Evaluation of the performance of secure keyword search using bit-string signatures in the blockchain

Agipa AIGISSINOVA[†], Hieu Hanh LE[†], and Haruo YOKOTA[†]

[†] Department of Computer Science, Graduate School of Information Science and Engineering,
Tokyo Institute of Technology, 2–12–1 Ookayama, Meguro, Tokyo, 152–8552 Japan
E-mail: agipa@de.cs.titech.ac.jp, hanhlh@de.cs.titech.ac.jp, yokota@cs.titech.ac.jp

Abstract Data privacy is one of the biggest difficulties that every organization faces nowadays. All organizations should provide security protections on personal data against unauthorized access and modifications of any data. Healthcare is such an example where patient data should be handled properly and the leakage or alteration of this data assists in a huge privacy-neglecting issue. Immutability can be provided by adopting the blockchain technology in healthcare. Encryption is a key baseline to data privacy protection. This paper introduces a new approach to secure keyword search over encrypted data in the blockchain. In this paper, we propose the implementation of the bit-string signatures in the design of a data structure to secure search and access data while hiding the search pattern and share the patient data records between authorized organizations, while ensuring the privacy and immutability of the data. Besides, the secure cryptographic protocol - proxy re-encryption is applied to support proper encryption and decryption without revealing the keys and access control over secret data. The goal of this paper is the evaluation of the keyword search using bloom filters over encrypted data in the blockchain.

Key words Blockchain, Privacy, Security, search encrypted data algorithm

1. Introduction

The rising amount of data fundamentally changes the collection and treatment of them by organizations. However, every organization has a constant fear of data leakage, which happens every day and causes after it. Therefore, data privacy and confidentiality issues should be managed necessarily way. One of the most demanding fields, where data privacy should be maintained is healthcare. Healthcare holds an extensive amount of personal data, which includes everything from personally identifiable information (PII) to financial information, social number, education, medical history, family or living address.

Motivated thus, new proposals should be applied to prevent breaches of sensitive information and security incidents to ensure integrity and privacy of data.

Blockchain technology can be applied to save integrity and immutability od sensitive data in healthcare. The immutability property of the blockchain refers to that every change that was made to the data can be viewed to every participant of the network. Even though it increases the availability and transparency, unfortunately, it decreases the privacy and confidentiality of transferred data [1]. However, this challenge can be mitigated by applying encryption on data before outsourcing it on the blockchain network. But, keyword search over encrypted data in blockchain becomes another issue to handle [2].To search over encrypted data in the blockchain, the user should download the entire database on the blockchain and query the data locally. This approach is not practicable due to the huge data size and significant maintenance costs.

There are enough researches on searching over the blockchain database and applying encryption on data. In [3], the authors proposed a blockchain-based privacy keyword search system, which enables oblivious keyword searches in decentralized storage systems. This scheme is still in a theoretical stage and practicability of using full mature of the decentralized storage system was not covered. Authors [4] introduced a framework that combines the decentralized storage system InterPlanetary File System (IPFS), the Ethereum and attribute-based encryption technology. This scheme outsources data in IPFS, which cannot guarantee the appropriate behavior of nodes in the decentralized storage system and the immutability of data can be doubtful. The [5] and [6] proposed a trustworthy and private keyword search on encrypted decentralized storage systems by applying the searchable symmetric encryption. These schemes mostly focus on the fair exchange between users to prevent the fraudulent behaviors of them in searching operation, and searchable encryption can be computationally expensive. However, all the above methods store the data outside the blockchain and do not guarantee the data immutability and privacy in outsourced storage.

Therefore, we want to improve the security and integrity of the data by storing encrypted data in the private blockchain by ensuring immutability. Also, the evaluation of keyword search over encrypted data is aimed in this paper by applying bit-string signatures as a trapdoor for the query.

The paper is organized as follows. Section 2 gives background information for applied methods in our approach and related works. Section 3 is going to give an overview of the method proposal and system architecture. Section 4 is planned to describe and report the experiment results. The conclusion and future works will be summarized in Section 5.

2. Background

In this section, related works and a briefly overview for applied techniques: Blockchain, Hyperledger Fabric, Bloom filter and Proxy re-encryption are described.

2. 1 Related Works

Recently, the adoption of blockchain technology in secure medical data sharing systems become popular among researchers and enterprisers. There are several works considered in managing medical records such as electronic health records, clinical trial, and precision medicine. The work in [7] proposed a decentralized electronic medical record (MedRec) that uses blockchain technology to maintain electronic medical records, while patients can use authentication, verification, and unalterable data sharing. Also, MedBlock presented by [8], suggests storing medical data in the blockchain, to prevent a single point of failure in traditional databases which can be the target of attackers. Unfortunately, the evaluation of the search was not presented in these papers. Li et al. [9] proposed a novel patient-centric framework and access control system of data stored in semi-trusted servers. They applied an attribute-based encryption technique to encrypt the patient's health record. However, this scheme has a main disadvantage: system computation cost will be increased every time the user changes the access policy of the data. Once a user modifies his access polices.

2. 2 Blockchain

Blockchain is a distributed ledger technology, which allows building an immutable, transparent, secure and publicly accessible repository of data. It is the system that consists of nodes, communicating through consensus protocol. The fundamental structure of blockchain is a chain of linked blocks, of which consists of a hash value of the previous block [10]. Generally, blockchain can be classified into two types: permissionless and permissioned.

Permissionless or public blockchains allow every trustless participant can read or write transactions in the ledger. Bitcoin and Ethereum are the most famous examples of permissionless blockchains, which uses a Proof-of Work [11] and Proof of Stake [12] algorithms for ensuring network consensus. Both algorithms require the participating nodes to solve a mathematical puzzle to add newly mined blocks at a certain cost, either the consumption of computation or money.

Permissioned blockchain employs an access control layer to allow certain actions to be performed only by certain identifiable participants. For that reason, a permissioned network is highly reliable for enterprise applications that require security and identity [13]. The most widely known instances of permissioned blokchains are Hyperledger Fabric [14] and R3 Corda [15].

Hyperledger Fabric is one of the open-source blockchain project launched by Linux Foundation in early 2016 [14]. Unlike Bitcoin and Ethereum, Hyperledger Fabric does not have any cryptocurrencies, and the mechanism to verify transactions is PBFT (practical byzantine fault tolerance) [16].

2. 2. 1 Hyperledger Fabric

Hyperledger Fabric offers a modular approach, which means that endorsement, consensus and storage protocols can be easily constructed under the use case, where developers can create plug-in components. It differs from other known blockchain systems as it is private and permissioned. Hyperledger Fabric is focused on identity management and privacy of information, that everyone in the network must get access via authorization and verification system. The identity management service in Fabric is called MSP - Membership Service provider. All participants must be enrolled in MSP to be a part of the network. Rather than other blockchain platforms, it does not have own native cryptocurrency but permits the execution of chaincodes, i.e. smart contracts in the Hyperledger language [17]. Hyperledger runs chaincode, implemented in Go/JAVA/Nodejs language. Moreover, it allows the interaction of different organizations so that they can keep different replicas of stored data, execute the code associated with transactions and verify the result of transactions executed by different members. Transactions in Fabric can be made by invoking chaincode methods. The chaincode runs with a Docker container, which is isolated from peer machines and Fabric codes. Every chaincode has a constant state - key-value store. Chaincode allows two operations: read and write to make changes into the key-value store. Fabric supports two types of database for key-value store, LevelDB database [18] and CouchDB [19].

2. 3 Bloom Filter Management

The Bloom Filters is a space-efficient probabilistic data structure that supports fast membership queries. The data

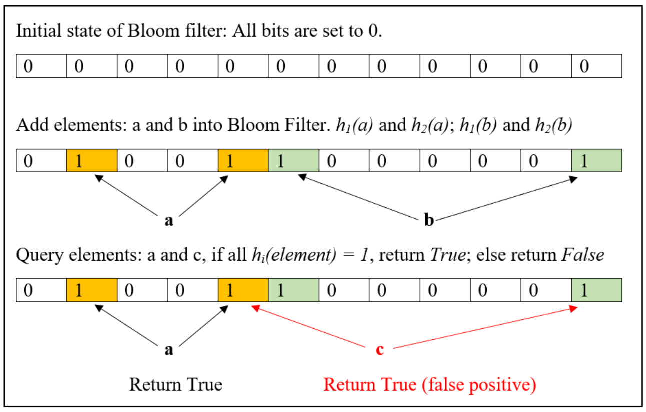

Initial state of Bloom filter: All bits are set to 0.
0 0 0 0 0 0 0 0 0 0 0 0
Add elements: a and b into Bloom Filter. h1(a) and h2(a); h1(b) and h2(b)
0 1 0 0 1 1 0 0 0 0 0 1
a
b
Query elements: a and c, if all hi(element) = 1, return True; else return False
0 1 0 0 1 1 0 0 0 0 0 1
a
c
Return True
Return True (false positive)

Figure 1 An example of Bloom filter with m=12 and k=2. Note that query element c results in false positive

structure was proposed by Burton H. Bloom in 1970 [20]. Bloom filter is a bit-vector of m bits which are all initialized to 0(Figure 1). There are k independent hash functions, $h(x) = (h_1(x), ..., h_k(x))$, employed for the Bloom filter to randomly map an element into array positions with a uniform distribution. To add an element to the filter, one should feed the element to k hashes and get k positions, then change those positions to 1 (Figure 1).

To query an element from a Bloom filter, one should calculate all the hash values of that element to obtain k array positions of the filter and thereafter checks if every of those k positions set to 1, then this query results in positive. If any bit at the k hashed position of the element is 0, the Bloom filter indicates that this element is not in the set. But sometimes, query through Bloom filter incurs false positive errors, i.e., incorrectly indicates that non-member element in the set of bit-vector. However, to realize acceptable false positive rate f_r is possible due to length m, the number of elements in the set n, and the employed hash functions k. In theory, the value of false positive rate can be estimated as follows:

$$f_r = \left[1 - \left(1 - \frac{1}{m}\right)^{nk}\right]^k \approx \left(1 - e^{-\frac{kn}{m}}\right)^k, \qquad (1)$$

where $(1 - 1/m)nk$ is very close to $e^{-kn/m}$ [21]. Therefore, to minimize the value of f_r, by minimizing $e^{-kn/m}$, we can derive the optimal value of k as:

$$k_{opt} = \frac{m}{n} \ln 2 \approx \frac{9m}{13n}. \qquad (2)$$

So false positive rate equals to $0.5k \approx 0.6185m/n$ [21]. This means that in order to maintain a fixed value of false positive rate with the defined k, the value of m must increase linearly with the number of elements n of the set [22].

To sum up, the accuracy of Bloom filter depends on the size of bit-vector, the number of elements in the set, and the number of hash functions applied to the filter. The rate of false positives is increasing by the number of elements added to the set.

2. 4 Proxy re-encryption

Proxy re-encryption is one type of cryptosystem, which allow third parties (proxies) to re-encrypt a ciphertext which has been encrypted for one party, so that it may be decrypted by another [23]. In this scheme, the ciphertext encrypted under user A's public key - can be re-encrypted that user B can decrypt it with his private key without revealing the user A's private key. Accordingly, user A allow proxy to re-encrypt the ciphertext by generating the re-encryption key for proxy. However, the proxy will not be able to gain any information about whether user A's key or encrypted data, since ciphertext is not fully decrypted by proxy. The re-encryption key is generated by a combination of user A's private key and the intended user B's public key. Thus, proxy re-encryption is flexible to create an access control management system and secure data sharing scheme.

The notion of proxy re-encryption first was introduced by Blaze et.al., in Eurocrypt'98 [24] and it was bidirectional, which leads to some security weaknesses. The first construction and improved scheme of PRE was proposed by Ateniese et.al., [25]. They demonstrated PRE for e-mail forwarding, distributed secure file systems and outsourced filtering of encrypted spam [25].

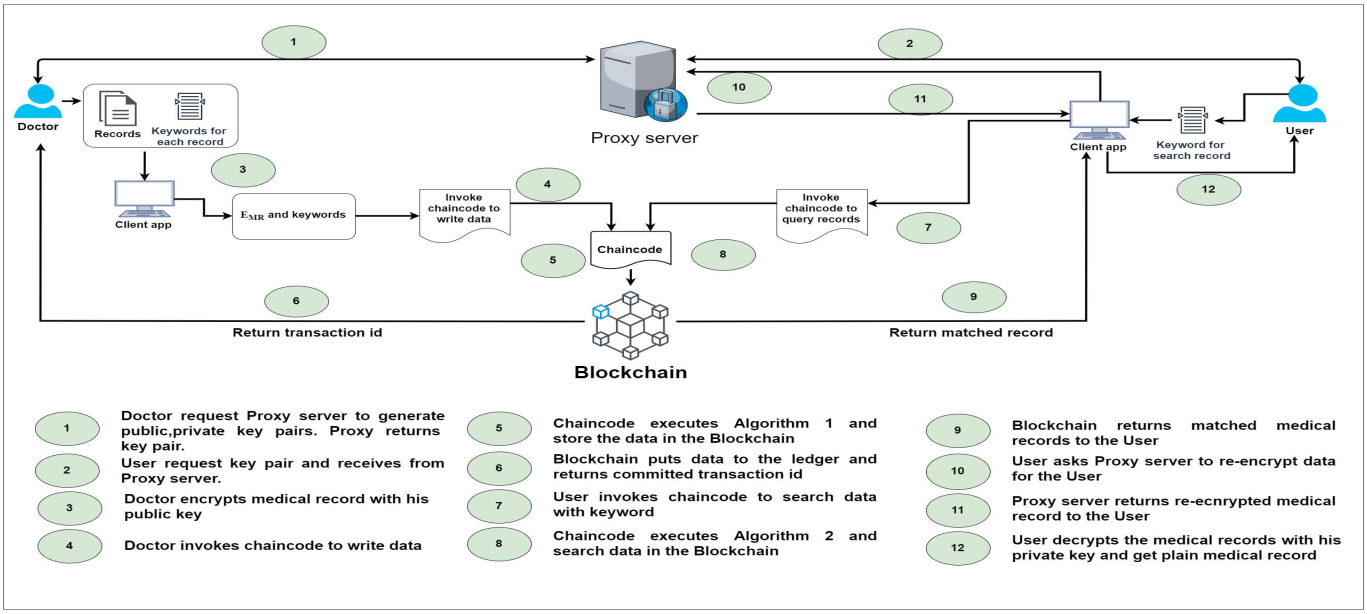
3. System architecture

In our signature-based proposal, we use a Bloom filter to represent the set of keywords from each data record as a fixed sequence of bits called a signature. Bloom filter is a reliable option in the case of space – and time-efficiency, which allows fast membership query processing. Importantly, searching processes using bloom filter are done by chaincode (smart contract) in the blockchain, and the information about keywords are protected from disclosure by the hashing function.

The main idea is that each record is represented by its signature, i.e., the sequence of bits that make up Bloom filter.

Data can be stored in Hyperledger Fabric as a collection of key-value pairs [26] and keys in the ledger must be unique. Thus, we use bit-string signatures as a key to represent the value – an encrypted data record. The encrypted data record is written to the ledger with concatenating transaction id with the word "Data" while the bit-string signature creates the key by concatenating the word "Bloom" with transaction id to store bloom filter.

Algorithm 1 (Figure 3) presents the pseudocode for the writing data record to the ledger by generating a bit-string vector for every record. To generate bit-string signatures,



1 Doctor request Proxy server to generate public,private key pairs. Proxy returns key pair.

2 User request key pair and receives from Proxy server.

3 Doctor encrypts medical record with his public key

4 Doctor invokes chaincode to write data

5 Chaincode executes Algorithm 1 and store the data in the Blockchain

6 Blockchain puts data to the ledger and returns committed transaction id

7 User invokes chaincode to search data with keyword

8 Chaincode executes Algorithm 2 and search data in the Blockchain

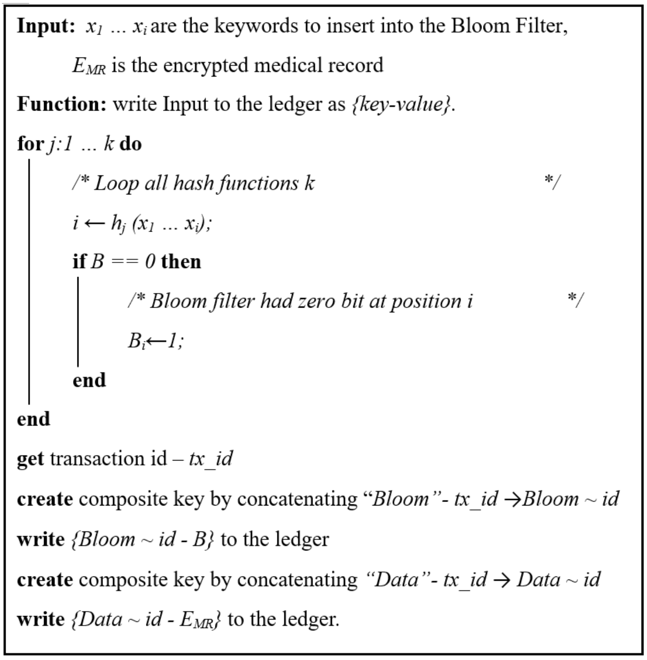
9 Blockchain returns matched medical records to the User

10 User asks Proxy server to re-encrypt data for the User

11 Proxy server returns re-ecnrypted medical record to the User

12 User decrypts the medical records with his private key and get plain medical record

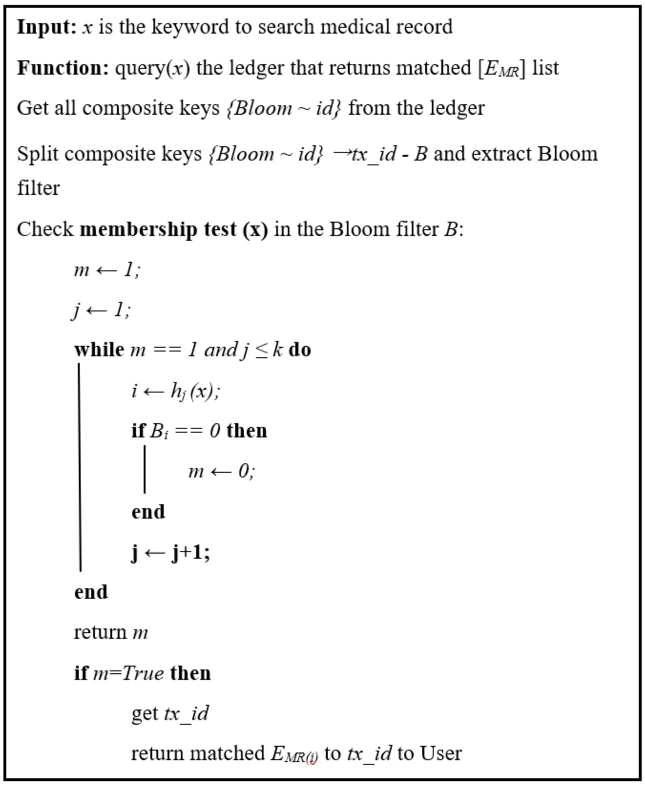
Figure 2 The system architecture



```
Input: $x_1$ ... $x_i$ are the keywords to insert into the Bloom Filter,
    $E_{MR}$ is the encrypted medical record
Function: write Input to the ledger as {key-value}.
for $j:1$ ... $k$ do
    /* Loop all hash functions k                    */
    $i \leftarrow h_j\ (x_1 \ldots x_i)$;
    if $B == 0$ then
        /* Bloom filter had zero bit at position i          */
        $B_i \leftarrow 1$;
    end
end
get transaction id – tx_id
create composite key by concatenating "Bloom"- tx_id →Bloom ~ id
write {Bloom ~ id - B} to the ledger
create composite key by concatenating "Data"- tx_id → Data ~ id
write {Data ~ id - $E_{MR}$} to the ledger.
```

Figure 3 **Algorithm 1. Write encrypted data records**

chaincode takes keywords, add them to bloom filter and generate the bit-vector. Then take the transaction id and create composite key, and store the data into the blockchain with composite keys and bit-vector signature, and encrypted data. For example, the transaction id is "192h00ff3498", the key for the data record is written as "Data-192h00ff3498" in the blockchain database. The key for bit-string signature is resulted to "Bloom-192h00ff3498" and the signature can look like "10100011101" depending on the length of the bit-vector.



```
Input: $x$ is the keyword to search medical record
Function: query($x$) the ledger that returns matched [$E_{MR}$] list
Get all composite keys {Bloom ~ id} from the ledger
Split composite keys {Bloom ~ id} →tx_id - B and extract Bloom filter
Check membership test (x) in the Bloom filter $B$:
    $m \leftarrow 1$;
    $j \leftarrow 1$;
    while $m == 1$ and $j \leq k$ do
        $i \leftarrow h_j\ (x)$;
        if $B_i == 0$ then
            $m \leftarrow 0$;
        end
        $j \leftarrow j+1$;
    end
    return $m$
    if $m$=True then
        get tx_id
        return matched $E_{MR(i)}$ to tx_id to User
```

Figure 4 **Algorithm 2. Search encrypted data records**

Algorithm 2 (Figure 4)shows the pseudocode for query ledger, chaincode checks a keyword for existence in the bloom filter and returns the matched encrypted data records. The fixed values of bits m and the number of hash functions k are defined initially in chaincode to preserve the initial state of Bloom filter since membership testing requires the same parameters as in the adding process.

The overall architecture of the proposed model are shown in Figure 2. The data set have been generated artificially and patient name or identity is anonymized in the data set to allow sharing among clinicians or government analytics. The three entities are included in our model:

1) The Data owner represents the Hospital and who owns and stores data records in the blockchain. Also, data owner generates re-encryption keys for users, who want to access the data.

2) User in the architecture represents Patients and Government analytics. They are allowed to access data by querying the blockchain ledger and later decrypt the data on his client application.

3) The Proxy server is responsible to generate private and public key pairs for the network participants and re-encrypt data by the request.

4) Blockchain store bloom filters and encrypted data records, and it can be accessed by invoking chaincode.

The workflow of the proposed model as follows for accessing the data record.

System Setup (Figure 2, Steps 1-3). In the setup phase, the data owner and user register to the Proxy server to generate the private/public key pairs. Also, they should identify their identities to the Fabric network.

Writing a data record (Figure 2, Steps 4-6). The writing the data records start with the encryption process by Data owner. The data is encrypted with the public key of the Data owner. Then, the Data owner invokes a chaincode to write the data. In this phase, chaincode executes Algorithm 1 (Figure 3) and sends the data to the blockchain. All blockchain network participants verify and validate the transaction, and add the transaction to the block. The transaction id is returned to the Data owner. It means that data is stored on the blockchain ledger.

Retrieving data records (Figure 2, Steps 7 - 12). To retrieve the data record, the User invokes the chaincode with intended to search keyword. Chaincode performs Algorithm 2 (Figure 4) and does a search operation in the blockchain. In this stage, matched encrypted data records are returned to the User. Now, the user requests the Data owner to generate a re-encryption key and Proxy server to re-encrypt the Figure 4 The system architecture data. The data owner generates a re-encryption key and sent it to the Proxy, then Proxy re-encrypt the data. After the user receives the re-encrypted data from Proxy, it can be decrypted by his/her private key in a client application.

4. Performance evaluation

The performance evaluation of a keyword search in the Hyperledger Fabric network is conducted. All experiments are operated on a VirtualBox version 6.0 workstation using Ubuntu OS 18.04. A host system is a machine with Intel(R) Core (TM) i7-8565U CPU, 1.99 GHz, 16 GB RAM, running Windows 10. For the blockchain service, Hyperledger Fabric version 1.4 network is created with Docker containers. The blockchain network contains endorser node, orderer node, three organizations with two peers for each.

The data set contains 10000 patient records and overall size is 3840 kB. This data set contains attributes as age, gender, disease conclusion, address, and others.

4. 1 Experiment Results

The experiments are conducted by varying the size of the data and the number of query requests that were sent to the blockchain network. In each experiment, the encryption, bloom filter generation, total time to put the encrypted data records to the ledger and searching operation are tested 10 times, and then the average processing time is recorded. Also, the 95% confidence level is estimated and the error bars on the graphs is calculated based on the standard deviation.

4. 1. 1 Evaluation of the Bloom filter's accuracy

The accuracy of the Bloom filter in our model, mentioned in Section 3, is performed to ensure the applicable false positive rate to sensitive data.

The number of elements, i.e. generated from attributes as age, gender and disease conclusion, $n = 9$. Experiment was done to data with size of 38.4kB. The results are given in Figure 5 and Figure 6. In the Figure 5 shows the dependence of a false positive rate to the number of bits m in the Bloom Filter, while Figure 6 indicates the variance false positives to the applied number of hash functions in the set.

This evaluation makes certain the reliance on efficiency of the Bloom filter to the number of bits and hash functions. Therefore, $f_r = 0.01$ false positive rate was achieved in the bit vector length m=74 and hash number k=6. These parameters will be used in later experiments.

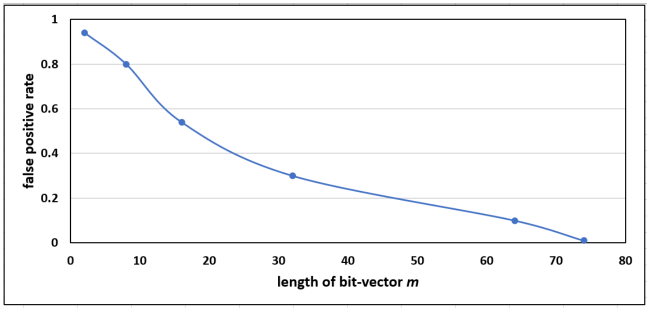


Figure 5 Dependency of the length of the bit vector to the false positive rate

4. 1. 2 Evaluation the Bloom filter and encryption performance

This experiment was performed to ensure that encryption and bloom filter generation do not require high computational cost. For this reason, first the total time of writing the data in blockchain is evaluated (Figure 7). Time is in seconds, and the number of records is increased from 10, 100,

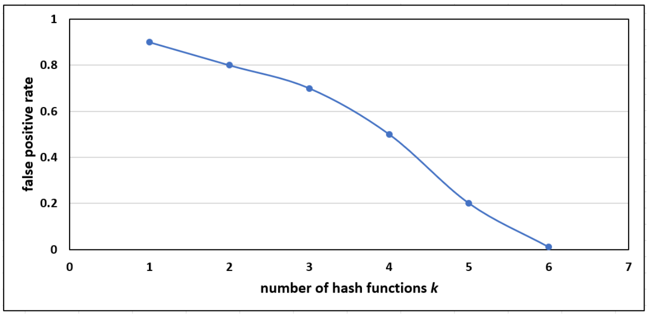


Figure 6 Dependency of the number of hash functions to the false positive rate

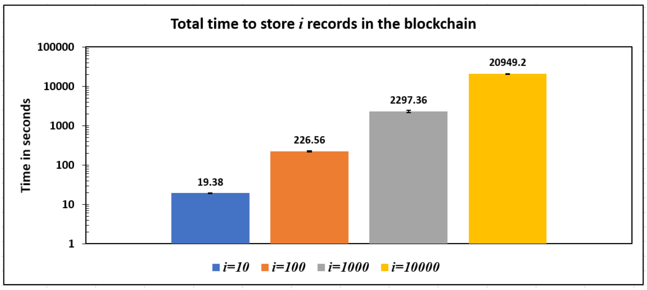


Figure 7 Total storing time the data records

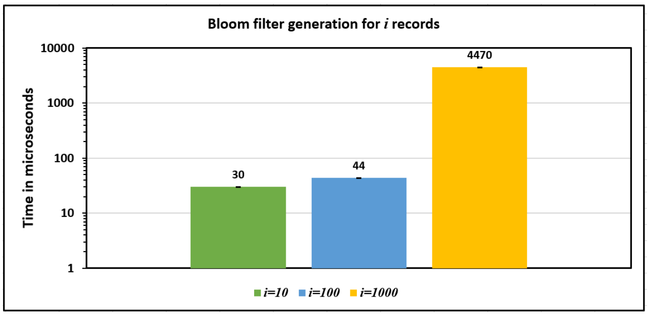


Figure 8 Bloom filter generation time of i records

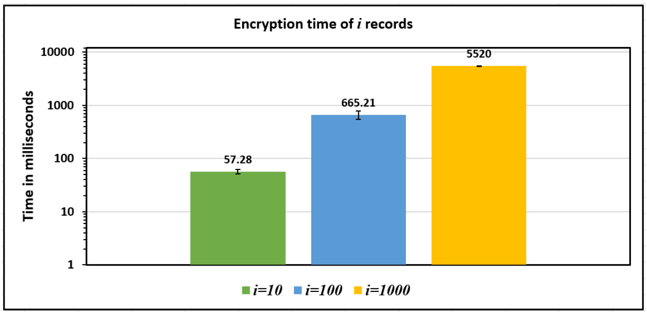


Figure 9 Encryption time of i records

1000 and 10000. Also, this time includes the verification and committing time the transaction by blockchain network participants.

As well as, the bloom filter generation and encryption time presented in Figure 8 and Figure 9, respectively. As stated in Figure 8, time increases proportionally to the number of records. According to Figure 9 encryption time for the number of records is presented and the estimated values result that the encryption process is very fast even to the increasing data size.

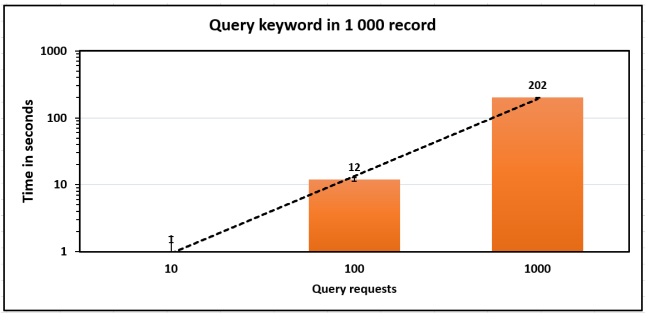


Figure 10 Searching time in the ledger with 1000 records

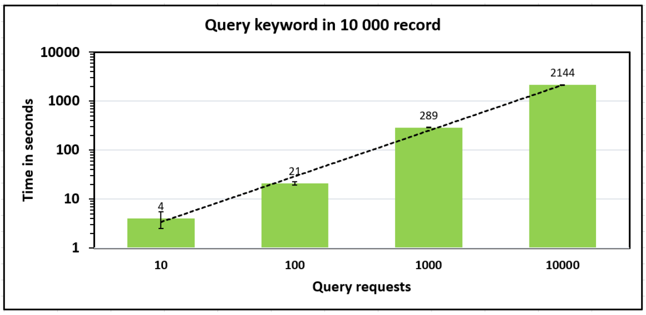


Figure 11 Searching time in the ledger with 10 000 records

This figures can ensure that Bloom generation is maximum *0.00002%* of total time to store the data in the blockchain, and encryption is *0.24%*. This findings can ensure the Bloom filter time-efficiency on the bit-string generation in the system and do not show overhead to the system.

4.1.3 Evaluation the querying process

The main purpose of the paper was to estimate keyword searching operation in the blockchain and this operation was conducted by varying the number of records in the ledger. It was helpful to realize the iteration time through the amount of the bit-string signatures to retrieve the data. Results are shown in Figure 10 and Figure 11. The trendline on the figures indicates the linearity of the searching time and it shows increasing time with the number of query requests to the blockchain.

That is to say, we can verify that the number query requests do not impact significantly to the system and user can search encrypted data in the ledger certainly without time overhead.

4.1.4 Effect of using Bloom Filter in the blockchain

This experiment was tested to verify effect of using bloom filter in multi-keyword search in the blockchain.

It was conducted for multi-keyword search on encrypted data over baseline method (without bloom filter) and the proposed method (with bloom filter) for the number of keywords: 2 and 3. The baseline method is to download all data records from the ledger, decrypt them and then apply a single-keyword search multiple times and take the intersection of the results as the final result. Accordingly, searching operation is executed without using bloom filter.

In the proposed method, the bloom filter is configured to do membership testing for multi-keyword at once. Then, only matched data records is decrypted as a succeeding outcome.

The figures of the baseline method and proposed method is presented in Figure 12. The comparison of time of the two methods in presented in Figures 13, 14, 15. The searching time decreased in the baseline method (without bloom filter) in the Figure 14 since the matching documents for searching first keyword was less than in Figure 13.

In terms of the baseline method, all the steps in the retrieving data records has time overhead than the proposed method. This method significantly affected by the downloading all data records stored in the ledger and decryption them in order to make search operations. For proposed method, the time does varies to the number of keywords and matched candidates to them in the ledger. Also, it contains the hash calculations for each keyword. However, the proposed method indicates a significantly outperform a baseline method to average 4807.1 times in time efficiency (Figure 15).

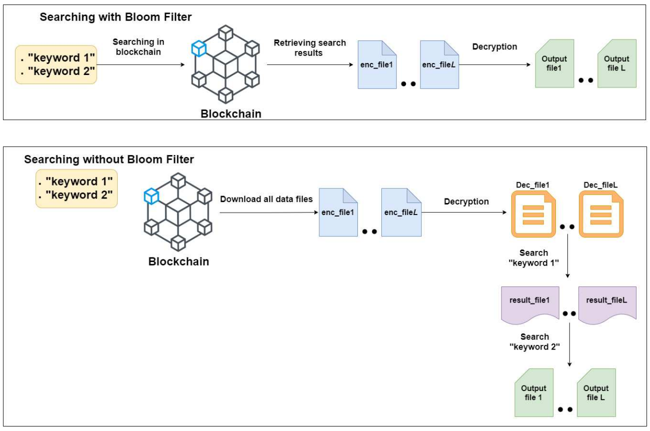


Figure 12 Searching with and without Bloom filter scheme

4. 2 Discussion

In this work, we proposed a blockchain-based data record sharing model to preserve the privacy and immutability of the data record. Our proposed model uses bit-string signatures, i.e., Bloom filter, permissioned blockchain, and proxy re-encryption scheme. This model allows share and store patient data records securely, retrieve the data without revealing the own private/public keys, control the access to the data and conveniently check the integrity of the data in the network. Consequently, issues related to the privacy and immutability of the data records are handled.

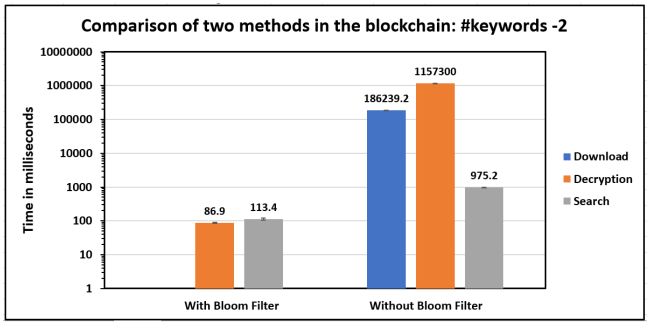


Figure 13 Searching with and without Bloom filter scheme, keywords-2

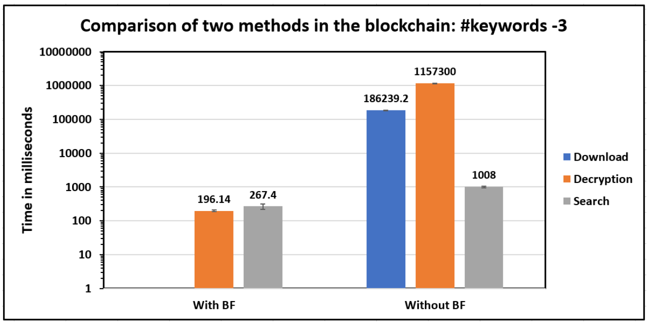


Figure 14 Searching with and without Bloom filter scheme, keywords-3

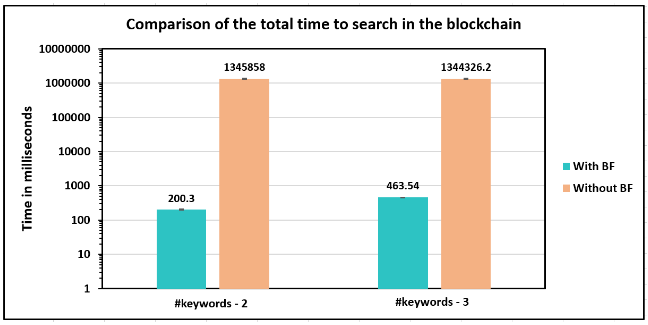


Figure 15 Total searching time of with and without Bloom filter scheme

The increasing number of attributes from the data set may increase the number of false positives in current Bloom filter's configuration. Thus, new configuration of parameters for increasing data set is required to accurate Bloom filter ' s accuracy.

Furthermore, some changeable parameters of Hyperledger Fabric configuration such as consensus protocol can increase the performance of the blockchain network.

5. Conclusions and Future Work

Data privacy has been found one of the demanding issues in the management of sensitive data, and healthcare with an increasing amount of data records needs new approaches to handle this issue.

This paper described a representative approach that allows us to store and search sensitive data in the blockchain network while preserving privacy, confidentiality, and immutability. There are several methods were used such as Bloom Filter, Proxy re-encryption and Hyperledger Fabric. Bloom filter allows the searching over encrypted data without overhead to the system, while proxy re-encryption is applied to support proper encryption and decryption without revealing the keys and access control over secret data. Hyperledger Fabric is used to represent the performance of the private blockchain and only authorized participants are allowed to join the network.

The evaluation of keyword searches over the encrypted data in the blockchain is performed and the searching time shows that it increases linearly with the number of a query request, and values do not show overload and could be acceptable. Also, we ensure that Bloom filter generation and encryption time requires insufficient overhead to the system and make sure the dependency on the accuracy of the Bloom filter to its parameters and find out proper parameter's to the later experiments.

Last but not least, we show the outperform of proposed method in searching over encrypted data in blockchain by comparing with the baseline method and get in average 4807.1 times in searching time.

In the future, we would like to perform evaluations for multi-keyword search with a larger data set. Furthermore, a complete study on the effect of changing the configurable parameters of Hyperledger Fabric and Bloom filter must be tested in future work.

References

[1] T.T. Kuo, H.E. Kim, and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," Journal of the American Medical Informatics Association, vol.24, no.6, pp.1211-1220, 09 2017.

[2] S. Tahir, and M. Rajarajan, "Privacy-preserving searchable encryption framework for permissioned blockchain networks," 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp.1628-1633, 07 2018.

[3] P. Jiang, F. Guo, K. Liang, J. Lai, and Q. Wen, "Searchain: Blockchain-based private keyword search in decentralized storage," Future Generation Computer Systems, 2017.

[4] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," IEEE Access, vol.6, pp.38437-38450, 2018.

[5] C. Cai, X. Yuan, and C. Wang, "Towards trustworthy and private keyword search in encrypted decentralized storage," 2017 IEEE International Conference on Communications (ICC), pp.1-7, 05 2017.

[6] C. Cai, J. Weng, X. Yuan, and C. Wang, "Enabling reliable keyword search in encrypted decentralized storage with fairness," IEEE Transactions on Dependable and Secure Computing, pp.1-1, 2018.

[7] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," pp.25-30, 08 2016.

[8] K. Fan, S. Wang, H. Ren, Yanhuiand Li, and Y. Yang, "Medblock: Efficient and secure medical data sharing via blockchain," Journal of Medical Systems, vol.42, no.8, p.136, 06 2018.

[9] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," IEEE Transactions on Parallel and Distributed Systems, vol.24, no.1, pp.131-143, 01 2013.

[10] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?," IEEE Cloud Computing, vol.5, pp.31-37, 01 2018.

[11] A. Gervais, G.O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, p.3–16, Association for Computing Machinery, 2016.

[12] E.P. of Stake FAQhttps://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ, [Online].

[13] P. Thakkar, S. Nathan, and B. Vishwanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," , 2018.

[14] C. Cachin, "Architecture of the hyperledger blockchain fabric," , 2016.

[15] R. Brown, J. Carlyle, I. Grigg, and M. Hearn, "Corda: An introduction," , 09 2016.

[16] M. Castro, and B. Liskov, "Practical byzantine fault tolerance," Proceedings of the Third Symposium on Operating Systems Design and Implementation, p.173–186, USA, 1999, USENIX Association.

[17] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, and et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," Proceedings of the Thirteenth EuroSys Conference, Association for Computing Machinery, 2018.

[18] GoLevelDB, "Leveldb key/value database in go," "https://github.com/syndtr/goleveldb", [Online].

[19] CouchDB, "Couchdb official website," http://couchdb.apache.org/., [Online].

[20] B.H. Bloom, "Space/time trade-offs in hash coding with allowable errors," Commun. ACM, vol.13, no.7, p.422–426, 07 1970.

[21] A. Broder, and M. Mitzenmacher, "Survey: Network applications of bloom filters: A survey.," Internet Mathematics, vol.1, 11 2003.

[22] L. Luo, D. Guo, R.T.B. Ma, O. Rottenstreich, and X. Luo, "Optimizing bloom filter: Challenges, solutions, and comparisons," IEEE Communications Surveys Tutorials, vol.21, no.2, pp.1912-1949, 2019.

[23] T.V. Ravindra, and J.S. K, "Secure group key sharing protocols and cloud system.," in Advanced Methodologies and Technologies in System Security, Information Privacy, and Forensics, ed. D. Mehdi Khosrow-Pour, pp.71-81, IGI Global, 2019.

[24] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," vol.1403, pp.127-144, 05 1998.

[25] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," ACM Trans. Inf. Syst. Secur., vol.9, no.1, p.1–30, 02 2006.

[26] Hyperledger, "Hyperledger Fabric Docs," https://hyperledger-fabric.readthedocs.io/en/release-1.4/, Online.