PRIVACY-PRESERVING SEARCHABLE ENCRYPTION SCHEME BASED ON PUBLIC AND PRIVATE BLOCKCHAINS

Syeda Samreen Fatima
CSE Department
Shadan Women's College Of Engineering and Technology,
Hyderabad, India
Samreensyeda028@gmail.com

Dr. K. Palani
CSE Department.
Shadan Women's College of Engineering and Technology,
Hyderabad, India
principalswcet2020@gmail.com

ABSTRACT

Users who employ cloud-based data outsourcing benefit from its convenience, but they also run the danger of having their private information compromised and data altered. By validating the search results, searchable encryption technology ensures the integrity of the data and allows users to conduct keyword searches on encrypted data while maintaining their privacy. Still, there are several related issues that need to be addressed, like poor verification efficiency and unpredictable query returns. In light of this, this study suggests a public and private blockchain-based Privacy-Preserving Searchable Encryption (PPSE) system. We outsource matching encrypted documents to a public blockchain and first store an encrypted index on a private blockchain. The encrypted index is used to find the encrypted documents. This technique can decrease the blockchains' storage overhead while enhancing transaction execution performance and data security. Additionally, in order to ensure data privacy and the accuracy of access control verification, we employ a smart contract to implement a secondary verification access control mechanism and limit data users' access to the private blockchain through authorization. Lastly, the security analysis and experimental findings show that, in contrast to current schemes, the suggested technique may ensure the effectiveness of the query in addition to enhancing the security of encrypted data.

I. INTRODUCTION

The requirements for processing and storage on cellphones, Organizations show a clear trend of outsourcing data to outside parties, like cloud servers, as the need for processing and storage resources rises. Prior to outsourcing to the cloud, Data Owners (DOs) typically decide to encrypt their data because outsourced data often contains sensitive information. Therefore, under the presumption of ensuring data privacy, looking for encrypted data has grown to be a significant issue. Song et al.[1] originally suggested Searchable Symmetric Encryption (SSE) as a solution to this issue. A number of related investigations were later started as a result of this research.

The primary goal of searchable encryption, or SE, is to retrieve data that conventional encryption techniques are unable to decrypt. SE techniques typically function by creating an encrypted index. The service provider receives both the encrypted data and the index from the DO. For a given keyword, the Data User (DU) provides the search token; the service provider uses the token and encrypted index to run search algorithms and find matches. Earlier techniques used scanning to return findings, and as the amount of data in the database increased, the efficiency of the scheme declined linearly. By creating an index and extracting keywords, several academics have enhanced SE technology to address the efficiency issue and limit the query complexity to the keywords present in the file set. Unfortunately, the majority of the initial schemes are static and cannot be changed on the fly. Users typically need to update data while keeping it on a cloud server. Dynamic SE technology has evolved to address these demands, increasing the SE scheme's adaptability and accessibility. Nevertheless, the security analysis becomes more intricate since the attacker can watch the data updating process and identify the connections between keywords and files to steal or alter the data. Thus, two security attributes—forward privacy and backward privacy are suggested in the security analysis of dynamic SE. While backward privacy assures that deleted documents won't leak any information as a result of later search operations, forward privacy makes sure that newly created files won't be connected to any searches conducted in the past.

Researchers to store data since the rise of Bitcoin have used Blockchains. As a result, blockchains are being employed extensively in the SE field[2], [3]. Blockchains may effectively secure the integrity and privacy of data and prohibit the modification of cloud storage information since all actions are visible and dependable, and data on them is open, unchangeable, and irrevocable. This work offers a Privacy-Preserving Searchable Encryption (PPSE) system with an access control mechanism based on private and public blockchains to address the issues of data privacy, integrity, and correctness. The PPSE outsources the search question to a smart contract, which generates an accurate and unchangeable answer without requiring the DO to be verified, and leverages the blockchain to replace the central server. Furthermore, smart contracts are used to implement an access control mechanism that limits DUs' access to the private blockchain, hence preventing harmful assaults and safeguarding data privacy. The following advances the domains of blockchains and SE with this research:

- ➢ Encrypted documents are located through encrypted indexes, and encrypted indexes are uploaded to private and public blockchains, respectively. This can lower blockchain storage overhead and enhance transaction execution efficiency and data security. In addition, the PPSE does not require the creation of an additional verification mechanism in order to ensure the integrity of query results.
- ➢ By implementing smart contracts, we add a supplementary verification access control method to safeguard data privacy, stop privacy leaks, and avert network attacks. We include it into the private blockchain and validate DUs twice to limit their entry to the private blockchain, enhancing the accuracy of verification.
- ➢ Much data is evaluated and analyzed using a simulation experiment of the local test network. According to the security analysis and experimental findings, the PPSE can guarantee query efficiency while simultaneously enhancing data security when compared to current cloud-based systems. In comparison to the blockchain method, it can also better increase query efficiency and achieve type II backward privacy.

II. RELATED WORK

SSE is an encryption primitive that allows the server The encryption mechanism known as SSE enables the server to sift through encrypted data directly. Song et al. made the initial proposal for it [1]. The efficiency of the search process will decrease linearly with the size of the database because it must scan the collected data in order to produce results. Numerous academics have studied this topic to enhance the effectiveness of queries. However, the majority of early schemes [4] does not support dynamic updating processes.

Dynamic SE technology has been suggested as a response to the evolving needs of users [5]. As a result, the system is flexible and useful because a client can add or remove data whenever they want, but doing so will result in data leakage. In 2014, Stefanov et al.[6] introduced the idea of forward and backward privacies to address the covert issue of information leakage throughout the update process. Forward privacy ensures that newly added files are unrelated to earlier search processes. The "dynamic" generation of encrypted datasets has obvious advantages, but it's also essential for other abuse assaults, like those that involve hostile file insertion. Backward privacy ensures that information from deleted documents won't be discovered during later search activities. Later, backward privacy was formally defined by Bost[7]. They presented a forward- and backward-compatible SE method in 2017 and classified the backward privacy level into three categories (Type III, Type II, and Type I), ranging from low to high. Numerous strategies have been presented [8], [9] to investigate backward privacy further; some of these schemes are able to attain Type I backward privacy. Later, in 2018, Sun et al.[10] proposed a symmetric puncturable encryption technique that introduces a new type of encryption for the SE field while also preventing the server from searching for deleted documents containing specific keywords. Patranabis and Mukhopadhyay [11] presented a dynamic SE system that supports conjunctive keyword searches and satisfies forward and backward privacy in 2021. More researchers have also worked on a number of other topics, including query function [12], [13], and performance optimization [14]–[16], in order to adapt to more sophisticated functions.

Security designs against a malicious server have not received enough attention, and the majority of current solutions primarily concentrate on an honest yet inquisitive cloud server. The cloud server turns into a malevolent server when an external attack or internal setup error takes place. This might result in changes to the server or the revelation of encrypted data, as well as the possibility of incorrect query results. There have been numerous verification systems put forth in response to the aforementioned issues. A one-way function and a pseudorandom function were employed by Soleimanian and Khazaei[17] in 2019 to finish the verification of open outcomes. Tong et al. [18] integrated k-means clustering with the Merkle hash tree in 2020 to provide a system that enhances security and verification efficiency. In a semantic setting, Yang and Zhu[19] found a solution to the correlation verification problem. By converting the verification procedure into a linear programming assignment, Yang and Zhu[19] were able to solve the correlation verification problem in a semantic setting. When users are trustworthy, these schemes can produce accurate findings; but, when users are dishonest, it becomes more difficult to acquire accurate data.

In order to address the issue of inaccurate query results resulting from dishonest users, blockchain technology has been implemented into SE. It can successfully verify both the integrity of query results and the non-tampering of encrypted data. Tang[20]'s blockchain-based method can maintain the original SE scheme's anonymity while achieving the necessary fairness. Hu et al. [21] proposed a smart contract-based SSE system. User files can be kept in any public cloud storage system, and the index of such files is kept in a peer-to-peer network smart contract. Nonetheless, the assumption that the blockchain is sufficiently secure allows for security to be ensured. Later, Chen et al. [22] implemented the index structure modification to the electronic case sharing system, based on the Hu et al. scheme. The trial demonstrated its usefulness, although there was a significant overhead. Jiang et al. [23] created a stealth authorization mechanism to strengthen the security of the blockchain and a publicly verifiable search framework for outsourced encrypted data based on a blockchain. In order to handle index query results in a secure and workable manner, they outsourced the corresponding encrypted data to the cloud and uploaded the encrypted index to the blockchain. Nevertheless, encrypted data kept on the cloud server might be altered or leaked, much as the plan put up by Guo et al.[24] to validate query results using a blockchain. Data thus face possible safety risks, and query outcomes are unpredictable.

III. METHODOLOGIES

Background Blockchain

A distributed public database, or blockchain, is a system of linked data exchanged, processed, and stored by a number of participants using point-to-point network communication technology, distributed consensus protocol, modern cryptography, and smart contract programming language[25]. A novel kind of blockchain-based decentralized computing platform called Ethereum is utilized in the PPSE. By implementing smart contracts, Ethereum enables users to carry out any complicated operation as needed.

Transparency, public verification, immutability, and unforgeability are attributes of a blockchain. Decentralization, which enables it to support data verification, sharing, processing, storage, and other services through multilateral autonomous technical means like consensus, is its greatest advantage[26]. Companies are developing blockchain-based apps due to the benefits of blockchain technology. Blockchains are classified as public or private based on the permissions granted to users.

Public Blockchain

A consensus blockchain that is open to all users to read, send, and get legitimate confirmation from is known as a public blockchain. A public blockchain's workload or equity certification mechanism keeps it secure. These blockchains follow the fundamental premise that each participant's economic benefit from a blockchain is based on how much they contributed to the consensus process. They function by combining digital verification with encrypted incentives. Additionally, by offering incentives, more people may decide to sign up for the blockchain network and help popularize the idea of blockchain projects. Thus, an incentive system plays a critical role in project operations for a public blockchain. Most people classify these blockchains as "fully decentralized." There is a network effect in a public blockchain.

Private Blockchain

A blockchain that only has writing authority held by one organization is said to be fully private. Permissions for publishing and accessing data are quite stringent. These kinds of data are typically not available to the public. Essentially, the private blockchain can build a system with greater access control, and modification or even reading permissions can be limited to a small number of users, in contrast to a completely unregulated and open system that ensures network security through an encrypted economy. Simultaneously, this technique maintains the blockchain's legitimacy and partial decentralization. The private blockchain is particularly efficient for a variety of processes since it has few nodes involved in blockchain activities, minimal transaction costs, easily-modifiable rules, and limited read permissions.

Smart Contract

Ethereum smart contracts are programs whose states are recorded in the blockchain. They are able to assist, confirm, and uphold the contract's procedures[27]. A unique address designates each smart contract, which also includes script code, a monetary balance, and storage in the form of a key/value store. Even the creator of the contract cannot change its code after it has been developed and deployed to Ethereum indefinitely[18].

Index Structure

The PPSE employs a KC-IDC index structure (KC is the number of times the keyword appears in the document ID, and IDC is the number of documents containing the keyword) since the forward index search takes too long. To increase query efficiency, this index structure may rapidly retrieve the document list that contains the keyword. It is a particular type of storage form that implements the "Keyword: Document Matrix," mostly made up of KC-IDC files and a keyword dictionary. Using the KC-IDC index, we can get a list of documents that include this term. We utilize Fig. 1 as an example to make the discussion of its structure easier to understand. The KC-IDC index entry for the term "key" is displayed in the figure. The ID of the document that contains this keyword is represented as a 16-bit string, where "6" indicates that the keyword "key" appeared six times in this document.

IV. SYSTEM MODE

Model Introduction

Four entities make up the system model, which is depicted in Fig. 2: DO, DU, public blockchain, and private blockchain.

(1) Do

The primary duty for the DO is to encrypt documents and indexes. After encryption, documents are posted to the blockchain for public use. The access control request is posted to the private blockchain along with encrypted indexes.

(2) Private Blockchain

Access control requests and encrypted indexes are kept in this location. The private blockchain obtains the DU's access control information after receiving encrypted indexes and access control requests. The private blockchain first verifies and checks the matching DU, then it uses the search token given by the received DU to run a corresponding query in the private blockchain. Finally, it delivers the encrypted index that was searched to the DU.

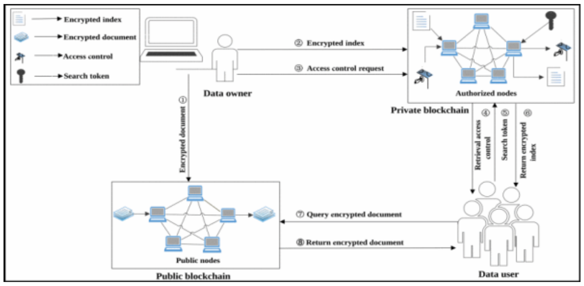


Fig 4. System model.

(3) Public Blockchain

This blockchain is used to store encrypted documents. After the public blockchain receives the request to retrieve the encrypted document sent by the DU, it performs the search in the public blockchain and returns the retrieved encrypted document corresponding to the encrypted index to the DU.

(4) Du

The DU validates and compares the access control information it receives from the private blockchain with its own token[28] before requesting more information. It is the matching DU if the two are equivalent. Next, in order to access the encrypted document sent by the public blockchain, the DU transmits its own search token to the private blockchain, queries the appropriate encrypted index, obtains the index results, and submits a request to retrieve the document to the public blockchain. If not, the search is carried out again.

With the qualities of decentralization, low cost, high timeliness, and high accuracy, smart contracts handle all search, add, and delete actions in the PPSE.

Table 1 Symbol Definitions.

Symbol	Description
DB	Document database
$PRF\{G, F\}$	Secure pseudorandom function
(u, s)	Keyword status
ptr_i, ptr_{i+1}	Index pointer and the next index pointer
t_w^1, t_w^2	Search token and key token
id_i, C_{id_i}	File index and encrypted index
L_R, α_w	Search list and random number
$Hash$	Hash function
H	Hash function used to generate token
$Token$	Access control token
RID	Request information ID
SID	Data sender ID
$REID$	Data recipient ID
w	Keyword
Map	State mapping
EDB	Encrypted index database
ST, N, Y	Search status. ST: Did nothing; N: Not searched; Y: Searched
U_i, U_j	Data owner and data user
P_α, V_α	Random number's encrypted index and the corresponding pointer
c	Ciphertext of information RI
PK	Hash function key

V. OUR PROPOSED MODEL

Privacy-Preserving Searchable Encryption (PPSE):

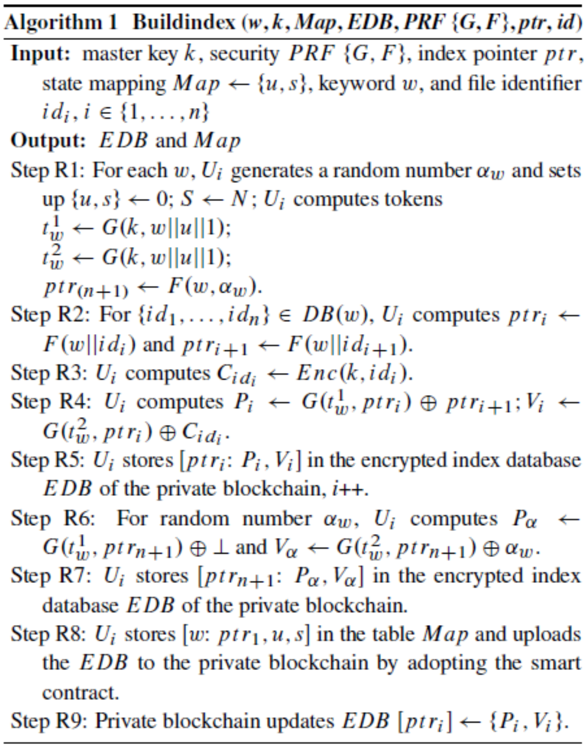
This paper proposes a Privacy-Preserving Searchable Encryption (PPSE) scheme based on public and private blockchains.

Public Blockchain:

A consensus blockchain that is open to all users to read, send, and get legitimate confirmation from is known as a public blockchain. A public blockchain's workload or equity certification mechanism keeps it secure. These blockchains follow the fundamental premise that each participant's economic benefit from a blockchain is based on how much they contributed to the consensus process. They function by combining digital verification with encrypted incentives. Additionally, by offering incentives, more people may decide to sign up for the blockchain network and help popularize the idea of blockchain projects. Thus, an incentive system plays a critical role in project operations for a public blockchain. Most people classify these blockchains as "fully decentralized." There is a network effect in a public blockchain. The characteristics of a public blockchain include neutrality, openness, decentralization, network effect, strong resistance to censorship, irreversibility, and irrevocability.

Private Blockchain:

A blockchain that only has writing authority held by one organization is said to be fully private. Permissions for publishing and accessing data are quite stringent. These kinds of data are typically not available to the public. Essentially, the private blockchain can build a system with greater access control, and modification or even reading permissions can be limited to a small number of users, in contrast to a completely unregulated and open system that ensures network security through an encrypted economy. In addition, this solution keeps some of a blockchain's legitimacy and decentralization. The private blockchain is particularly efficient for a variety of processes since it has few nodes involved in blockchain activities, minimal transaction costs, easily-modifiable rules, and limited read permissions.



```
Algorithm 1  Buildindex $(w, k, Map, EDB, PRF\{G, F\}, ptr, id)$
Input: master key $k$, security $PRF\ \{G, F\}$, index pointer $ptr$,
    state mapping $Map \leftarrow \{u, s\}$, keyword $w$, and file identifier
    $id_i, i \in \{1, \ldots, n\}$
Output: $EDB$ and $Map$
Step R1: For each $w$, $U_i$ generates a random number $\alpha_w$ and sets
    up $\{u, s\} \leftarrow 0$; $S \leftarrow N$; $U_i$ computes tokens
    $t_w^1 \leftarrow G(k, w||u||1)$;
    $t_w^2 \leftarrow G(k, w||u||1)$;
    $ptr_{(n+1)} \leftarrow F(w, \alpha_w)$.
Step R2: For $\{id_1, \ldots, id_n\} \in DB(w)$, $U_i$ computes $ptr_i \leftarrow$
    $F(w||id_i)$ and $ptr_{i+1} \leftarrow F(w||id_{i+1})$.
Step R3: $U_i$ computes $C_{id_i} \leftarrow Enc(k, id_i)$.
Step R4: $U_i$ computes $P_i \leftarrow G(t_w^1, ptr_i) \oplus ptr_{i+1}; V_i \leftarrow$
    $G(t_w^2, ptr_i) \oplus C_{id_i}$.
Step R5: $U_i$ stores $[ptr_i\colon P_i, V_i]$ in the encrypted index database
    $EDB$ of the private blockchain, $i$++.
Step R6: For random number $\alpha_w$, $U_i$ computes $P_\alpha \leftarrow$
    $G(t_w^1, ptr_{n+1}) \oplus \perp$ and $V_\alpha \leftarrow G(t_w^2, ptr_{n+1}) \oplus \alpha_w$.
Step R7: $U_i$ stores $[ptr_{n+1}\colon P_\alpha, V_\alpha]$ in the encrypted index
    database $EDB$ of the private blockchain.
Step R8: $U_i$ stores $[w\colon ptr_1, u, s]$ in the table $Map$ and uploads
    the $EDB$ to the private blockchain by adopting the smart
    contract.
Step R9: Private blockchain updates $EDB\ [ptr_i] \leftarrow \{P_i, V_i\}$.
```

Four Basic Algorithms

Setup (1^λ):

The DO enters the parameter λ, initializes the system locally, and outputs the master key k←{0,1}λ. After initializing the empty sets Map and EDB, Map is utilized to record the updating and search status. The KC-IDC index structure, which is used by the PPSE technique, is kept in the encrypted index set EDB as (w, id) pairs, where w and id stand for the file identifier and keyword, respectively.

The file identifier (id) and keyword (w) are entered, the update and search times are set to zero, and the search status is changed from S T to N, indicating that it hasn't been searched, as demonstrated in Algorithm 1. For every the search token is used to create a guide, or ptri, for each index after the search token and key token, t1w and t2w, are formed. The current encrypted index (Pi) and its corresponding pointer (Vi) are obtained by performing the XOR (exclusive OR) operation when the first pointer is initialized, assuming that there was no previous pointer and that it was set to null. These are then stored as pieces of data in the encrypted index database EDB and updated. Map contains the state that corresponds to the keyword. Ultimately, the smart contract is adopted and the encrypted index EDB is published to the private blockchain. Local Map changes are made by the DO.

Search (k, Map, G, w):

The DU receives the access control information from the private blockchain, as seen in Algorithm 2, and delivers a search token to the private blockchain in return. User Uj is identified as an authorized user following computation. The search is then carried out by calling the smart contract on the private blockchain. To get the keyword's status, an empty list LR must first be established. After judging ST, the DU creates a new token for the search term. In the event that it equals Y, the keyword has not been updated following the search; if not, the most recent index pointer, ptri, must be computed as the updated index of this keyword has not yet been searched.

After Uj receives the access control data, he sends the search token to the private blockchain. Uj is found to be an authorized user after computation. When the index pointer for keyword w corresponds to a null value, the private blockchain instructs the smart contract to search for the encrypted index associated with it. At that moment, an empty result list (LR) is formed.fter that, the matching DU Uj receives all of the ciphertexts and random number results discovered in the query, which are LR={Cidi,...,Cidn,αw}.

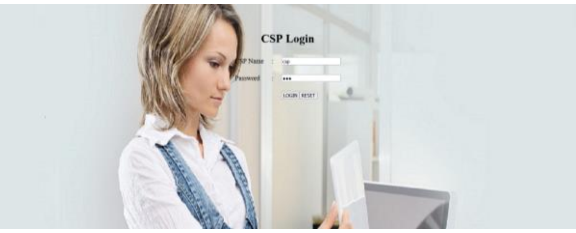
Addition (k, G, Hash, map):

The keyword w's update status is obtained by the DO, as seen in Algorithm 3. The first update following the keyword search is represented as S equal to Y, indicating that s searches still need to be updated. A new token and index pointer are created using the new key. S is finally set to N. If not, the prior key k and token can be used because the key hasn't been checked since the change. Once the pointer is linked to the preceding pointer, the new index is determined. Following the encryption of the data, the hash function operation is carried out. The hash value that is produced is then utilized as an input to invoke the smart contract, which simultaneously uploads the matching encryption to the public blockchain and adds the encrypted data to the private blockchain. Next, Map is updated by the DO.

Delete (b1lo Ckno, db(w), edb):

As demonstrated in Algorithm 4, the DO calls the deleted smart contract and performs the appropriate action based on the input block number by using the block number blockNo as an input. The DO also removes the encrypted documents from the public blockchain during the data destruction process.

VI. RESULT



CSP Name : csp
Password : ***

Fig 6.1 CSP Login Page

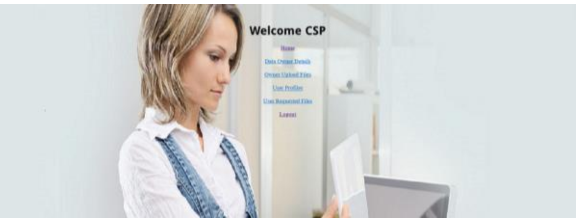


Fig 6.2 CSP Home Page

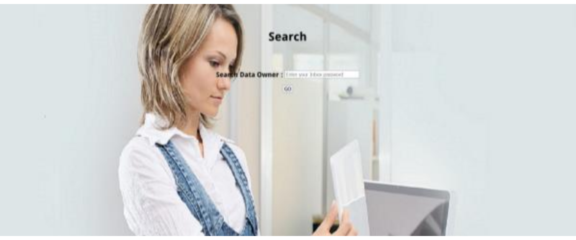


Fig 6.3 Search Page

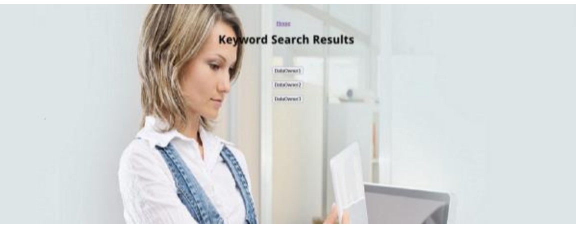
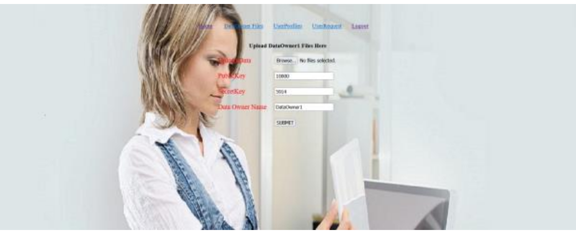


Fig 6.4 Search Results Page



No files selected.
PublicKey : 10800
SecretKey : 3914
Data Owner Name : DataOwner1

Fig 6.5 Upload Page

VII. CONCLUSION

Customary privacy-protecting Cloud servers are necessary for SE schemes to execute search operations. This study uses a decentralized security paradigm based on blockchain technology to address issues with harmful users' attacks and original malicious servers. In contrast to other verification techniques now in use, the PPSE does not demand verification from the DO or ask them to forward the results to a third party for verification. When data is stored on the blockchain itself, accurate and unchangeable outcomes can be achieved. To increase query performance and the security of the encrypted data, we implement an access control mechanism and keep the encrypted index in the private blockchain while outsourcing the relevant encrypted documents to the public blockchain. The safety analysis shows that the plan satisfies safety standards, and the outcomes of our prototype's experiments show that the plan is workable.

Future Work: The safety analysis shows that the plan satisfies safety standards, and the outcomes of our prototype's experiments show that the plan is workable.

REFERENCES

1. D. X. Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data", Proc. 2000 IEEE Symp. on Security and Privacy. S&P 2000, pp. 44-55, 2000.
2. Z. T. Guan, N. Y. Wang, X. F. Fan, X. Y. Liu, L. F. Wu and S. H. Wan, "Achieving secure search over encrypted data for e-commerce: A blockchain approach", ACM Trans. Int. Technol., vol. 21, no. 1, pp. 12, 2021.
3. H. Y. Li, T. Wang, Z. R. Qiao, B. Yang, Y. Y. Gong, J. Y. Wang, et al., "Blockchain-based searchable encryption with efficient result verification and fair payment", J. Int. Secure. Appl., vol. 58, pp. 102791, 2021.
4. Z. L. Liu, T. Li, P. Li, C. F. Jin and J. Li, "Verifiable searchable encryption with aggregate keys for data sharing system", Future Gener. Comput. Syst., vol. 78, pp. 778-788, 2018.
5. S. Kamara, C. Papamanthou and T. Roeder, "Dynamic searchable symmetric encryption", Proc. 2012 ACM Conf. on Computer and Communications Security, pp. 965-976, 2012.
6. E. Stefanov, C. Papamanthou and E. Shi, "Practical dynamic searchable encryption with small leakage", Proc. of Network and Distributed System Security Symposium, pp. 72-75, 2014.
7. R. Bost, "∑οφοξ: Forward secure searchable encryption", Proc. 2016 ACM SIGSAC Conf. on Computer and Communications Security, pp. 1143-1154, 2016.

8. R. Bost, B. Minaud and O. Ohrimenko, "Forward and backward private searchable encryption from constrained cryptographic primitives", Proc. 2017 ACM SIGSAC Conf. on Computer and Communications Security, pp. 1465-1482, 2017.

9. J. G. Chamani, D. Papadopoulos, C. Papamanthou and R. Jalili, "New constructions for forward and backward private symmetric searchable encryption", Proc. 2018 ACM SIGSAC Conf. on Computer and Communications Security, pp. 1038-1055, 2018.

10. S. F. Sun, X. L. Yuan, J. K. Liu, R. Steinfeld, A. Sakzad, V. Vo, et al., "Practical backward-secure searchable encryption from symmetric puncturable encryption", Proc. 2018 ACM SIGSAC Conf. on Computer and Communications Security, pp. 763-780, 2018.

11. S. Patranabis and D. Mukhopadhyay, "Forward and backward private conjunctive searchable symmetric encryption", Proc. of 28th Annu. Network and Distributed System Security Symp.

12. J. Li, Y. Y. Huang, Y. Wei, S. Y. Lv, Z. L. Liu, C. Y. Dong, et al., "Searchable symmetric encryption with forward search privacy", IEEE Trans. Depend. Secure Comput., vol. 18, no. 1, pp. 460-474, 2021.

13. X. Q. Liu, G. M. Yang, W. Susilo, J. Tonien, X. M. Liu and J. Shen, "Privacy-preserving multi-keyword searchable encryption for distributed systems", IEEE Trans. Parallel Distrib. Syst., vol. 32, no. 3, pp. 561-574, 2021.

14. G. Asharov, G. Segev and I. Shahaf, "Tight tradeoffs in searchable symmetric encryption", I. Cryptol., vol. 34, no. 2, pp. 9, 2021.

15. K. He, J. Chen, Q. X. Zhou, R. Y. Du and Y. Xiang, "Secure dynamic searchable symmetric encryption with constant client storage cost", IEEE Trans. Inf. Forensics Secur., vol. 16, pp. 1538-1549, 2021.

16. Q. Y. Song, Z. T. Liu, J. H. Cao, K. Sun, Q. Li and C. Wang, "SAP-SSE: Protecting search patterns and access patterns in searchable symmetric encryption", IEEE Trans. Inf. Forensics Secur., vol. 16, pp. 1795-1809, 2021.

17. A. Soleimanian and S. Khazaei, "Publicly verifiable searchable symmetric encryption based

18. Q. Y. Tong, Y. B. Miao, X. M. Liu, K. K. R. Choo, R. Deng and H. W. Li, "VPSL: Verifiable privacy-preserving data search for cloud-assisted internet of things", IEEE Trans. Cloud Comput.

19. W. Y. Yang and Y. S. Zhu, "A verifiable semantic searching scheme by optimal matching over encrypted data in public cloud", IEEE Trans. Inf. Forensics Secure., vol. 16, pp. 100-115, 2021.

20. Q. Tang, J. Zhou, X. Luo, Q. Shen and Z. Xu, "Towards blockchain-enabled searchable encryption" in Information and Communications Security, Cham, Switzerland: Springer, pp. 482-500, 2020.

21. S. S. Hu, C. J. Cai, Q. Wang, C. Wang, X. Y. Luo and K. Ren, "Searching an encrypted cloud meets blockchain: A decentralized reliable and fair realization", Proc. of IEEE INFOCOM 2018-IEEE Conf. on Computer Communications, pp. 792-800, 2018.

22. L. X. Chen, W. K. Lee, C. C. Chang, K. K. Choo and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing", Future Gener. Comput. Syst., vol. 95, pp. 420-429, 2019.

23. S. R. Jiang, J. Q. Liu, L. M. Wang and S. M. Yoo, "Verifiable search meets blockchain: A privacy-preserving framework for outsourced encrypted data", Proc. of 2019 IEEE Int. Conf. on Communications (ICC), pp. 1-6, 2019.

24. Y. Guo, C. Zhang and X. H. Jia, "Verifiable and forward-secure encrypted search using blockchain techniques", Proc. of 2020 IEEE Int. Conf. on Communications (ICC), pp. 1-7, 2020.

25. P. L. Li, H. X. Xu, T. J. Ma and Y. H. Mu, "Research on fault-correcting blockchain technology (in Chinese)", J. Cryptol. Res., vol. 5, no. 5, pp. 501-509, 2018.

26. X. Han, Y. Yuan and F. Y. Wang, "Security problems on blockchain: the state of the art and future trends (in Chinese)", Acta Autom. Sin., vol. 45, no. 1, pp. 206-225, 2019.

27. J. L. Sun, S. Huang, C. Y. Zheng, T. Y. Wang, C. Zong and Z. W. Hui, "Mutation testing for integer overflow in Ethereum smart contracts", Tsinghua Science and Technology, vol. 27, no. 1, pp. 27-40, 2022.

28. R. Z. Du, A. L. Tan and J. F. Feng, "An attribute-based encryption scheme based on unrecognizable trapdoors", Tsinghua Science and Technology, vol. 25, no. 5, pp. 579-588, 2020.