords in K .

Suppose the clients sends three keywords $K = \{\text{"triple-negative breast cancer"}, \text{"positive"}, \text{"early"}\}$ to the SP. Once SP receive these three keywords, it will search on all the health records on the entire blockchain to see if there exist some health records that match to these three keywords. e.g. $\{\text{"Nasopharyngeal Carcinoma"}, \text{"positive"}, \text{"early"}, \text{"chemotherapy three months ago"}\}$ is not a match, but $\{\text{"triple-negative breast cancer"}, \text{"positive"}, \text{"early"}, \text{"radiation therapy 1 month ago"}\}$ is a match.

Definition 2: Fine-Grained Authorization Among Clients: Given a client Alice and other client Bob, Alice asks Bob to authorize her a set of a set of encrypted derived keys $DK = \{dk_1, dk_2 \dots dk_n\}$ based on the key exchange protocol, then Bob calls the key derivation function (KDF) [10] to generate the corresponding DK and send back to Alice. After that, Alice can send search request to SP to look up some of the Bob's health records.

Usually, Multi-keyword search aka data provenance or lineage takes linear time to achieve since demanding search operation and constructing ADS directly on blockchain are limited by the native hash pointer that forms the chain. More importantly, each health record may be encrypted by a group of symmetric key using PRF for different attributes (e.g. treatment program, condition description or body index) within it since medical chain needs to enable fine-grained authorization for the client to trade some part of their health record with other peers in the network. It increases the difficulty level of applying efficient multi-keyword search and arranging cumulative key storage space. Hence, we add a set of bloom filters [6–8] to the blockchain to support multi-keyword search and avoid mismatched attributes by using owner's key as the index to shorten the search path

while we're doing binary search on the top of the bloom filter. Combing with this scheme, we propose encryption scheme the Healthcare Data Key Derivation Tree (HDKDT) to dynamically derive and create new encryption key from owner's original key to satisfy the requirement of fine-grained authorization for encrypting different contents.

4 Preliminaries

4.1 Bloom Filter

It's a space-efficient probabilistic data structure conceived by Burton Howard Bloom in 1970 that determine if an element is in a set and save a large amount of time and space comparing to hash table when the server needs to deal with massive data. The mechanism behinds the bloom filter is suppose we have an array of hash function [7] $\{h_0, h_1, h_2, \dots, h_{k-1}\}$ where k is the number of hash function, then we can apply each h_i to an element to get an array of index $IS_1 = \{i_{00}, i_{10}, i_{20}, \dots, i_{(k-1)0}\}$ and another element to get $IS_2 = \{i_{01}, i_{11}, i_{21}, \dots, i_{(k-1)1}\}$. The best part of these two arrays is they are unlikely to be the same. Also, it's relatively hard to find a collusion array IS_3 that equals to IS_1 or IS_2 or any other IS_i that forms by arbitrary string. In this case, the server has three variables, the size of bloom filter array m , the number of hash function k , the expected number of data n and defining error rate as er , then we have the formula below for the server to choose the most appropriate size of bloom filter and the number of hash function:

$$m = \frac{n \ln(er)}{(\ln 2)^2} \quad k = \frac{m}{n} \ln 2$$

Once the server picks up the number m and k that is close or satisfy the formula, it gets better performance on solving this problem.

4.2 Access Control Based on Global Logical Hierarchical Graph (GLHG)

To enable fine-grain authorization on the medical chain, the client has to generate its own key pair and regenerate derived keys based on the original key pair. The process of the data owners authorizing other users to access the data is essentially an authorized sharing of some of their own data key. In the meantime, to allow the data owners are able to perform selective data authorization access to different users, the data owners need to encrypt different parts of their data with different keys. As the time goes on, the number of keys will grow rapidly with the escalating amount of data, and users will suffer from storing a large scale of redundant keys in their own disk or memory.

Thus, a secure, efficient and flexible support for accessing authorization control key management methods are critical for sharing data. According to Peng et al. [1], users can apply a fine-grained key management scheme based on Global Logical Hierarchical Graph (GLHG) as $G[R, V, E, T]$, which greatly release the redundant space for storing derived keys. Here is some of the definition about GLHG:

Definition 3: The GLHG key derivation graph $G[R, V, E, T]$ satisfies the full coverage of the key derivation path iff the following condition hold: for $\{\forall v | v \in V, v.Level > 2\}$, such that $\cup\{v_i.acl, v_i \in V, (v_i, v) \in E\} = v.acl$.

Definition 4: The GLHG key derivation graph $G[R, V, E, T]$ satisfies the key derivation path that is not redundant iff the following conditions are true: for $\{\forall v_l | v_l \in V, v.l\text{level} > 2\}$ and $\{\forall v_l | v_l \in V, (v_l, v) \in E\}$, $\exists u \in v_l.acl$, such that $u \notin \{\forall v_l.acl | v_l \in V, (v_l, v) \in E, v_l \neq v_l\}$.

5 Efficient Multi-keyword Search Algorithm Over Blockchain

5.1 Constructions of the Bloom Filter Index

To obtain multi-keyword search functionality on medical chain whether it's encrypted or not, we essentially need to avoid mismatched case as best we can. Hence, the hospital can establish a set of bloom filter denoted as $BF_i = \{bf_{i0}, bf_{i1}, bf_{i2} \dots bf_{i(n-1)}\}$ during the creation of a block, where i indicates the index of the block and 2^n gives an upper bound on the size of the blockchain, then the hospital uses multiple sets of hash functions $\{\{h_{00}, h_{01} \dots h_{0(k_0)}\}, \{h_{10}, h_{11} \dots h_{1(k_1)}\}, \{h_{20}, h_{21} \dots h_{2(k_2)}\} \dots \{h_{(n-1)0}, h_{(n-1)1} \dots h_{(n-1) \cdot (k(n-1))}\}$ to build Bloom filter, where both $[h_{j0}, h_{j1} \dots h_{j(k_j)}]$ and the number of hash function denoted as $k_j + 1$ belong to bf_{ij} ($0 \leq j \leq n$).

Assuming the hospital wants to pack and verify a set of health records $\{hr_0, hr_1, hr_2 \dots hr_{m-1}\}$ into a new block, where hr_j contains a set of attribute values $\{a_0, a_1, a_2 \dots a_{p-1}\}$. Some of these attributes may be encrypted by its owner's key and some of them may not. And the client put one secret number $shift_{ii}$ into each attribute value a_{ii} of each health record before the client adds to the mining pool where $0 \leq ii \leq p$, then hospitals use $[h_{j0}, h_{j1} \dots h_{j(k_j)}]$ to calculate a set of index position $ps_j = \{l_{j0}, l_{j1}, l_{j2} \dots l_{j(k_j)}\}$ for a_0 by setting $l_{j(jj)} = h_{j(jj)}(a_0) + shift_0$ ($0 \leq jj \leq k_j$) and put them into bf_{ij} , then hospitals calculate ps_{j+1} for $bf_{i(j+1)}$ using $[h_{(j+1)0}, h_{(j+1)1} \dots h_{(j+1)k(j+1)}]$ and so on. The equation below shows this process in detail:

$$\begin{aligned}
 \bullet ps_0 &= \{l_{00} = h_{00}(a_0) + shift_0, l_{01} = h_{01}(a_0) + shift_0 \dots l_{0(k_0)} = h_{0(k_0)}(a_0) + shift_0\} \\
 \bullet ps_1 &= \{l_{10} = h_{10}(a_0) + shift_0, l_{11} = h_{11}(a_0) + shift_0 \dots l_{1(k_1)} = h_{1(k_1)}(a_0) + shift_0\} \\
 &\dots \dots \dots \dots \dots \dots \\
 \bullet ps_i &= \{l_{i0} = h_{i0}(a_0) + shift_0, l_{i1} = h_{i1}(a_0) + shift_0 \dots l_{i(k_i)} = h_{i(k_i)}(a) + shift_0\} \\
 &\dots \dots \dots \dots \dots \dots \\
 \bullet ps_{n-1} &= \{l_{n0} = h_{n0}(a) + shift_0, l_{n1} = h_{n1}(a) + shift_0 \dots l_{i(k(n-1))} = h_{i(k(n-1))}(a) + shift_0\}
 \end{aligned}$$

Intuitively, hospitals continue to calculate all the index positions for $a_1, a_2 \dots a_{p-1}$. The secret number $shift$ that belongs to each attribute value of each health record is based on the identifier id that equals to the hash h_{sk} of the symmetric key (SK) of its owner if the attribute is encrypted by that SK. SK is created by key derivation scheme that we will discuss later. We define $shift = h_{sk}(SK) + flag$, where the number $flag$ reveals which direction of each binary search case will go to (left, right or both) when SP starts this binary search on the top of the bloom filter to find matches. If the hospitals set $flag = +1$ during the creation step of a new block, then SP will go to block $2^{n-1}+1$ to block 2^n to find the matched attributes of the given keyword that are encrypted by the SK of the owner and set $flag = -1$ means go to block₀ to block 2^{n-1} . If an attribute within a health record is not encrypted, then we simply set $h_{sk}(SK)$ to be 0 since there is no encryption key. And the miner set $flag = 0$ to imply it's the end of the recursion call.

5.2 Search Phase

Before enabling efficient multi-keyword search on the bloom filter, the client needs to send a map denoted as $KI = \{(key_0, id_0), (key_1, id_1) \dots (key_{s-1}, id_{s-1})\}$ that contains multiple keyword-identifier pair (key_r, id_r) where $0 \leq r \leq s$. Then the client uses the same SK_r to compute id_r ($id_r = h_{sk}(SK_r)$) and some of them may not. At the first step, SP can start a binary search on the bloom filter bf_{n-1} to determine if there exist some attributes from block₀ to block 2^{n-1} or block $2^{n-1}+1$ to block 2^n or both that match to key_r while SP know the corresponding value of id_r . Once every decision of the next direction for every key_r is made, SP simply takes an intersection of them and apply recursions on the final decision set ds . (e.g. $ds = \{left\}$ if the global decision set $gs = \{[left, right], [left], [left]\}$ or $gs = \{\emptyset\}$ if $ds = \{[left, right], [left], [right]\}$). To sustain this optimization with binary search, we need $n \log(n)$ bloom filters to redirect instead of n . Because each block only store part of the bloom filter array that contains its attributes (e.g. in Fig. 2). And merging bf_{100} and bf_{101} forms a greater bloom filter with size of 26 denoted as bf_{10} . With this data structure below, SP can locate all the health records where one or more attributes in these records matches to each key_r . At last, SP will find all the exact health records that matches the KI and takes a union of all matched result sets found from different block that contains multiple health records and sends it back to the client.

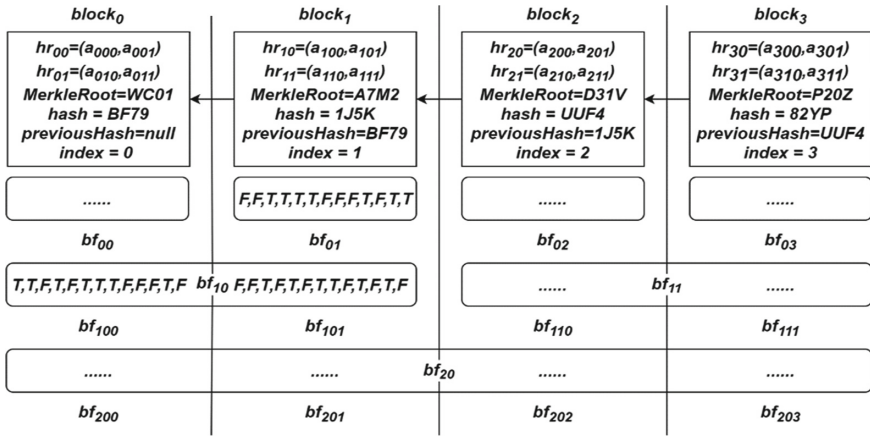


Fig. 2. Bloom Filter index structure on the Medical Chain

Notice that in Fig. 2, for attribute a_{110} , these three subscript numbers indicate the index of the block in the medical chain, the index of health record in the block and the index of the attribute in health record. And for bf_{101} , these numbers indicate the level (starting from 0) or index idx_1 of the bloom filter set BF_1 in the global bloom filter set BF , the index idx_2 of the bloom filter BF_2 in the set BF_1 and the index of the subarray (a part of BF_2) in BF_2 .

In Fig. 2, we have: (h_{ij0}, h_{ij1}) belongs to bf_{ij} and the length of each chunk of bf_{ij} equals 13. Specifically, we calculate the index position for each attribute by using $(h_{ijk}(a_{xyz}))$

+ $h_{sk}(SK_{xyz}) + \text{direction}) \% \text{len}(n * 13)$, then the true or false elements in bf_{01} is determined by the union of all the index positions ip we have calculated for $\{a_{100}, a_{101}, a_{110}, a_{111}\}$. If we assume $ip = \{\{3,11\}, \{2,5\}, \{9,2\}, \{4,12\}\}$, then the elements in bf_{01} is shown above. In this case, the direction number is 0.

Other cases such as bf_{10} follows the same step as we state above, remembering that the index position hospitals calculate for each attribute may reside in different chunks of bf , thus the client needs to wait for the creation of new blocks until this chunk is created together with the corresponding new block. If hospitals calculate all the index positions of a_{000} in bf_{10} is $\{0, 15\}$ which means the first element in bf_{100} and the third element in bf_{101} will set to be true. However, if $block_1$ has not created yet, the hospitals only set the first element bf_{100} to be true and bf_{10} is not available now since hospitals only have $block_0$ and only have the left half of bf_{10} which is bf_{100} . If the hospitals get the index positions for $\{a_{000}, a_{001}, a_{010}, a_{011}, a_{100}, a_{101}, a_{110}, a_{111}\}$ is $\{\{0,15\}, \{5,17\}, \{6,20\}, \{1,11\}, \{24,7\}, \{20,3\}, \{3,6\}, \{19,22\}\}$, then the element in bf_{10} is shown above.

Other factor that can have influence on the performance of bloom filter is the choice of hash function. Fast simple non-cryptographic hashes with collision resistant fashion which are independent enough for bloom filter to use can be FNV series such as FNV-1 and FNV-1a. BKDR hash (hash function of Hashmap in Java) also works fine, but it takes longer time to calculate the index since it combines the output index of each character of a string by simply adding them with Horner's rule while FNV does bitwise XOR. We can also set the size of the bloom filter to be power of 2 to accelerate the mod operation by using bitwise AND between the index and size - 1. We also put different prime number to distinguish these hash functions in each bloom filter while doing $hash = hash * prime \wedge \text{byte_of_data}$.

In real case, suppose Alice wants to see Bob's health records, then before this transaction proceed and executed by smart contract, it needs to check Alice's balance to see if she has sufficient money to spend. The only way to get balance for the smart contract is to look up the entire chain and sum up the balance. The only way to get balance for the smart contract is to look up the entire chain and sum up the balance. This procedure can be done efficiently by the scheme about it's transparent to the smart contract and accelerate the searching.

5.3 Key Derivation and Fine-Grained Authorization

In Sect. 4.2, we introduce a key derivation scheme called GLHG. It allows different data owned by different users can be encrypted with the same key under the premise of they share some common derived keys which means they share the same group of authorized users. For data owners, they don't need allocate a linear key storage space to keep track which key belongs to whom by creating key and authorized user entries and put them into an unsorted map. Because they can derive all the keys if there exists a derivation path from the key they're holding to these keys in the GLHG.

We believe that medical data is a personal asset and should be authorized by the patient himself. However, due to the lack of professional knowledge of individuals, complete autonomous authorization will make it difficult to control the scope of data sharing and lead to the inevitable privacy leaks. Therefore, we design a HDKDT to

achieve practical and fine-grained authorization for the medical data management scenarios. Generally, it extends GLHG by restricting the authorization of the patients with building an authorization tree of the patients (data owners) shown as following.

The structure of HDKDT is illustrated in Fig. 3, which contains two parts, i.e., the disease classification tree, and the level hierarchical structure. If the patient c upload two health records denoted as A and B to the HDKDT, and then authorize two doctors denoted as Doctor A and Doctor B with the corresponding keys. The doctor B at chief physician level can decrypt both records in terms of holding the corresponding symmetric key. For doctor B, it's impossible for him to calculate any derivation path to get the key that can decrypt the health record A.

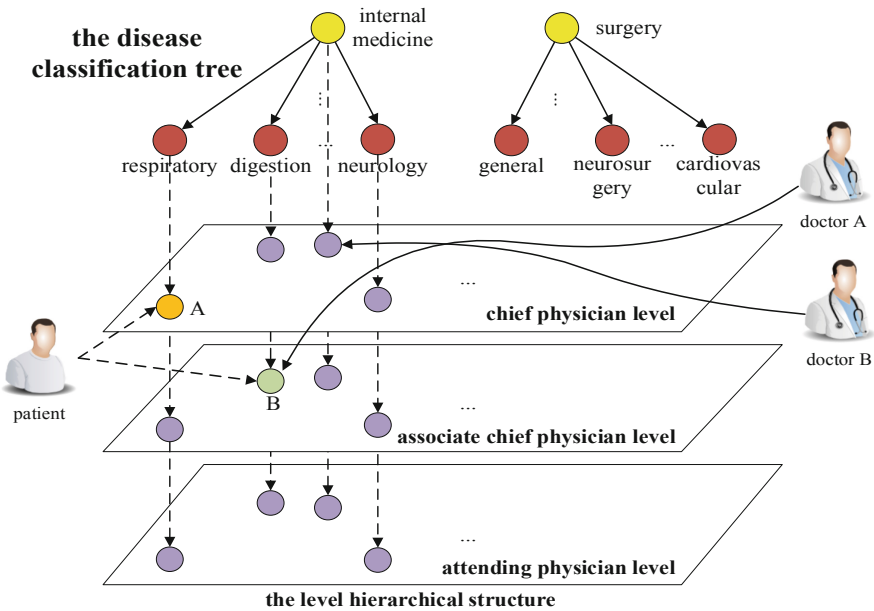


Fig. 3. Fine-grained authorization based on HDKDT

Instead of enabling the patients to randomly authorize their derived key to any other untrustworthy parties, we only allow the patients to authorize the derived key to other trusted third party such as a chief physician of Oncology in some surgical department. Hence, we can cut off lots of unnecessary authorization between the data owners and some anonymous users by managing the limit of authorization. Moreover, by setting up these restrictions, it turns out to be a tree data structure instead of a graph, which means it will be less redundant pointers. Our method can adapt to the fine-grained authorization scenario of medical data since it perfectly fit the classification of any hospital departments.

6 Performance and Security Analysis

6.1 Performance Analysis

By enabling a binary search on the top of blockchain with Bloom Filter, we can speed up the search process with $O(\log(N))$ where $N = 2^n$ is the number of blocks in the chain at worst case scenario. Theoretically, its performance gets better than those ADS in [4] and [5], since those ADS are not strictly $O(\log(N))$. Although we need some auxiliary spaces to store the number of $\log(n)$ Bloom filter with each block, which increases the space complexity to $O(N\log(N))$, it's still worthy to get $\log(N)$ on search process. On the other hand, with extending HDKDT instead of using traditional key derivation function such as PBKDF2 to establish the key management system, the data owners can reduce the key storage space by simply storing the source key and using it to derive all the authorized keys on the derivation paths within HDKDT.

6.2 Security Analysis

We can implement the global user access authorization policy $A[U][D]$ [1] based on the HDKDT key derivation mechanism $G[R, V, E, T]$. Its security and unforgeability are achieved by the following guarantees:

1. The security of the key derivation function. The unidirectionality and security of the derivation process have been analyzed in detail in [4, 5].
2. Uniqueness of user's key. For $\forall v_i.key, v_j.key \in UKey$, such that $v_i.key \neq v_j.key$.
3. The correctness and completeness of the key derivation path. 1) **correctness**: for $\forall u_i \in U$, if $\exists v \in V, \exists (v, v_j) \in E$ and $u_i \in v.acl$, such that $u_i \in v_j$, that is $\{v.key, v_j.key\} \in \varphi^*(u_i)$; 2) **completeness**: see Definition 1.
4. The equivalence between $A[U][D]$ and $G[R, V, E, T]$. 1) **correctness**: for $\forall v \in V$, if $\exists u_i \in U, \exists d_j \in D$, such that $\varphi(d_j) = v.key \in \varphi^*(u_i)$, then we have $d_j \in v.data$ and $u_i \in v.acl = acl(d_j)$, which is $A[u_i][d_j] = 1$. 2) **completeness**: for $\forall u_i \in U, \forall d_i \in D$, if $A[u_i][d_j] = 1$, then we have $u_i \in acl(d_j)$ and $\exists v \in V$, such that $d_j \in v.data$ and $u_i \in v.acl = acl(d_j)$, which is $\varphi(d_j) = v.key \in \varphi^*(u_i)$.

The properties **1.** and **2.** ensures that any user can derive any other key pairs that can be derived iff these users has the original key pair. And the properties **3.** and **4.** ensures that any user can only obtain the key of the data that he has access to through the HDKDT key derivation map.

7 Performance Evaluation

We conduct our experiments by creating two different data sets and put them into a blockchain-based system to restore the real scenario of medical chain, then simulating the operations of creating new block with Bloom filter, starting multi-keyword search, building HDKDT and authorizing derived key between the virtual clients by on that. We run both data sets on one laptop running Ubuntu 18.04.4 with 64 GB DDR4 2933 MHz

XMP RAM and one Intel i9-9900K core. The SP forms a local simulated medical chain network and receive the query request from the virtual clients.

To aim on the real performance of our schemes, we set the difficulty of mining to less than 5 to accelerate our experiments since the mechanism of the complex network topology such as consensus protocol and malicious node are not taken into account. We also set network latency to be a constant number from 100–1000 ms. After setting up our experimental environment, we perform a multi-keyword search using a number of keywords about 236K from the *The Lancet* journal. Most of the keywords are related to treatment analysis, medical terms and disease description made up by key and value pairs such as {"name": "Alice", "Pathological grade":2, "Local lymph node size": "2 cm", "Liver ultrasound": "metastasis" ... }.

7.1 Performance of Multi-keyword Search

We stimulate a virtual environment while the hospital node creates new blocks and the patient node performs multi-keyword search. We encrypt these keywords with the associated keys that hold by different owners. By this means, we get a medical chain network with about 50–120 nodes, 10 health records in each book, 5 to 10 attributes in each health record and a block size of 100 to 3000 and 1000–10000 nodes (short chain), 40–80 health records in each block, 20–30 attributes in each health record and a block size of 5000–15000 (long chain). Notice that there may exist false positive error during the search step of SP. These errors may lead to some mismatched case that should be cutting out before, but it will never miss any matched attribute since there is not true negative error in bloom filter. We take an average search overhead per second and draw the following graph that reveals the performance by applying different schemes such as traditional method with repeating single-keyword search multiple time. Figure 4 shows our experimental results.

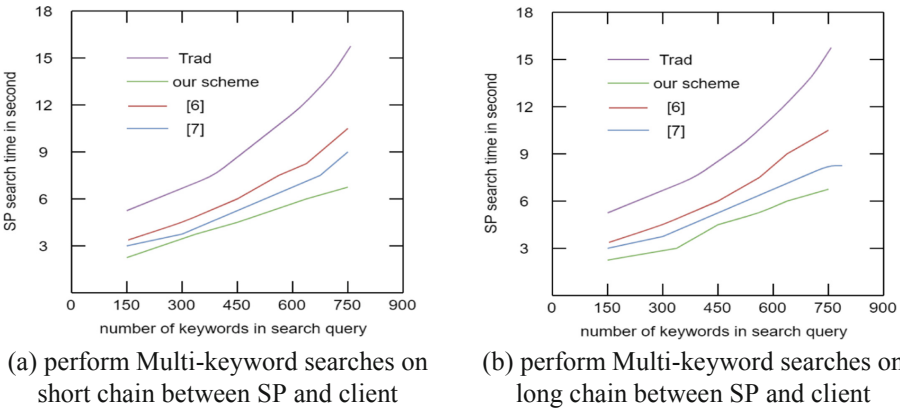


Fig. 4. performance test on multi-keyword search scheme

The running time overhead of traditional method is significantly affected by the number of keywords even if we have not count for network latency and packet loss may

occurred in the real network. And sending single-word search request multiple time increases the chance of network failure. Overall, the computational cost of our scheme speed up about 32% when the number of keys increases from 2–20 since we use bloom filter to locate the results and only need one send/receive for each search request.

7.2 Performance of Authorization

For enabling fine-grained authorization, we use PBKDF2 as the traditional method comparing to HDKDT. The virtual clients arbitrarily send 5–50 of their derived keys to other clients and generate their own health records with $h_{sk}(SK)$ and put them into the mining pool. We take the average memory cost and draw two curves based on them to the graph in Fig. 5.

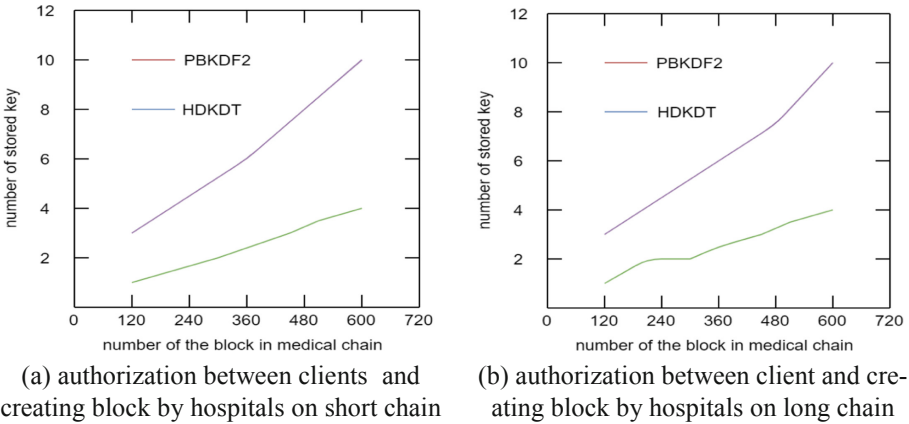


Fig. 5. performance test of HDKDT

From the graph about, we see as the number of authorized user increased, the volume of stored key is about 35% by implementing HDKDT comparing to PBKDF2. Because with HDKDT, we can avoid to store majority of the derived keys in local disk since we are able to calculate the specific key by following the corresponding derivation path with HDKDT.

8 Conclusion

In this paper, we propose both the multi-keyword search scheme and the fine-grained authorization scheme on the healthcare system. The technique to combine bloom filter with original blockchain-based system enables a fast search operation on the medical chain. The critical contribution lies in using hash function to redirect the direction of next recursion case for binary search, which greatly increases the possibility of skipping mismatched health records even if there exists some false positive case during the searching step.

On the client side, the memory cost of building key management system reduces dramatically by eliminating redundant derived key space using the HDKDT comparing to traditional method. Adopting the HDKDT to manage all the keys, and on the premise of ensuring the security and unforgeability of the derived key, we minimize the overhead of calculation, transmission during the authorization step.

For later optimization, we attempt to build an ADS within each block to further reduce the computation overhead of search and key storage space within some authentication data structure such as Merkle Hash Tree.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No. 62072349, U1811263, 61572378), the major technical innovation project of Hubei Province (No. 2019AAA072), the Science and Technology Project of State Grid Corporation Of China (No. 5700-202072180A-0-0-00), the Teaching Research Project Of Wuhan University (No. 2018JG052), and the Natural Science Foundation Of Hubei Province (No. 2017CBF420).

References

1. Fangquan, C., Zhiyong, P., Wei, S., Shulin, W., Yihui, C: Key management for access control in trusted cloud storages. *J. Comput. Res. Dev.* **50**(8), 1613–1627 (2013)
2. Porat, A., Pratap, A., Shah, P., Adkar, V: Blockchain Consensus: An analysis of Proof-of-Work and its Applications
3. Song, W., et al.: A privacy-preserved full-text retrieval algorithm over encrypted data for cloud storage applications. *J. Parallel Distrib. Comput.* **99**, 14–27 (2017)
4. Jiang, S., et al.: Privacy-preserving and efficient multi-keyword search over encrypted data on blockchain. In: Blockchain conference, pp. 405–410 (2019)
5. Xu, C., Zhang, C., Xu, J: vChain: enabling verifiable boolean range queries over blockchain databases. In: Proceedings of International Conference on Management of Data, Proceedings of SIGMOD Conference, pp. 141–158 (2019)
6. Bloom, B.: Space/time tradeoffs in in hash coding with allowable errors. *Commun. ACM* **13**(7), 422–426 (1970)
7. Carter, J.L., Wegman, M.N.: Universal classes of hash functions. *J. Comput. Syst. Sci.* **18**, 143–154 (1979)
8. Ramakrishna, M.V.: Practical performance of Bloom filters and parallel free-text searching. *Commun. ACM* **32**(10), 1237–1239 (1989)
9. Tamassia, R.: Authenticated data structures. In: Di Battista, G., Zwick, U. (eds.) *ESA 2003*. LNCS, vol. 2832, pp. 2–5. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39658-1_2
10. Papamanthou, C., Tamassia, R., Triandopoulos, N.: Optimal verification of operations on dynamic sets. In: Rogaway, P. (ed.) *Advances in Cryptology – CRYPTO 2011*. CRYPTO 2011. Lecture Notes in Computer Science, vol. 6841, pp. 91–110. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_6
11. Clifton, C., Doan, A., Elmagarmid, A., Kantarcioglu, M.: Gunther Schadow. Jaideep Vaidya, privacy-preserving data integration and sharing. DMKD, Dan Suciu (2004)
12. Song, W., Wang, B., Wang, Q., Shi, C., Lou, W., Peng, Z.: Publicly Verifiable Computation of Polynomials over Outsourced Data with Multiple Sources. *IEEE Trans. Inf. Forensic. Secur.* **12**(10), 2334–2347 (2017)
13. Zhang, C., Xu, C., Xu, J., Tang, Y., Choi, B: GEM²-Tree: a gas efficient structure for authenticated range queries in blockchain. In: ICDE, pp. 842–853 (2019)

14. Hu, S., et al.: Searching an encrypted cloud meets blockchain: a decentralized reliable and fair realization. In: INFOCOM (2018)
15. Ruan, P., et al.: Fine-grained, secure and efficient data provenance on blockchain systems. *Proc. VLDB Endow.* **12**(9), 975–988
16. Wang, S., Zhao, D., Zhang, Y.: Searchable attribute-based encryption scheme with revocation in cloud storage. *PLOS One*, pp. 210–223 (2018)
17. Niu, J., Li, X., Gao, J., Han, Y.: Blockchain-based anti-key-leakage key aggregation searchable encryption for IoT. *Int. Things J.* **7**(2), 1502–1518 (2020)
18. Siris, V.A., et al.: OAuth 2.0 meets blockchain for authorization in constrained IoT environments. In: *Proceedings of IEEE World Forum on Internet of Things* (3), pp. 64–367 (2019)
19. Zhang, A., Bai, X.: Decentralized authorization and authentication based on consortium blockchain. In: Zheng, Z., Dai, H.N., Tang, M., Chen, X. (eds.) *Blockchain and Trustworthy Systems. BlockSys 2019. Communications in Computer and Information Science*, vol. 1156, pp. 267–272. Springer, Singapore (2019). https://doi.org/10.1007/978-981-15-2777-7_22
20. Rashid, M.A., Pajooh, H.H.: A security framework for IoT authentication and authorization based on blockchain technology. In: *BigDataSE*, pp. 264–271 (2019)
21. Yamauchi, R., Kamidoi, Y., Wakabayashi, S: A protocol for preventing transaction commitment without recipient’s authorization on blockchain. In: *COMPSAC*, pp. 934–935 (2019)
22. Widick, L., Ranasinghe, I., Dantu, R., Jonnada, S: Blockchain based authentication and authorization framework for remote collaboration systems. In: *WOWMOM*, pp. 1–7 (2019)