Search on Encrypted COVID-19 Healthcare Data in Blockchain-Assisted Distributed Cloud Storage

Basudeb Bera, Ashok Kumar Das, and Sajal K. Das

Abstract

In present circumstances, Coronavirus Disease-2019 (COVID-19) is a very significant health issue affecting human life all over the world. In an Internet of Medical Things (IoMT) environment, the deployed wearable Internet of Things (IoT)-enabled smart devices in a patient's body sense vital information. COVID19-related information is very confidential and private. Thus, the data can be stored into private-blockchain-based distributed cloud storage. For big data analysis, it becomes difficult to decrypt each block residing in the blockchain. Hence, only useful blocks can be searched and then decrypted in order to lessen the computational burden. To deal with this, we discuss a search technique on encrypted COVID-19 data for a blockchain-assisted distributed encrypted database. Finally, blockchain implementation has been also provided to show its effectiveness.

Introduction

These days, hospital services are rapidly becoming digitized in an Internet of Medical Things (IoMT) environment and also are mobilized by exploring the possibility of several smart technology (e.g., smart sensors and wearable devices) to not only leverage data, but also minimize the risks. At the same time, it can provide efficient and easy-to-access services via multiple communication channels. IoMT produces a huge volume of private and confidential data gathered from various sources including wearable IoT devices. Therefore, the system becomes most effective and highly scalable if and only if the distributed cloud (also, known as distributed database) accessibility can be integrated with the structure. In addition, due to this pandemic situation (COVID-19), most organizations prefer their employees to work from their homes, and most of the financial sector needs to operate remotely (from home). Furthermore, many users opt to outsource data to the distributed cloud server in order to avoid health risks as well as mitigate the burden on local storage. However, privacy challenges arise when sensitive information is stored in remote servers, which in turn raises a serious security concern. A single cloud server is not totally trusted, and security of the data in the financial sector becomes a major concern. Thus, single-cloud services have not been fully realized due to users' concerns about data privacy and security along with single-server failure. To overcome these challenges and limitations, instead of using the traditional database in cloud, the data can be stored in blockchain format in a distributed cloud as the traditional database has several security threats (e.g., SQL injection attacks and data poisoning attacks). The data, in the form of transactions, can be stored in an encrypted way in a block in a blockchain as the data is strictly private and confidential in healthcare-related environments. Once the encrypted data is committed into a blockchain inside the cloud, no unauthorized entity (also called an adversary) can delete, modify, or inject malicious data into the blockchain, because the blockchain maintains the immutability property, and the data is cryptographically protected using the elliptic curve-based public key encryption and digital signature, Merkle tree, and one-way cryptographic hash function.

Baudeb Bera and Ashok Kumar Das are with the International Institute of Information Technology, India.

Sajal K. Das is with Missouri University of Science and Technology, USA.

Digital Object Identifier: 10.1109/IOTM.001.2100125

Major Challenges

Users typically employ encryption techniques in order to mitigate the security concerns of confidential healthcare-related data, such as COVID-19 vaccine-related information. However, once the encrypted data is placed, no processing (e.g., ordinary search mechanisms) can be carried out on the outsourced data on the encrypted data. If someone wants to process the encrypted stored data from the distributed blockchain, in the worst case he/she needs to decrypt each and every encrypted block in the distributed cloud where the blocks are stored in the blockchain. Such a technique is not efficient and takes huge computational time; nor is it reliable.

To mitigate the above-mentioned issues, a searchable encryption (SE) technique is preferable, and it is broadly studied to allow searching on encrypted data without revealing the information in the distributed cloud domain [1]. SE techniques enable different kinds of searching approaches on the encrypted data and support various levels of security [2]. The SE scheme can be classified based on the type of searching mechanism [3]:

1. Keyword search
2. Regular expression search
3. Semantic search

In the following, we list the major research challenges in the domain of encrypted search in the blockchain:

- Although the SE technique performs in distributed system domains, such as an IoMT environment, there is a lot of issues that need be addressed, like improving search speed and searching over multiple encrypted datasets, and efficiency of the blockchain data for the distributed cloud servers.
- Recently, various practical attacks such as leakage-abuse attack [4], file-injection attack, and chosen keyword attack [3] have been reported on SE, which raise various serious concerns in SE security. One of the attacks is forward privacy, which leads to inefficiency. In addition, the majority of the existing SE approaches do not withstand forward privacy attack [1].

2576-3180/21/$25.00 © 2021 IEEE

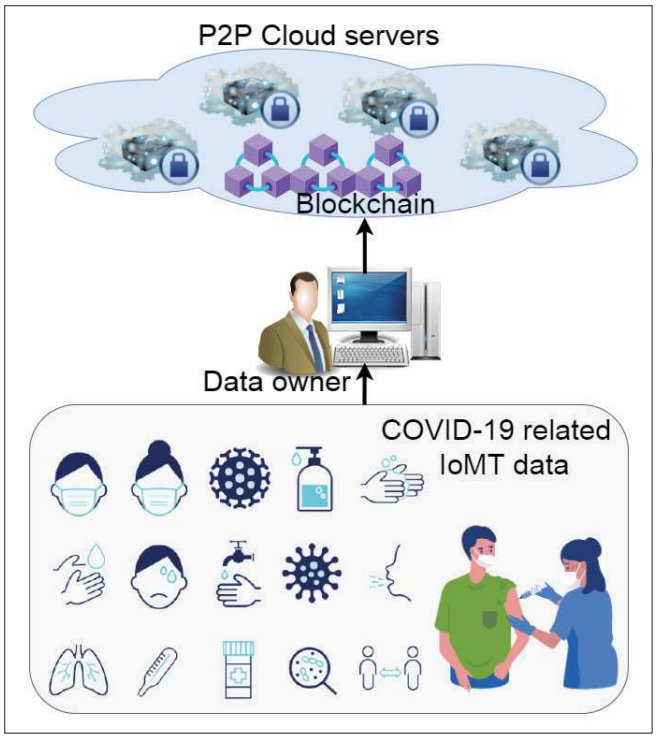

FIGURE 1. Network model for COVID-19 data store in blockchain.

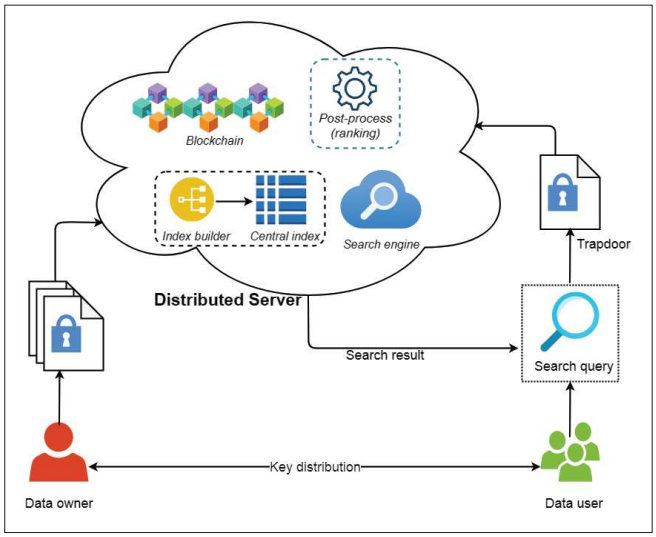

FIGURE 2. Searchable encryption process over blockchain based distributed servers (adopted from [2]).

System Models

In this section, we elaborate on the network as well as threat models of the proposed generalized SE technique for private blockchain in IoMT.

Network Model: In Fig. 1, COVID-19-related information generated by various entities involved in an IoMT environment needs to be stored in the blockchain (distributed cloud servers database). The data owner, being the administrator in a particular application such as a hospital, needs to securely gather the COVID-19-related patients' data and also securely load it into a peer-to-peer cloud server (P2P-CS) network. The data of IoMT applications is private and confidential; therefore, the data can be stored in encrypted formats such that without authorization, no one (including adversaries) can access the confidential data. It is worth noticing that the model is not restrcited only to COVID-19 related healthcare data, but it can be also applied for general healthcare applications as well. Since the data are stored in blockchain in encrypted formats, ordinary search is not viable. As a result, an SE technique must be adopted for search over the encrypted distributed database, which also helps to avoid data leakage attacks during search.

Threat Model: During the communication between various entities in the network, privacy leakage is treated as a serious security threat. In this article, we adopt the broadly accepted security threat model called the Dolev-Yao (DY) threat model [5]. According to the DY threat model, an unauthorized entity, adversary $\mathcal{A}$, not only can intercept the messages during communication, but do the tasks of deletion, modification, and also injection of the malicious information in the transmitted channel. Moreover, we consider the recently utilized security threat model known as Canetti and Krawczyk's model (CK-adversary model) [6], where $\mathcal{A}$ can intercept the transmitted messages that are exchanged over insecure channels. Under the CK-adversary model, $\mathcal{A}$ can hijack a session that may lead to compromise of the session state, including leakage of secret credentials such as secret keys during the session key agreement process, in addition to the capabilities under the DY threat model. Finally, we assume that the credentials stored in the memory of IoT-enabled wearable devices in the IoMT environment can be pulled out using power analysis attacks once a wearable device is physically stolen by $\mathcal{A}$.

Remark 1. *For secure communication among the various entities in the network, the entities need to establish secret (session) keys among them. The construction of session keys among the entities must be done with the help of both short-term secrets as well as long-term (permanent) secrets so that under the DY model and CK-adversary model, an adversary cannot generate or derive the session keys using the intercepted messages and temporal secrets from the communication channel and session hijacking attacks. This means that the ephemeral secret leakage (ESL) attack needs to be resisted during the secure communication against the adversary.*

Article Organization

The organization of this article is as follows. We discuss a generalized architectural model for SE over the distributed encrypted database. Next, we discuss recent works related to the SE domain. We then propose a generalized searchable encryption scheme for searching over encrypted distributed databases. We also virtually build a distributed system for blockchain with the help of Node.Js for blockchain creation and addition. Moreover, we implement the blockchain part and also the blockchain simulation. Finally, we conclude the article.

Generalized Searchable Encryption Model

Figure 2 provides a general overview of SE over the blockchain-assisted distributed servers environment.

Components

The considered model is composed of three components:

- **Data owner:** A data owner is an entity in the system whose task is not only to provide access to data users from the blockchain, but also to upload the encrypted information to the blockchain. Suppose the data owner has d_n number of gather documents that need to be outsourced to a distributed servers environment or blockchain. To protect the sensitive information, the owner encrypts the documents with his/her authorized key and stores them in blocks in the blockchain.
- **Data user:** A data user is another entity in the network who can search and collect the encrypted uploaded information from the blockchain only when he/she has authorization to do so. If a data user wants to search data from the encrypted servers in the blockchain, he/she must have an encryption key that will help to create some search queries (also called trapdoors). After that, the created trapdoors are forwarded to the server(s) to get back the search results, which contain a list of documents' identifiers or a list of relevant information.

Scheme	Techniques used	Advantages	Drawbacks/Limitations
Mamta *et al.* [9]	• Blockchain-based cloud-assisted searchable encryption	• Resistance to single server failure	• Heavy computational cost • Infeasible in real-time healthcare application
Chen *et al.* [10]	• Multi-keyword search-based public key encryption in blockchain-based cloud computing environment	• Application of smart contract • Secure against keyword guessing attacks	• Heavy computational cost • Infeasible in real-time healthcare application
Zhang *et al.* [11]	• Blockchain-based searchable encryption technology for medical data sharing	• Resistance to inside keyword guessing attack • Legitimacy of medical data	• Heavy computational cost • Not very applicable in IoMT
Wang *et al.* [12]	• Ciphertext access policy	• Supports multi-valued attributes	• Exposure of access policy using cipher components and public key component

TABLE 1. Existing state-of-the-art solutions with their used techniques, advantages, and limitations.

The data users are then allowed to decrypt the searched encrypted documents locally. For a real-time application, sometimes a data owner can also be treated as a data user.

- **Cloud server:** After receiving the gathered encrypted documents from data owners, the server accomplishes the following tasks:
  - –Store the uploaded encrypted documents.
  - –Construct a block that will be added into a blockchain. Each block in the blockchain will have a finite number of documents, say transactions (Tx_i).
  - –Perform a voting-based consensus algorithm, such as the widely recognized Practical Byzantine Fault Tolerance (PBFT) algorithm [7], to add the verified blocks into the blockchain.
  - –Provide a searching mechanism for the uploaded encrypted documents against trapdoors.
  - –Maintain and update the relevant data structures.

Searchable Encryption System over Private Blockchain Data

In this section, we discuss the problem statement and its associated possible remedies related to an SE system over a private blockchain.

Nowadays, blockchain is rapidly used as a potential resource for various sectors in order to outsource their confidential data to remote access with high accessibility. However, due to users' data privacy and security concerns, the data are stored in encrypted formats (encrypted transactions) in the blockchain (e.g., a private blockchain in the case of healthcare applications). Therefore, encryption techniques can be adapted to mitigate the security threats. Once the data is encrypted and stored in a blockchain in the form of transactions, no processing such as searching (simple plaintext searching) can be performed on the outsourced blockchain data in a distributed system [8]. This suggests that we need a searchable technique on encrypted transactions that are inside blocks stored in the blockchain without knowing the actual contents of the transactions. This also has an advantage in big data analytics where the decrypted transactions from the selected blocks using the SE technique lead to providing desirable predictions on the data stored in the blockchain. Another advantage associated with the SE technique in the blockchain is that it will reduce unnecessary computational burdens because not all the blocks are decrypted in the blockchain.

The SE techniques can be used to enable searching on the encrypted data over the blockhain servers. This mechanism performs various types of search techniques on the encrypted data, and at the same time, it also provides different levels of security. The SE techniques can be classified based on search types:

- Keyword search
- Regular expression search
- Semantic search

It is composed of a collection of algorithms, such as *KeyGen*, *Build-index*, *Trapdoor-Generation*, and *Search*, whose functionalities are discussed below.

- *KeyGen*: It is a process to create a key that encrypts the plaintext documents and decrypts the retrieved documents. A probabilistic key generation algorithm is used for KeyGen processes to compute a key based on a given security parameter. It is essential for documents storage in servers securely by the data owner. In addition, it is also needed to check that only authorized data users can access the documents. To encrypt the documents, two methods are mainly used:
  - –Symmetric encryption
  - –Public key encryption

  In symmetric encryption, the same key shared by the data owner and the data user is used simultaneously for encryption and decryption of the documents. On the other hand, in a public key encryption setting, the data owner uses two different types of keys: the first one for encryption, known as the public key, and the other for decryption, known as the private key.
- *Build-index*: An index structure is used in an SE system for tracking the instances of the keywords in documents. The initialization procedure of this index is known as Build-index, which takes a set of documents, d_n, and the key, k, which are then used in the KeyGen process as inputs. Next, the keywords from documents are extracted and then inserted into the index structure.
- *Trapdoor-Generation*: A data user runs this process by entering a search query. A key k is used in the encryption method by Trapdoor-Generation for the purpose of user search query. The server receives the encrypted trapdoors.
- *Search*: Given an encrypted trapdoor, the Search procedure is run by the server to match the documents, where the documents contain a set of keywords in the trapdoor. The user extracts the relevant documents and obtains the results by the decryption process.

In the SE technique, a user's sensitive data needs to be protected when it performs search queries at the server side; thus, no leakage of the information in plaintext data should occur, and the server must not be able to access the real data in the search queries as they are encrypted [2].

Related Work

Various cloud-based SE schemes have been proposed in a real-life environment [10, 13]. Recently, Mamta *et al.* [9] suggested blockchain-based cloud-assisted SE for healthcare systems. Their scheme resists a single server failure, which means it is free from a trusted authority by utilizing a distributed technology. In a healthcare environment, the devices are low battery powered and have limited memory. Their suggested scheme is based on bilinear pairings, which incur heavy computational cost; therefore, their scheme is infeasible in a real-time application related to the healthcare environment.

Chen *et al.* [10] suggested public key encryption with multi-keyword search in blockchain-based cloud computing, called BPKEMS. BPKEMS is also able to handle file data types and file updates to the cloud. It uses smart contract technology to make sure that the

Block Header (BH)
Block Version
Hash of the Previous Block
Merkle Tree Root
Timestamp
Proposer Identity
Public Key of Signer
Block Payload (BP)
List of d_n Encrypted Transactions (Tx_i)
Hash of the Current Block
Signature on current block hash

FIGURE 3. Structure of a block created by a server.

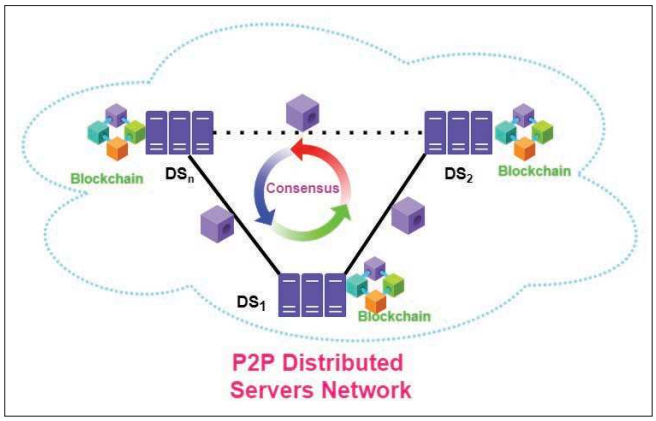

FIGURE 4. Blockchain formation by distributed servers.

fairness of transactions among the data owner and user without presence of a third party is preserved. Although BPKEMS is secure against keyword guessing attacks, it is not cost-effective because of the utilization of computationally heavy bilinear pairings. Therefore, it is not feasible for a real-life application.

Zhang *et al.* [11] proposed a blockchain-based SE technology for medical data sharing. Their suggested scheme is able to resist the inside keyword guessing attack (IKGA) and also offers the security to verify the legitimacy of medical data. Since their scheme is dependent on the bilinear-pairing-based cryptographic technology, it is not computationally efficient, which makes it difficult to apply for IoMT application as IoMT devices are resource-limited.

Wang *et al.* [12] designed a ciphertext access policy in which there is only one value per attribute. The AND gates are used in access structure formation on multi-valued attributes. However, Chaudhari *et al.* [14] found a security vulnerability in the scheme [12] in which the access policy of the scheme can be revealed with the use of the cipher components along with the public key component.

A summary of existing state-of-the-art solutions with their used techniques, advantages, and limitations is provided in Table 1.

Proposed Searchable Encryption Technique

In this section, we propose an index-based keyword search (IBKS) technique in order to search the transaction(s) residing in block(s) from a blockchain (Fig. 2). The simplest way to search a transaction (or a document) from a blockchain is the sequential search, where the data user needs to search the transaction in every block in the blockchain. However, the major challenge in the sequential search is that the time complexity quickly grows when the number of transactions in the blocks also grows. In other words, this type of search requires time complexity $O(N_d)$, where N_d denotes the total number of transactions for the blocks in the blockchain. In the following, we discuss the proposed encrypted search mechanism, which is very efficient compared to the sequential search

Encrypted Search

To mitigate the inefficiency problem in the sequential keyword search, we can adopt a data structure called an index, which holds a collection of keywords that are linked to their original documents by applying a document pointer (document identifier). In IBKS, the system only needs to verify the index structure for the matched keywords. In this proposal, the SE system generates an index for each transaction prior to encryption as well as uploads it into a block, and every index contains the encrypted form of some words associated with that transaction. If any block holds d_n number of transactions, the index table needs to also store d_n number of indices. Thus, each block holds an index table, and each index table is associated with a block sequence number. When a data user wants to search a particular transaction, he/she must enter some keyword and gets a block as a search result where the keyword matches a higher percentage. The detailed process can be described as follows:

- **KeyGen**$(1^k) \rightarrow sk$: A data owner runs this algorithm with a security parameter k and produces a symmetric key sk, which will be shared with the data user. In real time, the data owner and user are the same entities. The key sk is used to encrypt the transactions and the corresponding index.
- **Build-index**$(W, sk, Tx_i) \rightarrow Ind_w$: This algorithm generates an encrypted index Ind_w by the symmetric key sk of a transaction Tx_i for a keyword set W extracted from the transaction Tx_i. This algorithm is executed by the data owner.
- **Trapdoor-Generation**$(kws, sk) \rightarrow Trap$: This algorithm, executed by the data user, takes the symmetric key sk, and a possible search key word kws as inputs, and returns a secret trapdoor $Trap$.
- **Search**$(Ind_w, Trap) \rightarrow (\varnothing, location)$: The blockchain server or a distributed server in the blockchain center runs this algorithm by taking the input Ind_w as an encrypted index and a trapdoor $Trap$, and then returns a valid output location $location$ (e.g., the block sequence number) of the search data (transaction Tx_i) if the encrypted index Ind_w holds the keyword invisible inside $Trap$; otherwise, it returns an invalid result $\varnothing$. After obtaining the location of the searched transaction Tx_i, the data user can find the block and trace the location of that transaction in the block. Finally, the data user extracts the transaction and then extracts the plaintext by decrypting the encrypted transaction.

Blockchain Creation and Addition

The blockchain implementation process is done over a P2P-DS network. A server, DS, receives d_n (i.e., transactions threshold) number of transactions from the data owner and creates a block containing block version, previous block hash, Merkle tree root, timestamp, hash of current block, elliptic curve cryptography (ECC)-based public key of signer, and signature on the current hash (using Elliptic Curve Digital Signature Algorithm, ECDSA, signature generation function with the help of the signer's ECC-based private key), as shown in Fig. 3. Note that for blockchain purposes, we apply the Secure Hash Algorithm (SHA-256) as a cryptographic hash function, which produces 256 bits hash output on an arbitrary-length input string. Once a complete block is constructed, the block needs to be added to the blockchain by performing a consensus mechanism (e.g., we can apply the widely considered PBFT consensus algorithm), as shown in Fig. 4.

The details of the consensus process are explained below:

1. A leader (proposer), being an initiator, is first elected by a round-robin technique from the P2P-DS network.

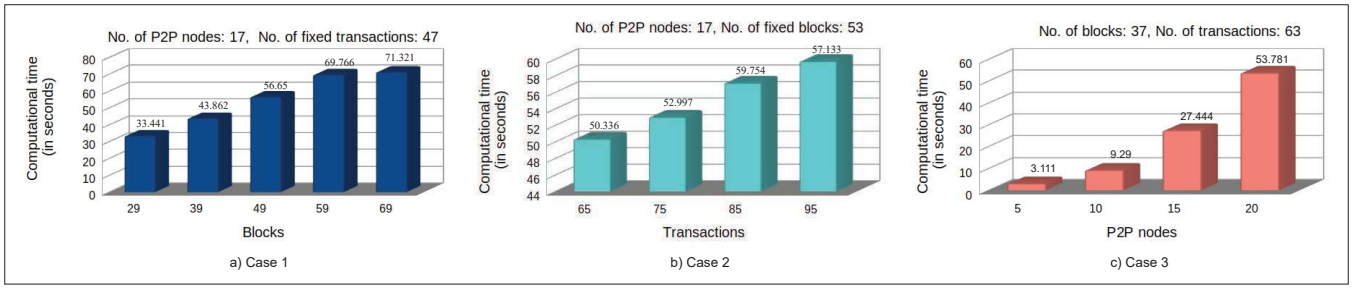

FIGURE 5. Blockchain simulation results.

2. The proposer proposes or broadcasts the newly created block to the entire P2P-DS network.
3. The receiver peer nodes (known as the followers) receive the proposed block and justify the following conditions:
   –The receivers verify the Merkle tree root over the transactions that were injected in the proposed block.
   –The receivers also verify the hash value of the current block (current hash block) and the signature on that hash value.
4. Once all the verifications are done successfully, the receivers reply with a valid block verification message.
5. After getting the reply messages from the receivers, the initiator validates whether the proposed block will be added to the blockchain or not.
6. When the total number of valid messages (V_m) is more than two times that of the faulty nodes (f) (faulty servers), that is, $V_m > 2f + 1$, the proposed block is inserted in the blockchain.
7. If the block is mined by the initiator, the initiator sends an acknowledge as a commit message to all followers.
8. Finally, the follower nodes add the proposed block into their distributed ledgers.

Blockchain Implementation Details and Results

In this section, the blockchain simulation is performed over a decentralized distributed system. For this purpose, we created a system virtually with a system configuration of Ubuntu 18.04.3 LTS, Intel® Core™ i5-8400 CPU @ 2.80 GHz× 6, Memory 7.6 GB, OS type 64-bit, disk 152.6 GB, and the script was written in node.js with VS CODE 2019. The decentralized distributed system has several distributed servers, and they are connected to each other to form a P2P-DS network. For the blockchain simulation, we consider the number of peer nodes (also called distributed servers) in the P2P-DS network as 17. As the blockchain technology runs over the distributed system, to make the decision or mine any new block in the blockchain center, a consensus algorithm needs to be run. Here, we have implemented the PBFT consensus algorithm with voting based mechanism for mining process. The simulation has been executed by considering the following two cases:

- **Case 1:** Fig.ure5a shows the simulation results for Case 1, where the simulation has done under a fixed number of transactions for each block, which is 47. Thus, we have fixed the number of transactions for each block, but varied the size of the chain. The simulation results for Case 1 demonstrate the first bar, which represents the total computational time (in seconds) (also total mining time or that to create the chain) having a chain size of 29. Similarly, the second bar of Case 1 shows the computational time with a chain size 39, and so on. Moreover, the results show that if the number of blocks mined is increased, the computational time also increases linearly.
- **Case 2:** In this case, the simulation results are provided in Fig. 5b, where the fixed number of mined blocks in each chain is 53. The simulation results in Case 2 clearly show that the time taken for the first chain with 65 transactions is less than the time taken for the second chain with 75 transactions. This is because verification time increases for more transactions in a block. The results also signify that the computational time increases whenever the number of transactions per block is increased, and it is linear.
- **Case 3:** Under this case, we have considered the blockchain size as 37, where each block holds a fixed number of transactions, which is 63. Thus, the number of the blocks in each chain is 37. We have then varied the P2P nodes in the DS network, where each server in the DS network contains a blockchain with a fixed length size. The number of P2P nodes varies from 5 to 20. The simulation results shown in Fig. 5c demonstrate that the variation of the computational time (in seconds) is linear when the number of P2P nodes increases.

Conclusion

In this article, we propose blockchain application to protect confidential information outsourcing in healthcare-related environments in a distributed manner. We then mention how to protect outsourcing data from several security threats. We discuss how the encrypted data can be searched (through searchable encryption methods) over the semi-trusted distributed servers environment without leaking any data during the search. Next, we present recent works on searchable encryption over encrypted distributed blockchain servers. Moreover, we present a searchable encryption technique and its overall architectural model. Finally, we provide our blockchain simulation by setting up a virtual distributed system. In the future, we would like to enhance the proposed framework with the help of lattice-based cryptograhic techniques applied to healthcare systems [15] so that the security of the framework can be significantly improved over traditional public-key-based cryptosystems such as ECC.

Acknowledgments

The authors thank the anonymous reviewers and the Associate Editor for their valuable feedback on this article, which helped us to improve its quality and presentation. The work of S. K. Das is partially supported by NSF grants under award numbers OAC-2104078, SaTC-2030624, DGE-1433659, and CNS-1818942.

References

[1] X. Song *et al.*, "Forward Private Searchable Symmetric Encryption with Optimized I/O Efficiency," *IEEE Trans. Dependable and Secure Computing*, vol. 17, no. 5, 2020, pp. 912–27.
[2] H. Pham, J. Woodworth, and M. Amini Salehi, "Survey on Secure Search over Encrypted Data on the Cloud," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 17, 2019, p. e5284.
[3] P. Chaudhari and M. L. Das, "KeySea: Keyword-Based Search with Receiver Anonymity in Attribute-Based Searchable Encryption," *IEEE Trans. Services Computing*, 2020. DOI: 10.1109/TSC.2020.2973570.
[4] D. Cash *et al.*, "Leakage-Abuse Attacks Against Searchable Encryption," *22nd ACM SIGSAC Conf. Computer and Commun. Security*, Denver, CO, 2015, pp. 668–79.
[5] D. Dolev and A. Yao, "On the Security of Public Key Protocols," *IEEE Trans. Info. Theory*, vol. 29, no. 2, 2018, pp. 198–223.
[6] R. Canetti and H. Krawczyk, "Universally Composable Notions of Key Exchange and Secure Channels," *Int'l. Conf. Theory and Applications of Cryptographic Techniques*, Amsterdam, The Netherlands, 2002, pp. 337–51.
[7] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," *3rd Symp. Operating Systems Design and Implementation*, New Orleans, LA, 1999.
[8] D. X. Song and D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," *IEEE Symp. Security and Privacy*, Berkeley, CA, 2000, pp. 44–55.
[9] B. B. Mamta *et al.*, "Blockchain-Assisted Secure Fine-Grained Searchable Encryption for a Cloud-Based Healthcare Cyber-Physical System," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 12, 2021, pp. 1877–90.
[10] Z. Chen *et al.*, "Blockchain-Enabled Public Key Encryption with Multi-Keyword Search in Cloud Computing," *Security and Commun. Networks*, vol. 2021, 2021, p. 6,619,689.

[11] Y.-l. Zhang *et al.*, "Deniably Authenticated Searchable Encryption Scheme Based on Blockchain for Medical Image Data Sharing," *Multimedia Tools and Applications*, vol. 79, no. 37, 2020, pp. 27,075–90.
[12] H. Wang, X. Dong, and Z. Cao, "Multi-Value-Independent Ciphertext-Policy Attribute Based Encryption with Fast Keyword Search," *IEEE Trans. Services Computing*, vol. 13, no. 6, 2020, pp. 1142–51.
[13] H. Pham, J. Woodworth, and M. Amini Salehi, "Survey on Secure Search over Encrypted Data on the Cloud," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 17, 2019, p. e5284.
[14] P. Chaudhari and M. L. Das, "Privacy Preserving Searchable Encryption with Fine-Grained Access Control," *IEEE Trans. Cloud Computing*, vol. 9, no. 2, 2021, pp. 753–62.
[15] R. Chaudhary *et al.*, "LSCSH: Lattice-Based Secure Cryptosystem for Smart Healthcare in Smart Cities Environment," *IEEE Commun. Mag.*, vol. 56, no. 4, Apr. 2018, pp. 24–32.

Biographies

Basudeb Bera (basudeb.bera@research.iiit.ac.in) received his M.Sc. degree in mathematics and computing in 2014 from the Indian Institute of Technology (IIT) (ISM) Dhanbad and his M.Tech. degree in computer science and data processing in 2017 from IIT Kharagpur. He is currently pursuing a Ph.D. degree in computer science and engineering from IIIT Hyderabad, India. His research interests are cryptography, network security, and blockchain technology. He has published 20 papers in international journals and conferences in his research areas.

Ashok Kumar Das [SM] (iitkgp.akdas@gmail.com) received a Ph.D. degree in computer science and engineering, an M.Tech. degree in computer science and data processing, and an M.Sc. degree in mathematics from IIT Kharagpur. He currently is working as an associate professor with the Center for Security, Theory and Algorithmic Research, IIIT Hyderabad, India. His current research interests include cryptography, system and network security, blockchain, and AI/ML security. He has authored over 285 papers in international journals and conferences in the above areas, including over 245 reputed journal papers. He is on the Editorial Boards of the *IEEE Systems Journal*, the *Journal of Network and Computer Applications* (Elsevier), *Computer Communications* (Elsevier), the *Journal of Cloud Computing* (Springer), *IET Communications*, and *KSII Transactions on Internet and Information Systems*.

Sajal K. Das [F] (sdas@mst.edu) is a professor of computer science and the Daniel St. Clair Endowed Chair at the Missouri University of Science and Technology, where he was the chair of Computer Science Department during 2013–2017. His research interests include cyber-physical security and trustworthiness, wireless sensor networks, mobile and pervasive computing, crowdsensing, cyber-physical systems and IoT, smart environments (e.g., smart city, smart grid, and smart healthcare), cloud computing, biological and social networks, and applied graph theory and game theory. He has published extensively in these areas with more than 900 research articles in high-quality journals and refereed conference proceedings. He holds five U.S. patents and coauthored four books. He serves as the founding Editor-in-Chief of Elsevier's *Pervasive and Mobile Computing Journal*, and as Associate Editor of several publications, including *IEEE Transactions on Dependable and Secure Computing*, *IEEE Transactions on Mobile Computing*, and *ACM Transactions on Sensor Networks*.