scientific reports

Check for updates

OPEN

SVBE: searchable and verifiable blockchain-based electronic medical records system

Norah Alrebdi[1], Abdulatif Alabdulatif[2]✉, Celestine Iwendi[3] & Zhuotao Lian[4]

Central management of electronic medical systems faces a major challenge because it requires trust in a single entity that cannot effectively protect files from unauthorized access or attacks. This challenge makes it difficult to provide some services in central electronic medical systems, such as file search and verification, although they are needed. This gap motivated us to develop a system based on blockchain that has several characteristics: decentralization, security, anonymity, immutability, and tamper-proof. The proposed system provides several services: storage, verification, and search. The system consists of a smart contract that connects to a decentralized user application through which users can transact with the system. In addition, the system uses an interplanetary file system (IPFS) and cloud computing to store patients' data and files. Experimental results and system security analysis show that the system performs search and verification tasks securely and quickly through the network.

With digital transformation and the appearance of some new concepts in the medical field, such as the Internet of Medical Things (IoMT), it has become important to digitize medical records completely. Electronic medical records (EMRs) provide several advantages, such as ease of accessibility from any device at any time, more efficiency, and saved time and cost. EMRs also enable medical organizations to adopt IoMT technologies, which can help effectively monitor patients statuses remotely and in real time, in addition to faster medical data processing. However, EMR systems face multiple challenges due to several aspects, such as security issues, privacy issues, and technical issues. Hence, EMR systems need a robust design to overcome these challenges and achieve the desired goals.

Many organizations and governments have adopted EMR systems based on cloud computing. Cloud computing provides many features, such as on-demand access, ubiquity, flexibility, and extensibility. The National Institute of Standards and Technology (NIST) defines cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"[1]. Despite the several advantages presented by cloud computing, the cloud suffers from central management. Centralizing management of the health records stored in the cloud may result in tampering, forgery, or unauthorized publication.

Several studies have proposed an EMR system based on blockchain techniques. Generally, a blockchain is a directed acyclic graph (DAG); it contains many blocks, and each block is connected with the preceding block by a hash[2]. Blockchain is regarded as a distributed ledger technology (DLT), which refers to storing distributed records and securing them using consensus protocols[3]. Moreover, blockchain is a decentralized technique, which means it does not require reliance on a third party to perform transactions and build trust between users. Additionally, the blockchain transaction record cannot change, thus ensuring untampered data. However, blockchain is characterized by transparency, which can threaten transaction privacy. In addition, storing large data is a major challenge in a blockchain environment[4].

The different blockchain functions and threats have made it challenging to adopt a specific blockchain methodology for EMRs. Consequently, the features in the proposed blockchain-based systems vary, as some proposals focus on one aspect without the other. Therefore, there is room for new suggestions that provide the multiple functions required for electronic medical systems. This research proposes a new system named the searchable

[1]Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia. [2]Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia. [3]School of Creative Technologies, University of Bolton, Bolton, UK. [4]University of Aizu, Aizuwakamatsu, Japan. ✉email: ab.alabdulatif@qu.edu.sa

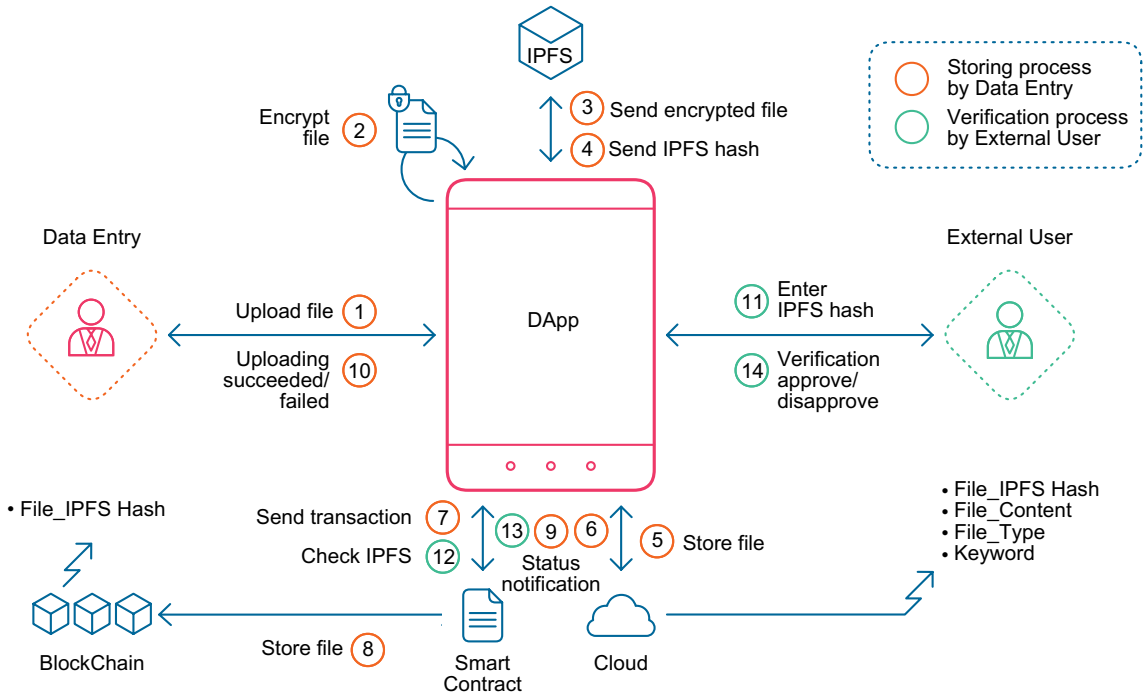

Figure 1. The system architecture of the file storing process conducted by the medical user and the verifying file process conducted by the external user. Several components are used to perform these two functions: SVBE DApp, IPFS, cloud, and blockchain. The stored data are kept in the cloud and blockchain databases.

and verifiable blockchain-based EMR System (SVBE). The proposed system will provide the ability to search, verify, and store encrypted EMRs based on blockchain. The main contributions of the paper are as follows:

- Design and implement a new EMR system based on blockchain that provides searchability and verifiability of the encrypted files.
- Apply the interplanetary file system (IPFS) and cloud storage, which help to reduce the costs in the proposed system.
- Conduct a comprehensive investigation of the developed system efficiency through several performance experiments on the following aspects: transaction execution time, medical file sizes, and cost of transactions. A security analysis of the system is also presented.

Methods

This section introduces the main components. In addition, it illustrates the goals of the system design, the workflow of each function, and the implementation details.

System architecture. This subsection presents the SVBE system architecture. Figure 1 illustrates the structure of the proposed system, in which the essential components are shown. The main components are as follows.

User application (decentralized application). This system provides a user application that can be used by medical agents, patients, and external beneficiaries. The application offers two functions related to the patients (add patients and verify patients) in addition to three functions related to the patient records. In this work, three functions related to the patient records are introduced. The first function adds the patient medical files. The second function is a validation service for the stored files that any external beneficiaries can use without worrying about tampering with original data based on the blockchain characteristics. The third function searches for any medical file stored in the system. One of the most important application tasks is to encrypt the uploaded files using the asymmetric cryptography algorithm ECC.

Interplanetary file system (IPFS). Medical systems usually deal with files that can be rather large, which causes difficulty in storing them directly on the blockchain. Hence, the SVBE system needs to deal with a reliable storage environment. The system adopted the IPFS decentralized environment to store the patients medical files. IPFS is considered a suitable solution for dealing with medical files due to several advantages, such as being free from a single point of failure and having high throughput in storage. An IPFS generates a unique hash for each stored file, giving users the ability to find that file by the hash address.

Cloud computing storage. The proposed SVBE system relies on storing all data in the cloud. The system stores all encrypted patient data such as personal data, file type, and the keywords of the stored files, in addition to the IPFS hash of the encrypted file. In addition, the system stores the actual content of the medical files after encrypting them to maintain data privacy.

Blockchain network. As mentioned earlier, since it is difficult to store large files in a blockchain environment, the system restricts the storage to the IPFS address of the medical files and all the encrypted patient personal information in the blockchain environment. The whole encrypted file is stored in the cloud.

Design goals. To be effective, the proposed system must achieve the following design goals:

- The system should ensure the security and privacy of the patient records. Therefore, the system has to apply strict rules to guarantee data integrity and confidentiality, and the system needs to prevent access to the records by any unauthorized entities.
- The system should have the ability to search for any file or patient despite the encryption.
- The system should provide the ability to verify the files by any outside related entity.
- The system must achieve high performance and low costs regarding latency, storage, and price to be suitable for adoption in the medical sector.

Workflow. The system provides three essential functions. The first function is storing a file in the patient record. The second function is verifying from any stored file. Finally, the third function is searching. The three functions are available only for the authorized health agents, except the verification function is available to any external entity. Each functional workflow is described in detail in the following subsections.

Files addition function. To add a new file, it is required that the patient who owns the file has a previous record in the system. Blockchain databases are not suitable for storing extensive data. To decrease the blockchain overhead, fewer data are stored in the blockchain database compared to the data stored in the cloud database. Thus, blockchain and cloud require different inputs sent by the Dapp based on the user inputs. The inputs of each party are explained as follows.

The following tuple shows the data that our system's decentralized application requires for adding a new file:

$$AddingFile_{DAPP} = (P_{id}, T_f, K_w, f) \tag{1}$$

where:

P_{id} = means the patient's identification number.
T_f = the type of file, whether it is a report, medical test, medical history, etc.
K_w = keywords that relate to the file content that can use it in the search process.
f = a new file of a patient that will store in a medical records system.

The following tuple describes the data that are sent to the blockchain upon adding a new file:

$$AddingFile_{BC} = (P_{id}, f_{hash}) \tag{2}$$

where:

f_{hash} = hash of the file generated from IPFS.

The following data are sent to the cloud:

$$AddingFile_{C} = (P_{id}, f_{hash}, T_f, K_w, f_{ec}) \tag{3}$$

where:

f_{ec} = encrypted file content.

As evident, the system excludes the entire content of the encrypted file stored in the blockchain due to the difficulty of storing large files or data in the blockchain database. Hence, it compensates by storing the encrypted file in the cloud. Storing the encrypted file in the cloud helps the medical user access and retrieve files directly from the cloud by entering the file IPFS hash. Figure 2 shows the workflow of the file adding process.

File verification function. The system provides a service to verify files circulating between users to prevent fraud and forgery. An IPFS hash is stored for each file exported from the medical system. Therefore, during the verification process, the user must enter the patient's ID in addition to uploading the file to be verified. The following tuple shows this step:

$$VerifyFile_{C} = (P_{id}, f_{ec}) \tag{4}$$

When the file is uploaded, all the steps applied when uploading a new file are performed, which means that if this file is trusted, it will get the same IPFS hash of the original file stored in the system. When the Dapp sends the IPFS hash to the blockchain, the blockchain informs the system that this file is identical to one of the files stored in the patient's record. Hence, a message will appear to the user stating that this file is trusted. In case of the file is forged or modified, its IPFS hash will not match any of the files stored in the patient record in the blockchain, and a message will appear to the user stating that the file is not trusted. The verification process relies upon the blockchain rather than the cloud because the blockchain environment is more reliable.

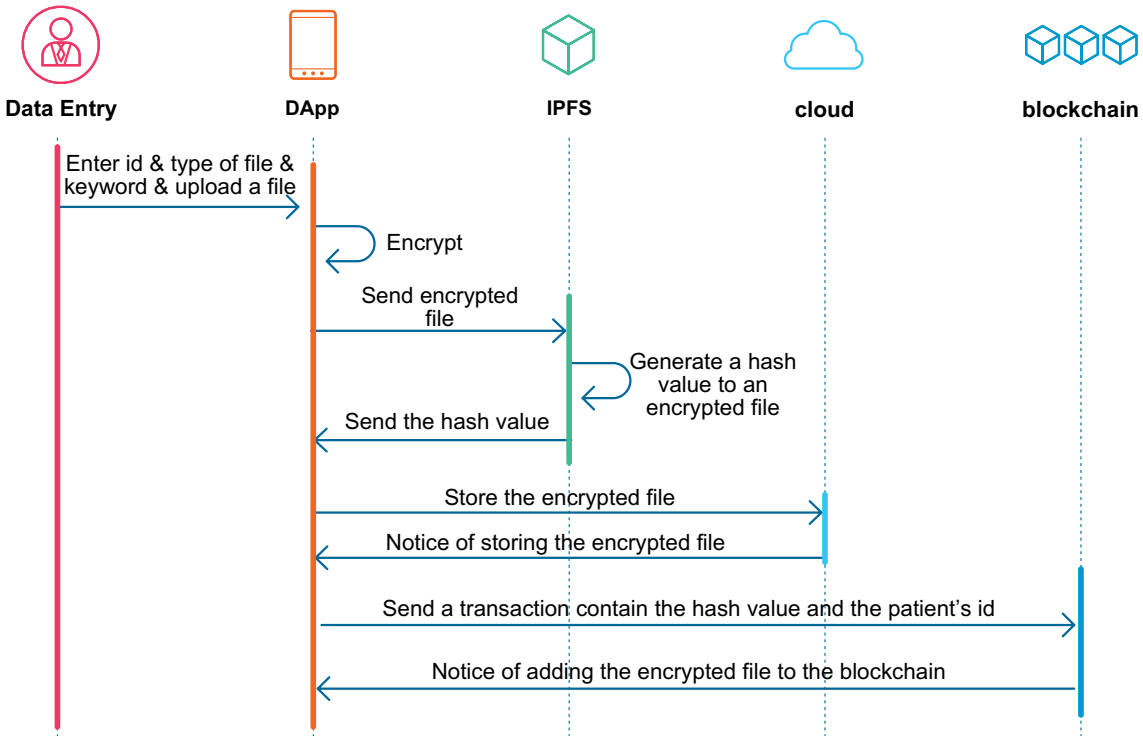

Figure 2. A sequence diagram of the process of adding files process to the system. The process starts by uploading a file, encrypting it, sending it to the IPFS to generate a hash, and then storing it in the cloud and blockchain.

Search function. Search is a required feature to make the system more functional. The medical user can search for any patients file they need by entering two mandatory inputs: the type of file and the keyword who wants to search for. The time range of the patient birth year is an optional input to narrow the search. If the medical user wants to narrow the search, they need to enter the time range they want to search through. The following tuple explains the search process.

$$SearchingFile_{DAPP} = (T_f, K_w, B_{starty}, B_{endy}) \tag{5}$$

where:

B_{starty} = demotes the start of the search time range.

B_{endy} = demotes the end of the search time range.

Figure 3 shows the search function. The search is performed directly by searching in the cloud database to reduce the costs involved in the blockchain environment. Moreover, an inverted index technique is used to index the keywords in the cloud to reduce the search cost. An inverted index form is a pair (key, value), where the key is a keyword and the value is a list of IPFS hash of files related to the keyword. Table. 1 shows the architecture of the files linked with the keywords in the cloud. The search results can be customized either by returning the file hash or the patient's ID.

Implementation.

The detailed implementation and tools used in the proposed system are presented. Additionally, the system design and description of the data stored in the cloud and blockchain databases are illustrated in the following subsections.

System settings. The proposed system consists of several parts: the user application, smart contract blockchain, IPFS, and the cloud database. For the blockchain, a simulated public Ethereum network is used. The model uses a local Ganache to run a local blockchain. Ganache provides 10 accounts, each with 100 ethers that are used to conduct the transactions. These accounts can be used by using MetaMask, a web browser extension that links the application with the smart contract. The smart contract was written in the Solidity language using the Remix code editor. The user application was written in Javascript. The system uses a React library to build the user frontend. The node package manager (NPM) of ECC was used to encrypt the data. To link the application with the blockchain network, the WebJs3 library was used, which is a library that interacts with MetaMask in the application. Furthermore, the system uses IPFS to generate a hash for each medical file. For cloud storage, the system uses two Firebase products: Cloud Firestore and Cloud Storage, to store both separate data and medical record files.

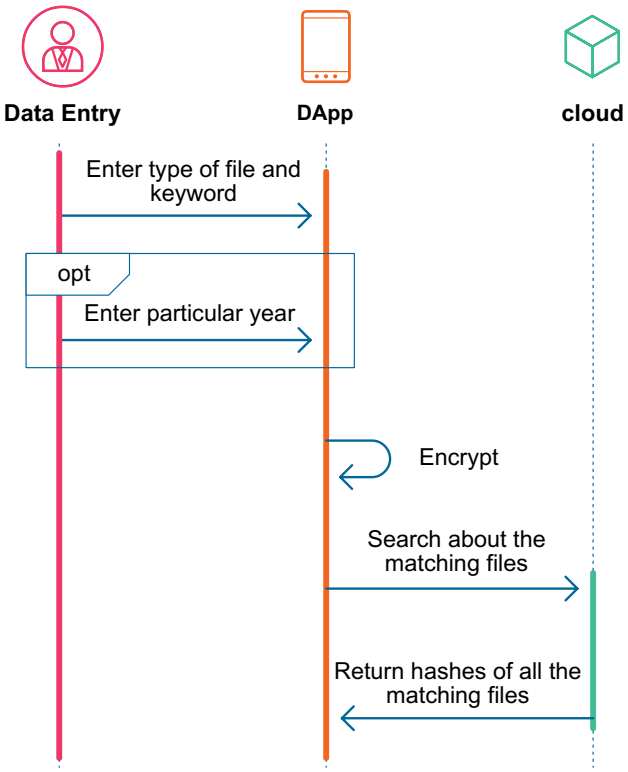

Figure 3. A sequence diagram of a search process in the system through which file type and the keywords are entered (additionally, it is possible to enter the time range), encrypted, and searched in the cloud, which together generate the results.

Key	Value
Rachitis	IPFS Hash of file 1, file 2,.
Allergies	IPFS Hash of file 3,.
Cancer	IPFS Hash of file 4, file 2,.

Table 1. Structure of the cloud-based proposed records system.

System design. A smart contract was built for the system, which the system administrator monitors. The smart contract conducts several functions, including add_file and verify_file. The pseudocode of these two functions is explained in Algorithm 1. The search function is conducted on the cloud database. The pseudocode of the search function is described in Algorithm 2.

Algorithm 1 add and verify File Functions

Input: id, IPFS hash

Add Function:

push the mapping value (id) to the key (IPFS hash) in the patient's records memory

Verify Function:

calculate the array length and assign it to $Flength$ variable
for $i \leftarrow 0$ to $Flength$ **do**
    if IPFS hash was stored with the id **then**
        $FileCheck = 1$
    end if
    break
end for
if $FileCheck = 1$ **then**
    return true
else
    return false
end if

Algorithm 2 search Function

Input: iFileType, ikeyword, iYear ▷ Year is optional

encrypt the input values
for each file in the cloud database **do**
    if ikeyword = stored keyword **then**
        if iFileType = stored FileType **then**
            if iYear = stored Year **or** empty **then**
                push IPFS hash into ResultsArray
            end if
            break
        end if
        break
    end if
end for
if $ResultsArray \neq empty$ **then**
    return $ResultsArray$ items
else
    return 0
end if

Cloud-based data storage. The proposed system uses two Firebase products: Cloud Firestore and Cloud Storage. Cloud Firestore is a NoSQL database that accepts several data types. The Cloud Firestore database is organized as a hierarchical structure. The root of Cloud Firestore is a collection; each collection has many documents depending on the stored data. Then, each document has several attributes. There are three collections in Cloud Firestore: patients, files, and keywords. Storing any patient on the patient collection requires five fields: ID, first and last name, and birthdate. The system stores the birth year as a separate attribute to allow an encrypted search for patients of a specified age. File collection contains two fields: patient ID, which links the file with its owner, and the IPFS hash of the file. The keyword collection includes two values: the keywords and the list of the IPFS hashes. The keyword is stored in a document, and each keyword document contains a list of IPFS hashes of files associated with the keyword. Cloud storage is a storage service suitable for storing files, voices, or images. Cloud storage stores the files and provides information about them, namely, the filename, size, type, and last modification. Cloud storage is used to store patient medical documents and images. Each file name with its IPFS hash to link each file with its owner's medical record.

Blockchain-based data storage. There are fewer data stored in the blockchain than data stored in the cloud. Generally, the data stored in the blockchain are divided into two parts: one for the patients and the second for the files. The patient part stores four attributes, which are the same as those stored in the cloud: ID, first and second name, and birthdate. Then, the file part stores two attributes: a patient's ID connected with the list of all IPFS hashes of its files.

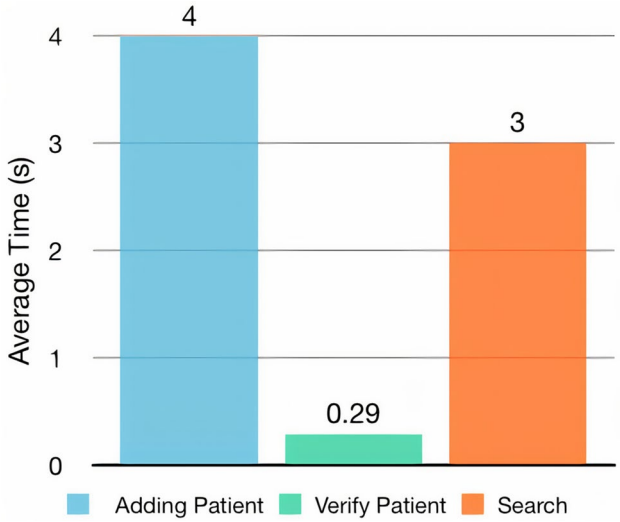

Figure 4. Transaction latency of functions.

Results

In this section, the proposed system is discussed in two aspects. The first aspect is system performance, where several experiments were conducted to calculate the transactions' latency, stored file sizes, and transactions' costs. For the credibility of the results, the average of 10 experiments of each function is shown. Each of the three aspects is discussed individually in a subsection. For the second aspect, the security of the system is analyzed.

Transactions latency. The time taken to complete the functions provided is an essential factor in evaluating system performance. Therefore, several latency calculations were performed that showed the performance of the proposed system. Figure 4 shows the average processing time of the patient adding function, patient verifying function, and search function. The table data shows that the verification time was the shortest, where it took fractions of a second. Furthermore, the time taken in the search function was less than that of adding patients. All three functions took a short amount of time. Notably, adding file function is conducted by using both databases: blockchain and cloud databases. In contrast, the verifying file function is conducted using the blockchain database only, while the search function is conducted using the cloud database only. Moreover, the patient adding function requires an initial step to ensure that the patient does not already exist in the system; this may explain the long time it took by the add patient function compared with the verification function.

Moreover, adding and verifying file functions were tested on four different file sizes. In addition, the time taken for the adding and verifying file processes on IPFS was calculated separately. Figure 5 shows the latency results of adding and verifying files in both situations. The figure shows that the latency of adding and verifying files usually increased with the increase in file size. However, the verification process through the IPFS took the most time during the four experiments, such that the entire verification process required 0–1 s more than the IPFS verification. The longest time was 24 s, which was associated with the file adding process taken with a file size of 1000 kilobytes (kB). Therefore, the transaction latency calculations were logical. Thus, this test achieved the fourth goal of the system design goals.

File size. The size of medical files varies and may tend to be large. In addition, encryption contributes to increasing the file size, making storing these files in the blockchain a costly process. To solve this problem, only the IPFS hash of the encrypted files is stored in the blockchain. In contrast, the entire encrypted content is stored in the cloud, as the cloud requires lower costs for storing large files. Figure 6 shows the size of the original files and the size after encryption using the elliptic curve cryptography (ECC) algorithm. It is clear from the figure that the file size increases after encryption by six times the original size.

Transactions costs. In the SVBE system, the functions are divided into two types: call functions and transactions. The transaction is a function that writes on the blockchain or cloud database, while the call function is the function that reads from the databases. Each transaction requires a cost specified based on the number of parameters, the structure of the function in a smart contract, and the used data types. The call function does not have a cost. Both add_patient and add_file functions in the proposed system are considered transactions, and the verify_patient, verify_file, and search functions are call functions. Table 2 shows the average amount of gas used by the two transactions: add_patient and add_file and the costs in US dollars. The value of the conversion used in the table is $2205.65, which is the current value of one ether. For the gas price, the standard value was used which is 20 GWEI. The high cost of the add_patient function may be due to the structure of the function. The add_patient function requires another function responsible for checking if the patient exists in the system

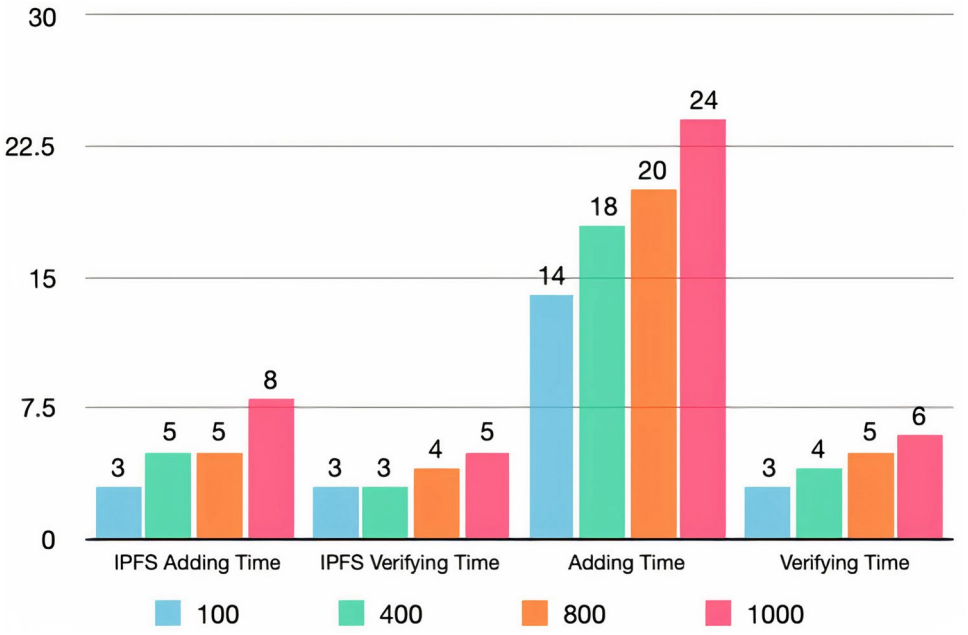

Figure 5. Transaction latency of adding and verifying files.

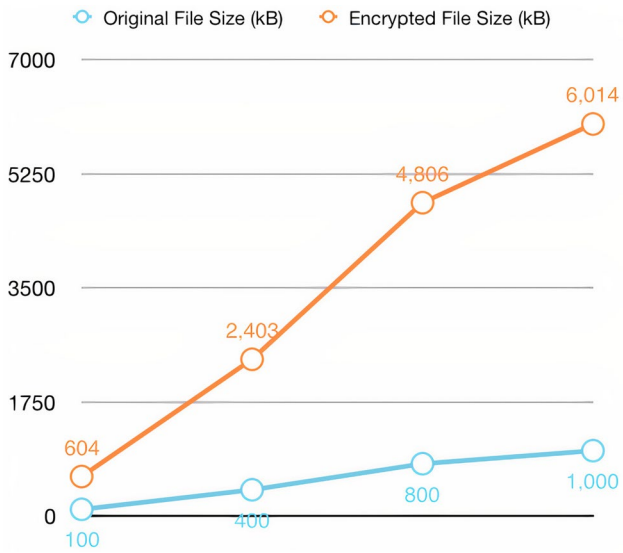

Figure 6. Comparison of the file size before and after encryption. The blue line illustrates the unencrypted file size, and the orange line indicates the encrypted file size.

Function name	Gas used	Cost (dollar)
Add patient	421,901	18.6
Add file	93,968	4.7

Table 2. Transaction cost of adding patients and Files.

beforehand or not. By contrast, the cost of the add_file function is lower than add_patient, which may demonstrate the benefit of storing an IPFS hash of the file instead of the entire file content.

Regarding the experimental results of the system performance, the system provides high performance in the latency aspect. In contrast, the increase in the size of the encrypted files is relatively large, but cloud storage can solve the problem, as it is suitable for storing such sizes. Despite the high cost of adding patient transactions,

the system provides several functions that do not require costs. Generally, the experimental results achieve the fourth system design goal. However, the system may need to reduce the cost further to improve the performance in this aspect.

Security analysis. The proposed system enables the search and verification of medical records securely and privately. The system is secured in several ways, such as using blockchain technology characterized by immutability. The immutability feature prevents any entity from tampering with or modifying the recorded data in the blockchain environment[5]. Thus, the system guarantees the integrity of the stored data[6]. Furthermore, the data entry process is limited to specific entities that the system administrator has to identify previously. Thus, this helps ensure the validity of the entered data and the inability of unauthorized parties to enter any data. In addition, limiting data entry to a certain number of entities makes it possible to investigate the cause when any error occurs in the entry data process. Therefore, these restrictions help to achieve the first system design goal.

Moreover, to ensure the patient privacy, any data and documents must not be stored in plaintext. Thus, the patient's files and personal data need to be encrypted before storing them in the system. Moreover, the system provides data confidentiality by preventing any unauthorized access or obtaining data explicitly. Thus, only the authorized entities can access the keys that decrypt the patient records and content. Hence, the system does not need to frequently share the keys through the network, protecting the keys from some attacks. Furthermore, the system uses asymmetric cryptography, which protects against eavesdropping attacks. More specifically, the encryption algorithm used in this system is ECC, which provides strong security, although it uses a short key size compared to other encryption algorithms[7]. Moreover, the encrypted files are stored in decentralized IPFS. The decentralization characterized by blockchain and IPFS protects against a single point of failure. In addition, decentralization is a peer-to-peer method, meaning it does not need trust in any third party to conduct transactions.

Regarding the search function, providing a search function using multiple keywords may lead to disclosing some data or guessing it, threatening patient data privacy. The system provides the search function using only one keyword to preserve privacy, although it can use the time range to obtain more effective results. Additionally, the search results can be controlled and made to contain only the IPFS hashes of the patients files instead of the IDs or the patient names to maintain patient anonymity. Thus, this achieves the second system design goal.

The system provides the file verification function to enable the permitted external entities to ensure the authenticity of the medical files even though they are encrypted. Consequently, the user who wants to verify a file does not need to obtain the plaintext file, as the ciphertext can do the job. Thus, this maintains patient data privacy and achieves the third goal of the system design goals, which is reduced patient privacy exposure.

Performance and security analysis show that the proposed system achieves the required design goals. In addition, the system has many features, such as flexibility and availability, so that different devices and systems can use it.

Discussion

Multiple studies have proposed several solutions to address the attacks and security threats based on different technologies[8,9]. For example, Tan et al.[10] introduced an approach called HoneyNet that includes threat detection and situational awareness of the artificial intelligence of things. Furthermore, information-centric networking was used to enhance communication security in smart grid[11]. Tan et al.[12] developed a traceable and direct revocation schema for medical records. In contrast, to enhance the communication performance, three types of nonlinear RF chain structures that reduce the power consumption of multiple-input multiple-output wireless communication systems were designed in[13].

Furthermore, several techniques are used to preserve data privacy. For example:[14] used attribute-based encryption (ABE) integrated with the 0-1 coding technology to enhance the encryption performance of the internet of health things data. Hang et al.[15] used ABE with a parallel outsourced decryption method. Additionally, a new method of secure arrangement based on matrix eigenvalue calculation was proposed in[16]. Moreover, the authors in[17] proposed enhanced retrieval models between the IoT and the cloud.

Regarding the blockchain-based systems, Wang et al.[18] proposed a new certificateless signature scheme integrated with the blockchain. Additionally, Xiong et al.[19] developed an efficient blockchain batch verification scheme using the elliptic curve digital signature algorithm. Moreover, Liu et al.[20] proposed a secure framework that used blockchain with mobile-edge computing to provide secure data sharing. Additionally, several studies have proposed medical records systems based on a blockchain. For instance, Liu et al.[21] designed a system that improves diagnosis processes in electronic health systems. Rahman et al.[22] introduced a tamper-proof health electronic record management system. Instead of using a blockchain to store health records, a blockchain-based healthcare system has been proposed to store addresses of mobile devices and sensors for the pervasive social network (PSN)[23]. Similarly, Xia et al.[24] stored the URLs of Fast Healthcare Interoperability Resources (FHIR) instead of the actual medical records. In contrast, a cloud-based electronic health records system was proposed based on blockchain and the IPFS[25]. Encrypted files are stored in the IPFS and linked to the blockchain through a patient ID and address. In addition, Jabarulla and Lee[26] proposed a blockchain-based management system to store and share medical images securely. The authors in[27] proposed a system for sharing medical data that guaranteed patient privacy by allowing only authorized parties to use it. Moreover, a new system was proposed in[28] to securely share records in emergency states using a multiparty computation circuit. In contrast, to enhance patients' control of their data, Xia et al.[29] suggested a system that allows the patients to give or restrict the access permissions of their files.

Searchable symmetric encryption (SSE) was first introduced by Song et al.[30] to secure searching on encrypted data. Curtmola et al.[31] used a single keyword in search. Chen et al.[32] developed a search method using multiple

keywords. Moreover, Cash et al.[33] proposed an SSE framework that concentrates on conjunctive search and Boolean queries. As an extension of Cash et al.'s work[33], Faber et al.[34] developed more capabilities: search in a range, wildcard, substring, and phrase queries. The authors in[35] proposed a solution for keyword typos based on the fuzzy search algorithm. Kamara et al.[36] provided the SSE schema using an inverted index. The red-black tree index was used in[37]. A tree-based index schema was introduced by Xia et al.[38] and conducted on encrypted cloud data. Moreover, Xiru et al.[39] proposed a searchable system of the encryption contents using keywords binary tree scheme to enhance the searchability in the blockchain. Similarly, some studies used SSE with blockchain to enhance search efficiency and ensure stored data privacy, such as in[40,41]. Several EMR systems based on a blockchain environment have been introduced. Moreover, some studies have provided search features in their blockchain EMR system. However the published studies do not propose a medical system that provides both features: the ability to search for encrypted medical files and the ability to verify them. This paper presents a design of an electronic medical records system that ensures privacy, security, searchability, and verifiability based on the blockchain technique.

Received: 6 October 2021; Accepted: 14 December 2021
Published online: 07 January 2022

References

1. Mell, P. & Grance, T. *The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology* (Technical Report, The National Institute of Standards and Technology, 2011).
2. Jabarulla, M. Y. & Lee, H. N. Blockchain-based distributed patient-centric image management system. *Appl. Sci. (Switz.)* **11**, 1–20 (2021).
3. Lai, R. & Chuen, D. L. K. Blockchain—From public to private. In *Handbook of Blockchain, Digital Finance, and Inclusion* (eds Chuen, D. L. K. & Deng, R.) (Academic Press, 2018).
4. Mazlan, A. A. *et al.* Scalability challenges in healthcare blockchain system—A systematic review. *IEEE Access* **8**, 23663–23673 (2020).
5. Fang, H. S. A., Tan, T. H., Tan, Y. F. C. & Tan, C. J. M. Blockchain personal health records: Systematic review. *J. Med. Internet Res.* **23**(4), e25094 (2021).
6. Ye, H. & Park, S. Reliable vehicle data storage using blockchain and IPFS. *Electronics (Switzerland)* **10**, 1130 (2021).
7. Benil, T. & Jasper, J. Cloud based security on outsourcing using blockchain in e-health systems. *Comput. Netw.* **178**, 107344 (2020).
8. Ding, F., Zhu, G., Alazab, M., Li, X. & Yu, K. Deep-learning-empowered digital forensics for edge consumer electronics in 5G HetNets. *IEEE Consum. Electron. Mag.*https://doi.org/10.1109/MCE.2020.3047606 *(2020).*
9. Yu, K., *et al. Secure artificial intelligence of things for implicit group recommendations* (2021). CoRR, arXiv:abs/2104.11699.
10. Tan, L., Yu, K., Ming, F., Chen, X. & Srivastava, G. Secure and resilient artificial intelligence of things: A HoneyNet approach for threat detection and situational awareness. *IEEE Consum. Electron. Mag.*https://doi.org/10.1109/MCE.2021.3081874 *(2021).*
11. Yu, K., Arifuzzaman, M., Wen, Z., Zhang, D. & Sato, T. A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid. *IEEE Trans. Instrum. Meas.* **64**, 2072–2085 (2015).
12. Tan, L. *et al.* Towards secure and privacy-preserving data sharing for COVID-19 medical records: A blockchain-empowered approach. *IEEE Trans. Netw. Sci. Eng.*https://doi.org/10.1109/TNSE.2021.3101842 *(2021).*
13. Gong, Y., Zhang, L., Liu, R., Yu, K. & Srivastava, G. Nonlinear MIMO for industrial internet of things in cyber-physical systems. *IEEE Trans. Ind. Inf.* **17**, 5533–5541 (2021).
14. Li, H. *et al.* An efficient ciphertext-policy weighted attribute-based encryption for the internet of health things. *IEEE J. Biomed. Health Inform.*https://doi.org/10.1109/JBHI.2021.3075995 *(2021).*
15. Feng, C. *et al.* Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV. *IEEE Trans. Veh. Technol.* **69**, 13784–13795 (2020).
16. Song, J., Han, Z., Wang, W., Chen, J. & Liu, Y. A new secure arrangement for privacy-preserving data collection. *Comput. Stand. Interfaces* **80**, 103582 (2022).
17. Wang, T. *et al.* A privacy-enhanced retrieval technology for the cloud-assisted internet of things. *IEEE Trans. Ind. Inform.*https://doi.org/10.1109/TII.2021.3103547 *(2021).*
18. Wang, W. *et al.* Blockchain-based reliable and efficient certificateless signature for IIoT devices. *IEEE Trans. Ind. Inform.*https://doi.org/10.1109/TII.2021.3084753 *(2021).*
19. Xiong, H. *et al.* On the design of blockchain-based ECDSA with fault-tolerant batch verication protocol for blockchain-enabled IoMT. *IEEE J. Biomed. Health Inform.*https://doi.org/10.1109/JBHI.2021.3112693 *(2021).*
20. Liu, L. *et al.* Blockchain-enabled secure data sharing scheme in mobile-edge computing: An asynchronous advantage actor-critic learning approach. *IEEE Internet Things J.* **8**, 2342–2353 (2021).
21. Zhang, A. & Lin, X. Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain. *J. Med. Syst.* **42**, 1–18. https://doi.org/10.1007/s10916-018-0995-5 (2018).
22. Rahman, M. S., Khalil, I., Mahawaga, P. C., Bouras, A. & Yi, X. A novel architecture for tamper proof electronic health record management system using blockchain wrapper, in *BSCI 2019—Proceedings of the 2019 ACM International Symposium on Blockchain and Secure Critical Infrastructure, co-located with AsiaCCS 2019*, 97–105 (Association for Computing Machinery, Inc, 2019). http://dl.acm.org/citation.cfm?doid=3327960.3332392.
23. Zhang, J., Xue, N. & Huang, X. A secure system for pervasive social network-based healthcare. *IEEE Access* **4**, 9239–9250 (2016).
24. Peterson, K., Deeduvanu, R., Kanjamala, P. & Boles, K. A blockchain-based approach to health information exchange networks (2017), in *NIST Workshop Blockchain Healthcare*, vol. 1, 1–10 (2016).
25. Nguyen, D. C., Pathirana, P. N., Ding, M. & Seneviratne, A. Blockchain for secure EHRs sharing of mobile cloud based e-health systems. *IEEE Access* **7**, 66792–66806 (2019).
26. Jabarulla, M. Y. & Lee, H.-N. Blockchain-Based Distributed Patient-Centric Image Management System. *Appl. Sci.* 2021, Vol. 11, Page 19611, 196 (2020).
27. Xia, Q., Sifah, E. B., Smahi, A., Amofa, S. & Zhang, X. BBDS: Blockchain-based data sharing for electronic medical records in cloud environments. *Information (Switzerland)* **8**, 44 (2017).
28. Parthasarathy, S., Harikrishnan, A., Narayanan, G., Lohith, J. J. & Singh, K. Secure Distributed Medical Record Storageusing Blockchain and Emergency Sharing Using Multi-Party Computation. 2021 11th IFIP Int. *Conf. on New Technol.Mobil. Secur.* NTMS 20211–5 (2021).
29. Madine, M. M. *et al.* Blockchain for giving patients control over their medical records. *IEEE Access* **8**, 193102–193115 (2020).
30. Song, D. X., Wagner, D. & Perrig, A. Practical techniques for searches on encrypted data, in *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, 44–55 (IEEE, 2000).

31. Curtmola, R., Garay, J., Kamara, S. & Ostrovsky, R. Searchable symmetric encryption: Improved definitions and efficient constructions, in *Proceedings of the ACM Conference on Computer and Communications Security*, 79–88 (2006).
32. Chen, L., Qiu, L., Li, K. C., Shi, W. & Zhang, N. DMRS: An efficient dynamic multi-keyword ranked search over encrypted cloud data. *Soft. Comput.* **21**, 4829–4841 (2017).
33. Cash, D. *et al.* Highly-scalable searchable symmetric encryption with support for Boolean queries, in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). LNCS*, vol. 8042, 353–373 (2013).
34. Faber, S. *et al.*, Rich queries on encrypted data: Beyond exact matches, in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9327, 123–145 (Springer Verlag, 2015).
35. Wang, B., Yu, S., Lou, W. & Hou, Y. T. Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud, in *Proceedings—IEEE INFOCOM*, 2112–2120 (Institute of Electrical and Electronics Engineers Inc., 2014).
36. Kamara, S., Papamanthou, C. & Roeder, T. Dynamic searchable symmetric encryption, in *Proceedings of the ACM Conference on Computer and Communications Security*, 965–976 (2012).
37. Kamara, S. & Papamanthou, C. Parallel and dynamic searchable symmetric encryption, in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). LNCS*, vol. 7859, 258–274 (2013).
38. Xia, Z., Wang, X., Sun, X. & Wang, Q. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE Trans. Parallel Distrib. Syst.* **27**, 340–352 (2016).
39. Liu, X., Wang, G., Yan, B. & Yu, J. KCB-BC-SSE: A Keyword Complete Binary Tree Searchable Symmetric Encryption Scheme using Blockchain. *Procedia Comput. Sci.* **187**, 377–382 (2021).
40. Li, H., Tian, H., Zhang, F. & He, J. Blockchain-based searchable symmetric encryption scheme. *Comput. Electr. Eng.* **73**, 32–45 (2019).
41. Chen, L., Lee, W. K., Chang, C. C., Choo, K. K. R. & Zhang, N. Blockchain based searchable encryption for electronic health record sharing. *Futur. Gener. Comput. Syst.* **95**, 420–429 (2019).

Acknowledgements

The authors would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

Author contributions

N.A. and A.A. developed the SVBE model and drafted the main manuscript, N.A., C.I., and Z.L. adapted the presented methods to conduct the experiments, N.A., A.A., and C.I. conducted the experiments, N.A., A.A., and Z.L. wrote the manuscript, A.A. and Z.L prepared figures and analysed the results. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.