Blockchain-Enabled Searchable Encryption for Secure and Efficient Sharing of IoHT-Generated Electronic Medical Records in Cloud-Based Healthcare

Suyamburaj RAJESH KUMAR [1], Velusamy GOMATHI [2], Kanagamani VIVEKRABINSON [3]

[1] Dept. of CSE, KPR Institute of Engineering and Technology, Coimbatore, 641407, India
[2] Dept. of CSE, National Engineering College, Kovilpatti, 628503, India
[3] Dept. of CSE, Kalasalingam Academy of Research and Education, Krishnankoil, 626126, India

suyamburajesh@gmail.com, vgcse@nec.edu.in, vivekrabinson1993@gmail.com

Submitted March 19, 2025 / Accepted April 8, 2025 / Online first May 13, 2025

Abstract. *Protecting the security of data generated by wearables and monitoring devices is critical in smart wards, especially when healthcare schemes use cloud storage services to save patients' Electronic Medical Records (EMRs). These devices operate in wireless communication environments, where data integrity and transmission security are vital. Despite the fact that encryption helps protect information, it often reduces the benefits of sharing the information generated using Internet of Health Things (IoHT) devices with others. As individuals increasingly share their EMRs with third parties, developing an effective searchable encryption framework for sharable EMRs remains a crucial task. Furthermore, cloud-based access control might result in heavily centralized control. To address this, we proposed a blockchain-assisted technique for sharable EMRs that incorporates a searchable encryption scheme compatible with a resource-constrained wireless system that does not require any secure channel. The encrypted EMRs are saved in the cloud, while the encoded keyword indexes are kept on the blockchain, assuring tamper resistance, integrity, and accountability of the encrypted indexes. Our technique also enables exact recovery of encrypted EMRs using a multi-keyword search, removing the necessity for third-party verification. Compared to prior searchable encryption systems, our technique reduces storage costs while increasing computational efficiency. Furthermore, our system is immune to keyword-guessing attacks, a must-needed one that many previous solutions fail to address wireless medical data security.*

Keywords

Blockchain, Electronic Medical Records (EMR), Internet of Health Things (IoHT), Keyword-Guessing Attacks (KGA), multi-keyword search, Searchable Encryption (SE)

1. Introduction

Smart health focuses not just on delivering treatment to sick persons irrespective of the place and time but also on maintaining secrecy and controlling information exchange [1]. The Internet of Health Things attempts to attain these goals through 5G communication technologies, pervasive connections, and smart IoT devices. In a smart hospital/clinic, wearables and tracking equipment create Electronic Medical Records (EMRs), which can be maintained locally for real-time health tracking and diagnosis. However, depending only on hospital-based servers for EMR storage places a major strain on server resources and limits data exchange [2]. To overcome this issue, several clinics have started to use cloud storage servers, which enable healthcare recipients to upload their medical records to the cloud [3]. This strategy reduces the storage requirements of hospital servers while also facilitating data exchange. Nonetheless, Cloud Storage Providers (CSPs) are not always trustworthy, and some may exploit clients EMRs for financial gain, jeopardizing the patient's privacy. To protect patient privacy, EMRs are encrypted before they are sent to the cloud. Encryption provides data secrecy and access control, but it also hinders data exchange [4].

However, the data encoding mechanism makes it hard for doctors to search for certain phrases in medical records. The easiest option is for doctors to download all of the encrypted information, decode it independently, and perform the search over it. However, this is impracticable because of the extremely high processing and transmission overhead especially in resource-constrained wireless environments. To overcome this issue, researchers introduced a technology named Searchable Encryption (SE) [5], [6]. SE is an encoding approach that enables doctors to do keyword searches on encoded content. In practice, we often encounter situations where multiple data owners are involved, such as an electronic medical record of a patient

DOI: 10.13164/re.2025.0289

that needs to be jointly accomplished by various doctors, departments, or healthcare institutions [7]. Existing Public Key encryption with Keyword Search (PEKS) techniques need the EMR holder to encode each term for every recipient. If many doctors are authorized to search the client's EMR using multiple queries, this might result in substantial computational and storage costs [8]. For instance, if a hundred authorized doctors search the patient's EMR for a hundred keywords, the patient must create 10,000 ciphertexts matching these keywords. Furthermore, the patient must delegate the encrypted information to the cloud storage server, raising communication costs. Thus, having an effective search technique for exchanging encrypted information is critical. Although several investigators have suggested multi-user searchable encryption [9], [10], these techniques often need numerous identities, which adds to the computational load for data owners.

The present Cloud-assisted Internet of Medical Things (CIoMT) search encounters the difficulty of incredibly centralized cloud server power, as it manages access control, data storage, searching, and monitoring all at once, potentially affecting patient privacy. Blockchain integration with the Internet of Things or IoMT systems has received a lot of consideration [11] because of its benefits, including distribution, accountability, and non-tampering. As a decentralized database, blockchain delivers a novel mode to store and transmit information, improving transaction transparency, fairness, and security [12], [13]. Precisely, blockchain technology assures data originality and allows for equitable data exchange. Its anti-tampering features ensure data confidentiality and authenticity, allowing users to obtain comprehensive and accurate search outcomes without extra validation. Furthermore, blockchain can track evidence about data validity and detect suspicious server behavior [14], [15].

To overcome these issues, we presented the Secure Channel free Blockchain-Assisted Public Key Encryption with Keyword Search (SCBA-PEKS) method. This technique decreases the computational and storage requirements for sharable medical reports in a multi-user environment. Our technique needs the data owner/patients to produce only one encoded text regardless of the set of doctors in the scheme. The main goals of our proposed technique are listed here:

- Our suggested SCBA-PEKS approach simplifies EMR sharing in a multi-receiver scenario, allowing patients to share their EMRs with various providers while lowering computational and storage overhead. The patient just has to create one index file for their EMR, regardless of the number of recipients.
- We use a permissioned blockchain to enable safe, one-to-many health record exchange. The information uploading procedure and keyword index are all logged on the blockchain to track and validate the integrity of the EMR.
- Next, the blockchain employs the Pedersen scheme for doctors (data consumers) to generate trapdoors, hence improving security and resilience.
- The suggested technique assures that both indexes and trapdoors are indistinguishable. Its cryptographic qualities protect against keyword-guessing attacks, a notable concern in wireless medical data security.

The remaining sections of the paper are organized as follows: Section 2 describes the state of art research works published and their shortcomings. The proposed SCBA-PEKS system model and the security model are presented in Sec. 3. Section 4 showcases the implementation of our proposed work. The security analysis of our proposed approach is introduced in Sec. 5, and the performance analysis is demonstrated in Sec. 6. Finally, Section 7 concludes our paper.

2. Related Work

The searchable encryption [6] technique is majorly divided into two categories: Symmetric Searchable Encryption (SSE) and Public Key Searchable Encryption (PEKS). While SSE methods have a minimal computational cost, they frequently have key exchange difficulties. In 2004, Dan et al. [16] presented a PEKS as a solution for recovering the encoded text in public key cryptosystems using the keyword search. Following that, various upgraded PEKS schemes were created, enabling encoded indexes to be compared with keyword-related trapdoors to offer the capabilities to search while maintaining keyword anonymity. For example, Yang et al. [17] introduced a PEKS technique that is immune to Keyword Guessing Attacks (KGA) and showed its secrecy, excluding the random oracle model. Qiong Huang et al. [18] presented a certified encoding approach using public keys with query search to address the KGA employed by insiders. However, Baodong Qin et al. [19] showed that Qiong Huang's technique is susceptible to selected multi-ciphertext attacks from outside adversaries. To solve these flaws, Baodong Qin et al. introduced a novel authenticated encoding technique with keyword-based searches based on the public key that protects the system against selected multi-ciphertext assaults from external adversaries and keyword-guessing assaults from internal adversaries.

To avoid keyword estimating attacks by a hostile server, the researchers in [20] utilized the information owner's private key, ensuring that only the information holder may encode the keywords, preventing adversaries from initiating such attacks. This technique is also resistant to selected plaintext assaults. In [21], certificate-based searchable encoding was used to protect against the attacks on keyword guessing, with characteristics like implicit authentication, no requirement for a secure connection, and no key escrow. To overcome keyword guessing attacks, [22] employed an authentication-based encryption technique similar to [19], which employs a lightweight scheme that eliminates costly bilinear pairing procedures. Recent

upgrades to PEKS schemes have focused on both functionality and security. Pan et al. [23] suggested a PEKS technique that achieves non-differentiability across multiple ciphertexts and trapdoors. Zhang et al. [24] suggested a post-quantum secure PEKS for the industrial Internet of Things (IoT) that also protects against keyword-guessing attacks employed by inside adversaries. Cheng-Yi Lee et al. [25] suggested a searchable encryption technique for IoT that supports bidirectional search (both owner and user can perform search). Given the use and importance of PEKS schemes that enable multiple keywords, Peiming Xu et al. [26] developed a searchable encryption technique to perform a boolean search over encoded e-mails using hidden structures that support multiple keywords. In Xueqiao Liu et al.'s [27] multi-keyword public key approach, several authorized servers were employed to speed up search responses and limit the possibility of key leaking.

Blockchain's immutability and distributive nature have aroused great interest in its use in e-health security and privacy. Shamshad et al. [28] created an E-Medical information-sharing approach employing consortium and private blockchains to save the keyword index and encoded text. Their method combines several approaches, including PEKS, to enable effective and protected information exchange. Leyou Zhang et al. [29] suggested a blockchain-enabled health data-sharing strategy that supports fair keyword searches and uses cryptographic algorithms to identify rogue blockchain nodes. They also adopt both off-chain and on-chain storage techniques to compensate for the blockchain's restricted space constraints. Zhang et al. [30] employed blockchain to prevent online KGA by recording each keyword request raised by a client as a transaction on a public blockchain. Jiang et al. [31] use the PEKS approach with blockchain to improve search consistency in cloud-assisted IoT systems. Yang et al. [32] presented a blockchain-enabled PEKS technique to guard against fraudulent users, which logs search queries for traceability. Furthermore, the authors in [33] presented a health information-sharing architecture utilizing ciphertext-based attribute encryption and permissioned blockchains to assure data privacy and permission management.

To enable safe and effective data exchange for cloud-stored IoT data, Yu et al.'s system [34] combines blockchain with PEKS and Attribute-Based Encryption (ABE). Fanfan Shen et al. [35] created a blockchain-assisted searchable system for exchanging electronic medical information that uses dual permission, with the blockchain assuring fair communications and checking the encoded text. To address search geniuses, Qing Wu et al. [36] suggested a multi-authorization keyword search scheme based on attributes in a multi-cloud environment using consortium blockchain. He et al. [37] created a decentralized application for a charitable donation in which the donors can perform a multi-keyword search for the beneficiaries they wish to help. Even though the blockchain-integrated PEKS schemes have improved security and versatility in searching, they are mostly concerned with protecting against keyword guessing attacks, enforcing fair keyword searches, or adding attribute-based encryption for access control. However, none of them mends the original cryptographic primitive to allow simple one-to-many transmission, which is highly needed for scalability. While existing works enhance security by adding a blockchain logging of search queries, preventing insider attacks, and adding access control mechanisms, they do not address the overhead of computation or efficiency issues while dealing with large-scale encrypted searches.

3. System and Security Model

3.1 System Model

Encryption and search are critical components of any data-sharing system in the wireless communication environment. Our suggested SCBA-PEKS model contains four major actors, as indicated in Fig. 1:

Data Owner (DO): Before outsourcing an EMR to authorized data users, the data owner (patient) must encrypt it to safeguard privacy. To allow authorized data users to search the EMR efficiently, the owner creates a secure index containing relevant keywords. Then, the DO sends the encoded EMR and associated index to a server via the wireless communication channel. To offer search capabilities to authorized users, the owner generates recovery keys for each user and delivers them to the cloud storage provider. The quantity of keys generated for the EMR is determined by the number of authorized users allowed to search.

Data Users (DU): Authorized users, such as doctors, can do keyword searches on the encrypted EMR. Each user creates a query phrase set and submits it to the blockchain for trapdoor creation. A search is successful when the query phrase matches the index, letting the user receive the encoded EMR from the cloud storage. This process leverages signal processing techniques to ensure efficient and accurate retrieval of medical data over wireless networks.

Blockchain: The blockchain has three types of nodes. The first kind (peers in Hyperledger Fabric) accepts requests from the DO, uses the smart contract to track the procedure of transferring the encoded EMR to the storage server, and puts the encoded keyword index value in the ledger to perform a trapdoor search. The second category comprises consensus nodes (Hyperledger Fabric: endorsers and orderers). The third category nodes are responsible for employing the Pedersen Protocol to produce the trapdoors for DU to perform the search.

The cloud server (CS): The role of the storage service provider is to retain the owner's encoded EMR and deliver it to the authorized doctor/user after getting the valid search results from the blockchain. This ensures that sensitive medical data is securely stored and transmitted over wireless communication channels, maintaining data integrity and confidentiality.

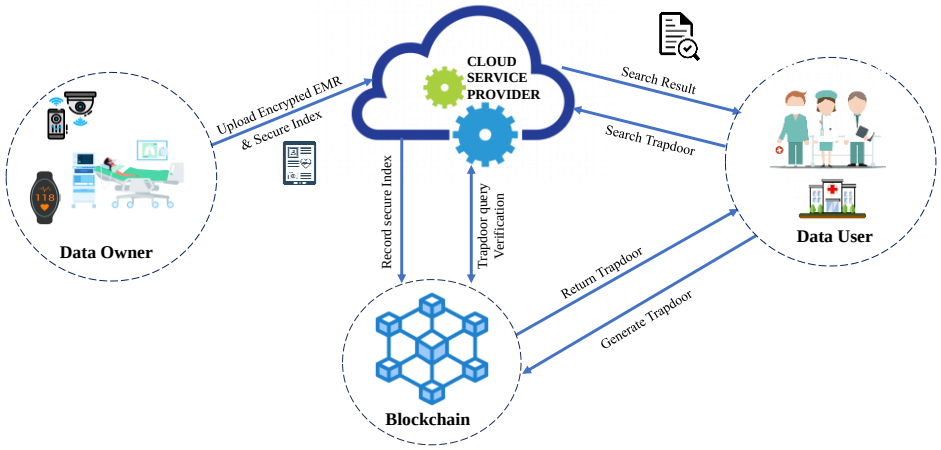

Fig. 1. System design.

3.2 Proposed System Overview

The key procedures in our suggested SCBA-PEKS scheme are separated into the following stages:

(1) **Setup(λ) → SPP:** Our approach creates a blockchain with a set of nodes, unlike existing PEKS systems that rely on a trusted center for setup. The ledger's initial block includes the security variable λ and the scheme's public parameter *SPP*.

(2) **Keygen(SPP) → (SK, PK)**: This method is used by the blockchain to produce the public search key (PK_{BC}). To generate this PK_{BC}, the Pedersen protocol selects a set of blockchain nodes (explained in Sec. 3.2) on a threshold basis. The storage server, DO, and DU also utilize the key creation procedure to compute their own private and public keys: (SK_C, PK_C), (SK_O, PK_O) and (SK_U, PK_U).

(3) **Encrypt & Index Gen(SPP, M, SK_O, PK_{BC}, PK_{U_i}) → (CT, I, RK):** Given a public parameter SPP, a medical data M and the private key of the sender SK_O, this method encrypts the record to compute the ciphertext CT. The patient/data owner then creates the recovery key RK for the EMR, guaranteeing that only the intended receiver may do keyword searches and decrypt the EMR. The DO additionally chooses a set of keywords *w*, creates the Index I for the EMR, and transfers it to the cloud server for storage.

(4) **TrapdoorGen(PK_{BC}, PK_C, W, SK_{U_i}) → (T_1, T_2, T_3):** For a set of keywords w_i, the Pedersen node uses the Pedersen Protocol to construct the parameter W_i^{β} in collaboration with other Pedersen nodes. The Pedersen node then demands the contract of trapdoor generation to compute the whole trapdoor T_1, T_2, and T_3 for the data user. The blockchain uses this trapdoor to search the term indexes contained on the blockchain.

(5) **Search(T_1, T_2, T_3, I, RK) → True/False:** Using the produced trapdoor, the blockchain performs a search operation using the trapdoor, file index, and recovery key to determine if the encrypted record includes the search phrase. After the successful completion of the search process, the blockchain notifies the server to provide the appropriate ciphertext CT to the data user.

(6) **Decrypt(CT, SKU, RK) → M:** With the ciphertext CT, secret key of user SKU, and recovery key RK, this algorithm decrypts the record and returns the original medical record if all input parameters are true.

3.3 Security Model

In this subsection, we will discuss the two major cryptographic properties and how our system is resilient against them.

3.3.1 Index Non-Differentiability

Our technique considers two distinct forms of Probabilistic Polynomial Time (PPT) attackers. The first, $\mathcal{A}_1$, indicates a malevolent cloud server that cannot differentiate between two encrypted challenge keywords (query). This applies if $\mathcal{A}_1$ may request a trapdoor oracle for any query other than the two challenge queries. The second, $\mathcal{A}_2$, indicates a malevolent outside adversary (including the blockchain) who may create a trapdoor for any query using the Pedersen scheme but lacks the cloud server's private key.

Indistinguishability under Index Non-Differentiability (IND-IX) is demonstrated using the below two security scenarios (games) involving adversaries $\mathcal{A}_1$/ $\mathcal{A}_2$ and a challenger $\mathfrak{B}$.

Game 1. A group of blockchain peers is selected from the network, and they serve as the challenger $\mathfrak{B}$.

(1) Initialization: The scheme's public information is recorded in the ledger. $\mathcal{A}_1$ produces a pair of keys (PK_C, SK_C) and broadcasts the public key PK_C to the ledger. $\mathfrak{B}$ generates its public key PK_{BC} via Pedersen Protocol and delivers it to $\mathcal{A}_1$.

(2) Query Phase I: $\mathcal{A}_1$ queries the Index oracle and Trapdoor with a polynomial quantity of keywords to yield appropriate responses.

(3) Challenge: $\mathcal{A}_1$ transmits two challenge keywords w_0 and w_1 to $\mathfrak{B}$, ensuring that these keywords have not ever been previously inquired in the Index or Trapdoor oracle. Then, the challenger $\mathfrak{B}$ picks a value q randomly from the set {0, 1}. Finally, the challenger $\mathfrak{B}$ computes the index IN^* using the IndexGen method and returns the IN^* to $\mathcal{A}_1$.

(4) Query Phase II: During this phase, $\mathcal{A}_1$ can query the Index oracles and Trapdoor adaptively, with the exception of the two keywords that are challenged.

(5) Guess: $\mathcal{A}_1$ generates its prediction q'. And we can say that $\mathcal{A}_1$ succeeds the Game 1 if and only if $q' = q$.

Game 2. Here, the cloud service provider acts as a challenger $\mathfrak{B}$.

(1) Setup: $\mathcal{A}_2$, a third-party attacker or rogue blockchain can access the public search key PK_S without the knowledge of the respective secret key. $\mathfrak{B}$ computes its keys (PK_C, SK_C) and transmits PK_C to $\mathcal{A}_2$.

(2) Query Stage I: $\mathcal{A}_2$ queries the index oracle with a polynomial quantity of keywords to obtain relevant outcomes.

(3) Challenge: $\mathcal{A}_2$ communicates two challenge keywords, w_0 and w_1 to $\mathfrak{B}$ if these keywords have yet to be previously inquired in the Index oracle.

(4) Query Stage II: $\mathcal{A}_2$ may still adaptively question any keyword to the Index oracle, with the exception of the keywords w_0 and w_1 that are challenged.

(5) Guess: Similar to Game 1.

The benefit of $\mathcal{A}_1$/ $\mathcal{A}_2$ in overcoming the index non-differentiability of a proposed SCBA-PEKS approach is explained as

$$Ad_{\mathcal{A}_1/\mathcal{A}_2}^{\text{IND\_IX}}(\lambda) = \left|\Pr[q=q'] - \frac{1}{2}\right|. \quad (1)$$

Definition 1. The suggested approach is index non-differentiable against Chosen Keyword Attacks for any polynomial-time attackers $\mathcal{A}_1$/$\mathcal{A}_2$, and the benefit $Ad_{\mathcal{A}_1/\mathcal{A}_2}^{IND\_IX}(\lambda)$ is insignificant.

3.3.2 Trapdoor Non-Differentiability

$\mathcal{A}_3$ is a malevolent external adversary who excludes the blockchain and the cloud storage server. Trapdoor non-differentiability is characterized as the succeeding security game among an attacker $\mathcal{A}_3$ and a challenger $\mathfrak{B}$.

Game 3. In this game, the challenger $\mathfrak{B}$ is a group of blockchain nodes.

(1) Setup: Initially, the challenger $\mathfrak{B}$ performs both the Setup(λ) process and KeyGen(PP) algorithms. The produced parameters SPP and PK_R are delivered to $\mathcal{A}_3$.

(2) Query 1: $\mathcal{A}_3$ chooses any keyword from the keyword space and asks the challenger to generate the trapdoor for that keyword. $\mathfrak{B}$ then answers with the trapdoor T to $\mathcal{A}_3$.

(3) Challenge: After Query 1, $\mathcal{A}_3$ generates two query keywords (w_0, w_1), and transmits them to $\mathfrak{B}$. Query 1 does not support queries for w_0 or w_1. After getting the keywords, the challenger picks a random number $q \in \{0, 1\}$, builds a challenge Trapdoor T, and transmits it to $\mathcal{A}_3$.

(4) Query 2: $\mathcal{A}_3$ maintains the same number of trapdoor requests as in Phase 1, subject to the limitation that w_0 and w_1 cannot be queried.

(5) Guess: $\mathcal{A}_3$ results the guess q', and succeeds the Game 3 if $q' = q$.

The following describes $\mathcal{A}_3$'s gain in succeeding the Indistinguishability under Trapdoor Non-Differentiability (IND-TD) game:

$$Ad_{\mathcal{A}_3}^{\text{IND\_TD}}(\lambda) = \left|\Pr[q=q'] - \frac{1}{2}\right|. \quad (2)$$

Definition 2. If the benefit $Ad_{\mathcal{A}_3}^{IND\_TD}(\lambda)$ of any PPT attacker, $\mathcal{A}_3$ is negligible; then the proposed technique is trapdoor-indistinguishable from keyword guessing attack.

4. Scheme Construction

We created a blockchain-enabled searchable encryption scheme to make it easier for patients to exchange their medical-related information generated by wearable IoT devices with various hospitals/doctors without requiring one-on-one contacts and linear encoding costs. This approach enables blockchain-based tracing, multi-keyword searching, decentralized trapdoor creation, one-to-many data sharing, and index-trapdoor non-differentiability. The overall flow of the proposed SCBA-PEKS scheme is shown in Fig. 2.

4.1 Setup

The cloud server begins the setup procedure to produce the global parameter SPP, where the security parameter is denoted as λ. It begins by constructing two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the identical prime factor p, with g serving as a generator of $\mathbb{G}_1$. The service provider then takes a random element h from $\mathbb{G}_1$ and creates a bilinear map $e: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Additionally, it selects a cryptographic hash function $H: \{0, 1\}^* \rightarrow \mathbb{Z}_p$. Hence, the global value is represented as SPP = {g, p, e, h, H, $\mathbb{G}_1$, $\mathbb{G}_2$}.

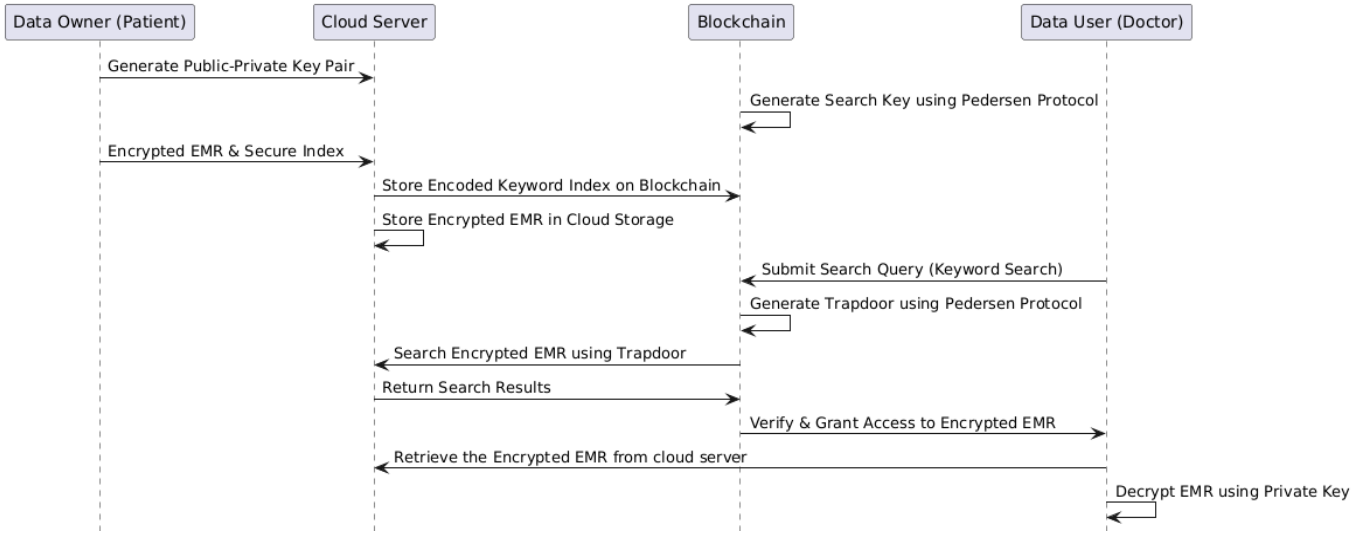

Fig. 2. Overall flow of the proposed design.

4.2 Keygen

First, the server generates a value $\alpha \leftarrow \mathbb{Z}_p^*$ randomly as its private key SK_C and generates public key PK_C as $PK_C = g^{\alpha}$. Similarly, the patients (Data Owners) and doctors (Data Users) select η and σ as their private keys (SK_O and SK_U) and generate their public keys as $PK_O = g^{\eta}$ and $PK_U = g^{1/\sigma}$ respectively. Furthermore, the cloud server identifies a group of blockchain nodes as Pedersen nodes. These nodes are responsible for obtaining a master share $\mathfrak{d}_i$, by conducting the first three phases of the Pedersen scheme as explained in Sec. 3.3. Once greater than k number of peers transmit $g^{\mathfrak{d}_i}$, the public search key PK_{BC} may be recovered as

$$PK_{BC} = g^{\mathfrak{d}} = \prod_{u=1}^{k} \left(g^{\mathfrak{d}_u} \right)^{\Gamma(u)} \tag{3}$$

where $$\Gamma(u) = \prod_{v=1, v \neq u}^{k} \frac{v}{v-1} (\mathrm{mod}\, p). \tag{4}$$

4.3 Encrypt & Indexgen

During this step, the data owner (patients) prepares the essential data to be sent to the server, such as a secure index, an encoded EMR, and a set of recovery keys. To preserve secrecy, the EMR must be encoded before transferring to the cloud server. Once the EMR has been generated from the patient/DO side, the DO selects a random integer δ from $\mathbb{Z}_p$ and encodes the produced EMR (M) by

$$CT_1 = M \oplus e\left(g^{\eta}, g^{\delta}\right),\ CT_2 = g^{\delta}. \tag{5}$$

The DO then computes the recovery key RK_M of the EMR for each data user as follows:

$$RK_M = \left\{ PK_{U_i}^{\eta} \right\}_{U_i \in U}. \tag{6}$$

U_i is the set of data users (doctors) taken from the user set. By producing the recovery key set $RK_{M \to U_i}$, access is restricted, allowing only the intended data user to execute the keyword search and decrypt the specific EMR. After the encryption and recovery key creation operations are completed, the owner retrieves the keyword set W from the EMR. The owner then creates a secure index IN for every keyword $w_i \in W$ as follows:

$$IN = \left\{ PK_C^{h} \cdot PK_{BC}^{\eta \cdot H(w_i)} \right\}_{w_i \in W}. \tag{7}$$

Finally, the DO sends the secure index IN, encoded record $C = \{C_1, C_2\}$ and recovery key set RK_M to the cloud service provider. It then keeps the encoded record and the recovery key in the server while the secure index IN is recorded on the blockchain.

4.4 TrapdoorGen

First, the doctor (data user) creates the keyword set $W' = H(w')$ and transmits it to blockchain nodes, which are nominated as Pedersen nodes for trapdoor creation. Then, these nodes work with other Pedersen peers to construct the search keyword set W' using the master secret $\mathfrak{d}_i$ and by following the process stated in the Pedersen secret sharing approach:

$$W' = \prod_{u=1}^{k} \left(\left(H\left(w_u' \right) \right)^{\mathfrak{d}_u} \right)^{\Gamma(u)} \tag{8}$$

where $$\Gamma(u) = \prod_{v=1, v \neq u}^{k} \frac{v}{v-1} (\mathrm{mod}\, p). \tag{9}$$

Next, the data user constructs the trapdoor using their private key μ and the Pedersen node by activating the trapdoor creation contract. The data user picks a random number $t \in \mathbb{Z}_p$ and calculates the trapdoor as follows:

$$T_1 = \mathrm{PK}_{\mathrm{C}}^{t};\ T_2 = \mathrm{PK}_{\mathrm{BC}}^{W'\cdot t}\cdot \mathrm{PK}_{\mathrm{C}}^{\sigma};\ T_3 = g^{h}. \quad (10)$$

Finally, the doctor transmits the trapdoor $T = \{T_1, T_2, T_3\}$ to the server for searching the EMR.

4.5 Search

After getting the search trapdoor from the doctors, the server initiates the search procedure on the blockchain to find out if the encrypted record CT has the keyword w' by comparing it with the index recorded on the blockchain. Specifically, the server evaluates each keyword w_i', to determine if

$$e\left(\frac{\mathrm{IN}}{T_3}, T_1\right) = e\left(T_2, \mathrm{RK}_M^{\sigma}\right). \quad (11)$$

If the equation is true, the blockchain notifies the server that the doctor is authorized to access the EMR information. Otherwise, it reports an error to the server. Following successful verification, the storage server forwards $\{\mathrm{CT}, \mathrm{RK}_{M\to U_i}\}$ to the doctor for further proceeding.

4.6 Decrypt

When the doctor/user U_i receives the ciphertext $\{\mathrm{CT}, \mathrm{RK}_{M\to U_i}\}$ from the cloud server, they decode it to extract the record M using the decryption method. The report can be obtained as follows.

$$M = C_1 \oplus e\left(\mathrm{RK}_{M\to U_i}, C_2\right)^{\sigma}. \quad (12)$$

The original EMR can be obtained only if the decrypting doctor's public key meets the re-encryption key requirements.

5. Security Analysis

This part explores the accuracy and security of our suggested SCBA-PEKS approach. Theorems 1 and 2 concern the security of electronic medical records and the keywords associated with them. We begin by showing that the cloud storage provider cannot derive the original data of any EMR if both the Divisible Computational Diffie-Hellman (DCDH) and Inverse Computational Diffie-Hellman (InvCDH) assumptions are true. Subsequently, we demonstrate that our suggested approach is safe against keyword-guessing outbreaks using the Divisible Decision Diffie-Hellman (DDDH) model by creating two comparable games. Finally, we demonstrate index non-differentiability compared to selected keyword assaults using the Decisional Bilinear Diffie-Hellman (DBDH) statement depending on the games given in Sec. 3.2.4.

5.1 Correctness of the Scheme

5.1.1 Correctness Search

We provide the following justification to illustrate the dependability of the trapdoor and encrypted index matching method:

$$\begin{aligned}
e\left(\frac{\mathrm{IN}}{T_3}, T_1\right) &= e\left(\frac{\mathrm{PK}_{\mathrm{C}}^{h}\cdot \mathrm{PK}_{\mathrm{BC}}^{\eta\cdot H(w_i)}}{g^{h}}, \mathrm{PK}_{\mathrm{C}}^{t}\right)\\
&= e\left(\frac{g^{\alpha\cdot h}\cdot g^{\beta\cdot\eta\cdot H(w_i')}}{g^{h}}, g^{\alpha\cdot t}\right)\\
&= e\left(\frac{g^{\alpha\cdot h}\cdot g^{\beta\cdot t\cdot W'}}{g^{h}}, g^{\alpha\cdot\eta}\right)\\
&= e\left(\frac{g^{\alpha\cdot h}\cdot g^{\beta\cdot t\cdot W'}}{g^{h}}, g^{\frac{\eta}{\sigma}\cdot\sigma\cdot\alpha}\right)\\
&= e\left(\frac{g^{h}\cdot g^{\alpha}\cdot g^{\beta\cdot t\cdot W'\cdot\sigma}}{g^{h}}, g^{\frac{\eta}{\sigma}\cdot\alpha}\right)\\
&= e\left(g^{\alpha\cdot\sigma}\cdot g^{\beta\cdot W'\cdot t}, g^{\frac{\eta}{\sigma}\cdot\alpha}\right)\\
&= e\left(\mathrm{PK}_{\mathrm{C}}^{\sigma}\cdot \mathrm{PK}_{\mathrm{BC}}^{W'\cdot t}, \mathrm{PK}_{\mathrm{U}}^{\eta\cdot\alpha}\right)\\
&= e\left(PK_{C}^{\sigma}\cdot PK_{BC}^{W'\cdot t}, RK_{M}^{\alpha}\right)\\
&= e\left(T_2, RK_{M}^{\alpha}\right).
\end{aligned} \quad (13)$$

5.1.2 Correctness Decrypt

Now, we demonstrate that how the valid users U_i are successfully obtaining the EMRs. It is equal to showing the accuracy of (12). We have

$$\begin{aligned}
&C_1 \oplus e\left(\mathrm{RK}_{M\to U_i}, C_2\right)^{\sigma}\\
&= M \oplus e\left(g^{\eta}, g^{\delta}\right) \oplus e\left(\mathrm{RK}_{M\to U_i}, g^{\delta}\right)^{\sigma}\\
&= M \oplus e\left(g^{\eta}, g^{\delta}\right) \oplus e\left(g^{\eta/\sigma}, g^{\delta}\right)^{\sigma}\\
&= M \oplus e\left(g^{\eta}, g^{\delta}\right) \oplus e\left(g^{\eta}, g^{\delta}\right)\\
&= M.
\end{aligned} \quad (14)$$

As a result, the valid user U_i has the ability to search and decode the encrypted record using keywords successfully.

5.2 The Security Proof

Theorem 1: The EMR is safe and secure if both the Decisional Composite Diffie-Hellman and Inverse Computational Diffie-Hellman statements are fulfilled in $\mathbb{G}_1$.

Proof: Equation (5) shows that the chance of an attacker $\mathcal{A}_1$ in decrypting a record is equal to computing g^{η}. In our method, the cloud server may get $g^{1/\eta}$, $g^{\eta/\sigma}$, $g^{1/\sigma}$.

Case 1: If the DCDH assumption holds in $\mathbb{G}_1$, the cloud server cannot realistically compute g^{η} from $g^{1/\eta}$ with considerable probability.

Case 2: Let's take η/σ as a, $g^{1/\sigma}$ as b, thus η may be represented as a/b. The statement of InvCDH makes it computationally impossible for an attacker to infer $g^{a/b}$ from (g, g^a, g^b) with nonnegligible probability. Similarly, the cloud server cannot reliably generate g^{η} from (g, $g^{\eta/\sigma}$, $g^{1/\sigma}$) with considerable possibility.

In summary, the storage server cannot extract the medical data M from the encrypted text CT with a significant possibility if both the DCDH and InvCDH constraints are true in $\mathbb{G}_1$.

Theorem 2: The proposed SCBA-PEKS approach is IND-KGA safe in the standard oracle model, provided the Divisible Decision Diffie-Hellman statement is met in $\mathbb{G}_1$.

Proof: The CS cannot determine the keywords from the secure index since it does not know μ. Thus, this theorem effectively demonstrates the security of keywords in our architecture. Let $\mathcal{A}_2$ be a polynomial-time opponent in the IND-KGA game, and $\mathcal{S}$ be the simulator designed to play a DDDH game. Here, we used the parameters g_1 and g_2 to represent the public key of blockchain PK_{BC} and the public key of the CS PK_C. The simulator $\mathcal{S}$ takes a DDDH instance ($A = g_2^a$, $B = g_2^b$, $C = g_2$, E), and tries to differentiate $E = g_1^{a/b}$ from a random element in $\mathbb{G}_1$. To demonstrate the security of the suggested approach, define the following two games:

Game 4. Assume $E = g_1^{a/b}$. Game 4 is substantially identical to the IND-KGA game, with the following modifications:

a. Setup: $\mathcal{S}$ picks an integer $l \in \mathbb{Z}_p$ randomly and generates the user U_i's public key as $PK_U = B = g_2^{b \cdot l}$. Thus, the user U_i's private key is $\sigma = 1/(b \cdot l \cdot a)$. Finally, the simulator $\mathcal{S}$ sends PK_U to $\mathcal{A}_2$.

b. Challenge: Upon getting keywords (w_0, w_1), $\mathcal{S}$ selects a random bit $\tau \in \{0, 1\}$. Then $\mathcal{S}$ sets $T_1 = A^l$, $T_2 = C \cdot E^{H(w_\tau)}$, and $T_3 = g^h$ respectively. Finally, $\mathcal{S}$ delivers the trapdoor $TD = \{T_1, T_2, T_3\}$ to $\mathcal{A}_2$.

c. Game 4 is equal to the IND-KGA exercise only if the created trapdoor is legitimate. Assume $r' = a \cdot l$, it will be derived as:

$$\begin{aligned} T_1 &= g_2^{a \cdot l} = g_2^{r'} = \mathrm{PK}_{\mathrm{C}}^{r'}, \\ T_2 &= A \cdot E^{H(w_\tau)} = g_2 \cdot g_1^{\frac{a}{b} H(w_\tau)} = g_2 \cdot g_1^{\frac{a \cdot l}{b \cdot l} H(w_\tau)} \\ &= g_2^{\sigma} \cdot g_1^{H(w_\tau) \cdot r'} \\ &= \mathrm{PK}_{\mathrm{C}}^{\sigma} \cdot \mathrm{PK}_{\mathrm{BC}}^{H(w_\tau) \cdot r'}, \\ T_3 &= g^h. \end{aligned} \tag{15}$$

Thus, Equation (15) is equal to (11). For $\mathcal{A}_2$, Game 4 is equal to the IND-KGA game. So, the advantage Ad or chance for $\mathcal{A}_2$ in winning Game 4 is:

$$Ad_{\mathcal{A}_2}^{G1}(\lambda) = Ad_{\mathcal{A}_2}^{\mathrm{IND\_KGA}}(\lambda). \tag{16}$$

Game 5. This one is identical to that of Game 4, only the value $E = g_1^{a/b}$ is substituted by a random variable $E \in \mathbb{G}_1$. As E is uniform in $\mathbb{G}_1$, we have:

$$\Pr\left[(\tau' = \tau)\right] = \frac{1}{2}. \tag{17}$$

Thus, the benefit for $\mathcal{A}_2$ in winning Game 2 is:

$$Ad_{\mathcal{A}_2}^{G2}(\lambda) = \left| \Pr\left[(\tau' = \tau)\right] - \frac{1}{2} \right| < \varepsilon'. \tag{18}$$

where ε' is an insignificant value. Meanwhile, the likelihood for $\mathcal{A}_2$ to differentiate between Game 4 and 5 is equivalent to the possibility to differentiate $g^{a/b}$ and the variable E, it is possible to have:

$$\begin{aligned} Ad_{\mathcal{A}_2}^{\mathrm{IND\_KGA}}(\lambda) &= Ad_{\mathcal{A}_2}^{G1}(\lambda) \\ &= Ad_{\mathcal{A}_2}^{G2}(\lambda) + Ad_{\mathbb{G}_1, \mathcal{A}_2}^{\mathrm{DDDH}}(\lambda) \\ &= \varepsilon' + \varepsilon = \varepsilon. \end{aligned} \tag{19}$$

If the DDDH hypothesis holds in $\mathbb{G}_1$, then ε is insignificant. Henceforth, the benefit for $\mathcal{A}_2$ to succeed in the IND-KGA game is insignificant.

Theorem 3: Our proposed approach is secure against the chosen keyword attacks under Index Non-Differentiability Chosen Keyword Attack (IND-CKA) specified in Game 1 and Game 2 (mentioned in Sec. 4.3.1) if the DBDH hypothesis is true.

Lemma 1. Considering the DBDH problem is impossible to solve, the suggested technique is IND-CKA safe against a PPT adversary $\mathcal{A}_1$.

Proof. Consider that the attacker $\mathcal{A}_1$ is a malicious user (including the blockchain and the external adversary) in our approach. We build a simulator called $\mathcal{S}$ to find the solution for the DBDH problem given a tuple (g^a, g^b, g^c, Z), where the benefit of $\mathcal{S}$ is defined as $\varepsilon' = \varepsilon/q_I$. This is assuming that $\mathcal{A}_1$ has a benefit of ε in breaking the suggested approach and generates at most q_I queries to the index oracle. The following scenario describes the interactions between $\mathcal{A}_1$ and $\mathcal{S}$.

1. Setup: The ledger contains our scheme's public parameter $SPP = \{\mathbb{G}_1, \mathbb{G}_2, g, p, e, h = g^a, H\}$. In order to create a public search key $PK_{BC} = g^{\beta}$, the attacker $\mathcal{A}_1$ executes the Pedersen Protocol. Meanwhile, $\mathcal{S}$ creates the public key of the server as $PK_C = (g^{\alpha})^{v}$, where the parameter v is randomly picked from $\mathbb{Z}_p^*$ and published in the ledger. The challenger is sent the public key t to $\mathcal{A}_1$.

2. Query Index I: When an index query is raised by the adversary $\mathcal{A}_1$ about the keyword $w_i \in \{0, 1\}^*$, the simulator $\mathcal{S}$ chooses an element h from the SPP and computes the index as

$$\mathrm{PK}_{\mathrm{C}}^{h} \cdot \mathrm{PK}_{\mathrm{BC}}^{\mu \cdot H(w_i)}. \tag{20}$$

3. Challenge: In this phase, two never-queried keywords w_0 and w_1 are sent to the simulator by the adversary $\mathcal{A}_1$. $\mathcal{S}$ then randomly picks $b \in \{0, 1\}$. Afterwards, $\mathcal{S}$ invokes the Pedersen scheme to get $(g^b)^\beta$ of the keyword w_b for computing the challenge index (I^*), which is computed as:

$$Z^{\nu} \cdot \left(g^{b}\right)^{\beta \cdot \mu \cdot H(w_b)}. \tag{21}$$

4. Query Phase II: With the exception of the two challenge keywords, $\mathcal{A}_1$ may still adaptively query the index oracle model for any keywords. But the simulator $\mathcal{S}$ gives the same reply as the first index query to the adversary $\mathcal{A}_1$.

5. Predict: As a guess for b, $\mathcal{A}_1$ produces $b' \in \{0, 1\}$. Guessing of $Z = (g, g)^{abc}$ by $\mathcal{S}$ yields 1 if $b' = b$ and 0 otherwise.

It is easily confirmed that when $Z = (g, g)^{abc}$, the security game results produced in our scheme by $\mathcal{S}$ are precisely calculated. The challenge index and the public keys of blockchain and cloud servers are similar to the actual generation owing to the unpredictability of the element v. Additionally, the indexes delivered by $\mathcal{S}$ during Query Index I are properly disseminated since H is selected homogeneously.

Probability Investigation: Assuming that the set of keywords challenged to the index oracle has not been queried by $\mathcal{A}_1$ and $\mathcal{S}$ accurately responds to $\mathcal{A}_1$ during the actual attack. As shown in the reference [40], the probability of $\mathcal{S}$ returning the right responses is at least $\varepsilon/q_{\mathrm{I}}$. Given that the probability of $\mathcal{S}$ terminating the challenge phase and index query is zero and the possibility of $\mathcal{S}$ successfully solving the DBDH problem is no less than $\varepsilon/q_{\mathrm{I}}$.

6. Performance Analysis

This section compares our proposed approach to earlier searchable encryption systems. Furthermore, we implement many existing techniques and compare their computational and communication overheads to those of our proposed approach. Table 1 compares our proposed system SCBA-PEKS to available schemes [29, 34, 35]. The symbols '✓' and '×' indicate if a certain feature is provided.

Functionality/Scheme	Zhang et al. [29]	Yu et al. [34]	Fanfan Shen et al. [35]	Proposed
One-to-Many	✓	✓	✓	✓
Multi Keyword	×	×	✓	✓
Secure Channel free	×	×	✓	✓
Trapdoor Ind (Offline KGA)	✓	×	×	✓
Blockchain	✓	✓	✓	✓

Tab. 1. Function comparison.

Table 2 compares the communication overheads of our method to those of comparable schemes for the generation of various keys, secure index, encryption, trapdoor, and search. The symbols in Tab. 2 are as follows: l – Sum of attributes in an access policy; Sym – symmetric cryptography; N_{u} – Sum of attributes in the user attribute set; C_{E} represents exponentiation over group $\mathbb{G}_1$; C_P represents bilinear pairing process; C_{H} mentions map to point hash operation; C_M – A multiplication process over $\mathbb{Z}_p^*$, $\mathbb{G}_1$, $\mathbb{G}_2$; $\mathscr{b}/f$ - Sum of keywords in index/trapdoor; C_{O} – A bit exclusive-OR operation. Compared to the other three approaches, our scheme's data owner uses fewer computing resources in a multi-receiver environment. The key generating overheads in the scheme [34] are low, while schemes [29], [35] rely on the Pedersen protocol like our proposed solution. Furthermore, index creation in single-keyword schemes [35] and ours involves constant operations, but in multi-keyword approaches, it is directly proportional to the quantity of keywords $\mathscr{b}$. In terms of trapdoor production and search operations, our scheme and scheme [35] enable the data user to construct trapdoors with many keywords at the same time, greatly lowering search overhead when compared to schemes [29], [34]. In schemes [29] and [34], the sum of trapdoors required is proportional to the sum of keywords. Overall, our method outperforms the various existing schemes in terms of computing effectiveness, making it the optimal choice when many users are authorized to search the owner's encrypted medical records (EMRs).

Table 3 summarizes our proposed scheme's storage and transmission costs in comparison to other available methods. $|\mathbb{G}_1|$ and $|\mathbb{G}_2|$ represents the size of elements in group

Scheme	Index Generation	Key Generation	Encrypt	Trapdoor Generation	Search
Fanfan Shen et al. [35]	$(2+\mathscr{b})\ C_{\mathrm{E}} + C_{\mathrm{P}}$	**DU:** $4C_{\mathrm{E}} + lC_{\mathrm{E}} + C_{\mathrm{H}}$; **DO:** $3C_{\mathrm{E}} + C_{\mathrm{H}}$; **Proxy:** $4C_{\mathrm{E}} + 3C_{\mathrm{P}} + 2C_{\mathrm{H}}$;	$(1+3l)C_{\mathrm{E}}+1C_{\mathrm{E}}+(2+l)C_{\mathrm{H}}+ Sym$	$(2+f)\ C_{\mathrm{E}} + C_{\mathrm{P}}$	$(2+f)\ C_{\mathrm{P}}$
Yu et al. [34]	$\mathscr{b}C_{\mathrm{P}} + 2\mathscr{b}C_{\mathrm{E}} + 2\mathscr{b}C_{\mathrm{H}}$	*Pedersen*	$C_{\mathrm{O}} + C_{\mathrm{H}} + 7C_{\mathrm{E}} + 2C_{\mathrm{M}} + C_{\mathrm{P}}$	$C_{\mathrm{P}} + 2C_{\mathrm{E}} + 2C_{\mathrm{H}}$	$C_{\mathrm{P}} + 2C_{\mathrm{E}} + 2C_{\mathrm{O}}$
Zhang et al. [29]	$2lC_{\mathrm{E}} + \mathscr{b}C_{\mathrm{E}}$	*Pedersen*	$3lC_{\mathrm{E}} + C_{\mathrm{E}} + Sym$	$(2 + N_{\mathrm{u}})\ C_{\mathrm{E}} + C_{\mathrm{H}}$	$2C_{\mathrm{P}} + N_{\mathrm{u}}C_{\mathrm{E}}$
Proposed	$2\mathscr{b}C_{\mathrm{P}} + (2\mathscr{b} + 1)\ C_{\mathrm{E}} + \mathscr{b}C_{\mathrm{H}}$	**PSK:** *Pedersen*; **CS, DO, DU:** $1C_{\mathrm{E}}$	$C_{\mathrm{O}} + C_{\mathrm{E}} + C_{\mathrm{P}}$	*Pedersen* + $(f+1)\ C_{\mathrm{E}}$ + $2f\ C_{\mathrm{H}} + (f+1)\ C_{\mathrm{M}}$	$C_{\mathrm{E}} + 2C_{\mathrm{P}}$

Tab. 2. Computation cost.

Scheme	Ciphertext Length	Index Length	Trapdoor Length
Fanfan Shen et al. [35]	$\|\mathbb{G}_1\| + \mathbb{G}_2\|$	$(2 + e)\|\mathbb{G}_1\|$	$(2 + f)\|\mathbb{G}_1\|$
Yu et al. [34]	$3\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	$\|\mathbb{G}_1\| + \|\log_p\|$	$2\|\mathbb{G}_1\|$
Zhang et al. [29]	$\|\mathbb{G}_2\| + (4l + 1)\|\mathbb{G}_1\|$	$\|\mathbb{G}_2\| + 2l\|\mathbb{G}_1\|$	$(\|\mathbb{G}_2\| + 1)\|\mathbb{G}_1\|$
Proposed	$\|\mathbb{G}_1\|$	$\|\mathbb{G}_1\| + e\|\mathbb{G}_2\|$	$2\|\mathbb{G}_1\|$

Tab. 3. Communication and storage overhead.

$\mathbb{G}1$ and $\mathbb{G}2$, respectively. Our suggested technique has a reasonably low overhead for blockchain nodes that use the Pedersen scheme to generate global values and keys. Furthermore, for single keyword schemes, the secure index generation and trapdoor sizes are fixed. However, for multiple keyword schemes, the above-said overhead will vary based on the sum of keywords. Notably, in our system, the trapdoor volume for searching with multi-keywords is unaffected by the sum of keywords. Overall, the proposed SCBA-PEKS scheme's computation and storage overhead are appropriate for IoT applications.

To make performance more understandable, we built our scheme in Java, along with schemes [29, 34, 35], utilizing the Pairing-Based Cryptographic (PBC) library [24]. We tested our proposed technique with other schemes on a PC running on Ubuntu 20.04.4 LTS 64-bit OS equipped with an Intel Core i7 processor and 8 Giga Byte of memory. We also employed a type A 1024-bit field order and 160-bit group order elliptic curve to create the cryptographic environment. To test the effectiveness of the blockchain implementation, we used Hyperledger Fabric v0.6.0 running on Docker. The experiment aimed to assess computational and storage costs. The experimental results shown in the upcoming figures are the average value of 10 runs.

In Fig. 3, we compare the computational burden of the encryption technique across different schemes while increasing the number of encrypted EMRs from 100 to 1000, assuming 100 data users per EMR. The computational cost of the encryption technique is almost linear with the quantity of EMRs. Notably, Yu et al. [34] show much greater computational overhead than previous systems, whereas our scheme has reduced computational costs and improved performance.

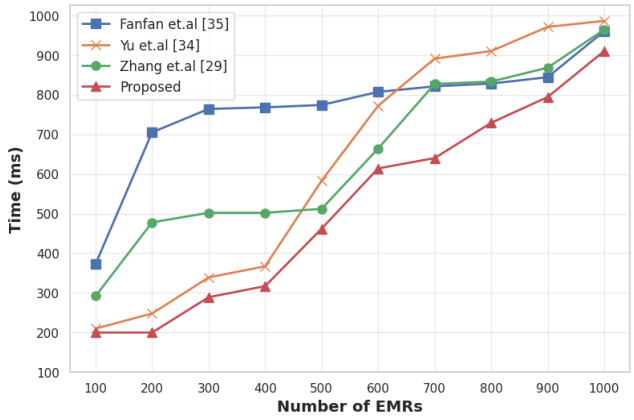

Fig. 3. Computation overhead of EMR encryption and recovery key creation.

In Figs. 4 and 5, we assess the computational overhead of the index creation and trapdoor creation processes by altering the sum of keywords in the index from 10 to 100 while maintaining a fixed amount of Pedersen nodes (5) engaged in trapdoor generation for each data consumer. The findings indicate that the computational overheads of all approaches rise linearly with the sum of keywords. Our suggested approach produces good outcomes as the sum of keywords grows.

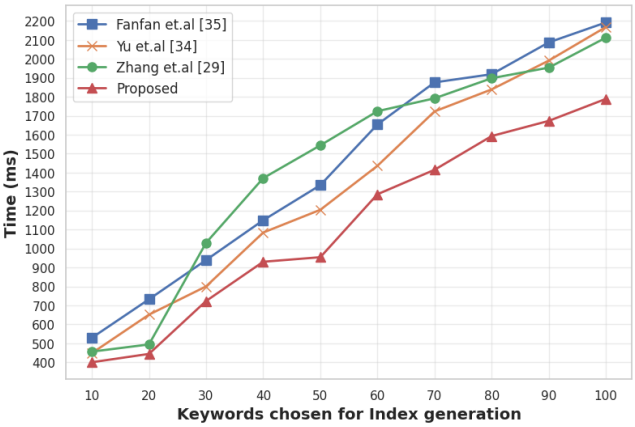

Fig. 4. Computation cost of Index calculation process.

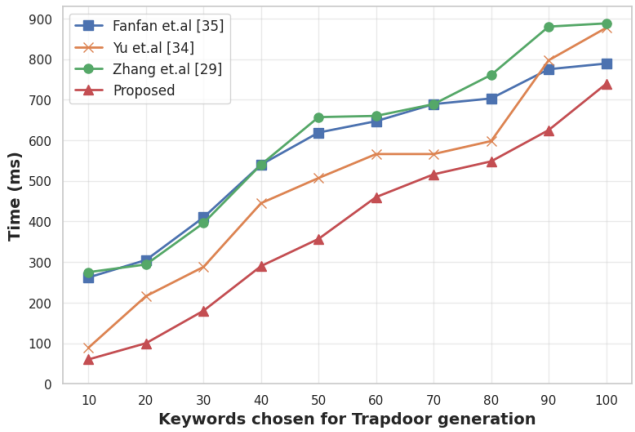

Fig. 5. Computation cost of trapdoor creation process.

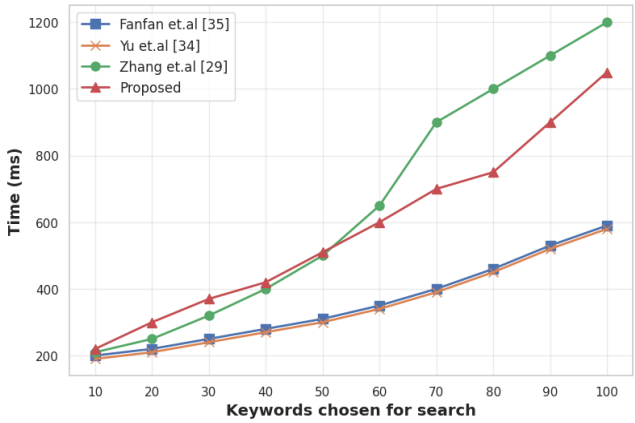

Fig. 6. Computation overhead of keyword search.

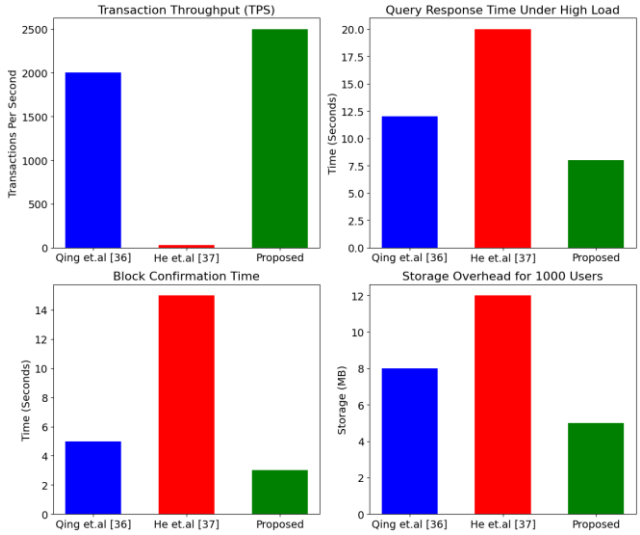

Fig. 7. Comparative performance analysis.

Figure 6 depicts the search algorithm's computing load, with a focus primarily on index and trapdoor matching and ignoring the post-search consensus step. We compute the computational overhead by altering the sum of keywords from 10 to 100. It is clear that the computing burden of all systems grows linearly with the sum of keywords. The quantity of keywords used in the trapdoor query has a significant impact on search performance. Schemes [29] and [34] use a single keyword search technique, resulting in quicker speed but worse data retrieval accuracy. In contrast, our suggested technique outperforms existing approaches with respect to search result correctness.

Figure 7 shows the comparative analysis of blockchain overload in high-user scenarios of different schemes. From the results, we can see that the proposed technique achieves higher transaction throughput (2500 Transactions Per Second (TPS)), faster query response time (~8 seconds), and lower block confirmation time (~3 seconds) compared to existing approaches. Qing Wu et al. [36] do an average job, while He et al. [37] fail with merely 30 TPS, with slower queries (~20 sec) and high storage overhead (12 MB per 1000 users) due to Ethereum's limitations. The result concludes that SCBA-PEKS is the most scalable and practical technique for securing the efficient searchability of EMRs in large-scale deployment.

The proposed system outperforms state-of-the-art (SOTA) results due to its improved cryptographic design, efficient search algorithms, and improved blockchain implementation. It significantly decreases computational overhead by maintaining a linear increase in cost, guaranteeing faster encryption, index calculation, and trapdoor creation. Unlike existing schemes that rely on single-keyword searches, the proposed system supports multi-keyword searches, leading to higher accuracy without compromising query speed. Its blockchain integration achieves greater transaction throughput and lower latency compared to He et al. [37], while also minimizing block confirmation time and storage overhead. In contrast, existing SOTA methods suffer from higher computational costs, inefficient search mechanisms, and blockchain limitations, which hinder their performance in high-user scenarios.

7. Conclusion

In this article, we provide Secure Channel free Blockchain-Assisted Public Key Encryption with Keyword Search (SCBA-PEKS) for wireless communication environments. Our technique meets practical demands in a multi-user scenario by allowing owners to easily delegate search capabilities to users while incurring minimum computational and storage costs. Utilizing blockchain, our approach employs the Pedersen Protocol for threshold trapdoor generation, enabling efficient one-to-many searches. Each authorized user can successfully query encrypted EMRs and extract appropriate records leveraging signal processing techniques to ensure accurate retrieval. Our security analysis and the correctness of our studies demonstrate the scheme's robustness to IND-KGA assaults, addressing critical security challenges in wireless data transmission. Comprehensive comparisons, comprising performance evaluations and theoretical assessments show that the proposed system is much more efficient with respect to computation and storage costs than other existing approaches, making it suitable for resource-constrained wireless devices. In the future, we are planning to enhance the performance of the blockchain using various layer 2 scaling solutions and more efficient multi-keyword search with better ranking mechanisms. Additionally, we aim to introduce dynamic policy updates for flexible access control in hospital scenarios so that their practical performance can be assessed in wireless healthcare networks.

Acknowledgment

Due to institutional and intellectual property constraints, the source code and detailed model are not publicly available at this stage. However, the manuscript provides sufficient algorithmic and architectural details to support understanding and reproducibility.

Appendix A

A1. Bilinear Map

Consider $\mathbb{G}_1$ and $\mathbb{G}_2$, two cyclic groups with a significantly prime order p. Assume g is a generator of $\mathbb{G}_1$. A bilinear map e: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is defined when the following three requirements met:

1. *Bilinearity:* For each $m, n \in \mathbb{G}_1$, $e(g^m, g^n) = e(g, g)^{mn}$.
2. *Nondegeneracy:* $e(g, g) = 1$ where g is a generator of $\mathbb{G}_1$.

3. *Computability:* For u and $v \in \mathbb{G}_1$, $e(u, v)$ can be easily calculated.

A2. Cryptographic Assumptions

Assume $\mathbb{G}$ is a cyclic group of a significant prime factor p with a g generator. We make the succeeding hypothesis in our proposed approach.

- **Divisible Decision Diffie-Hellman (DDDH) Assumption**

Consider (g, g^m, g^n, r) where m, n, and r are randomly picked in $\mathbb{Z}_p$. We define the function of advantage as Ad of an attacker $\mathcal{A}$ as:

$$Ad^{\mathrm{DDDH}}_{\mathbb{G}_1,\mathcal{A}}(\lambda) = \left|\Pr\left[\mathcal{A}\left(g, g^m, g^n, g^{m/n}\right)=1\right] - \Pr\left[\mathcal{A}\left(g, g^m, g^n, g^r\right)=1\right]\right|. \quad (22)$$

Here, λ denotes the security criteria. The DDDH statement applies when $Ad^{\mathrm{DDDH}}_{\mathbb{G}_1,\mathcal{A}}$ is insignificant for the attacker [38].

- **Divisible Computation Diffie-Hellman (DCDH) Assumption**

Consider (g, g^m, g^n) where m, n are randomly picked in $\mathbb{Z}_p$. The benefit for an attacker $\mathcal{A}$ to calculate $g^{m/n}$ is insignificant [38].

- **Inverse Computational Diffie-Hellman (InvCDH) Assumption**

Consider (g, g^m), where m is randomly picked in $\mathbb{Z}_p$, the benefit for an attacker $\mathcal{A}$ to compute $g^{1/m}$ is trivial [38].

- **Decisional Bilinear Diffie-Hellman (DBDH) Assumption**

Consider the function of bilinear pairing e: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and a tuple (g, g^m, g^n, g^o, B) as input, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are two groups of prime order p, g is a generator of $\mathbb{G}_1$, and m, n, o are randomly picked from $\mathbb{Z}^*_p$, an algorithm $\mathcal{B}$ attempts to identify the value of B as $B = e(g, g)^{mno}$ or $B = e(g, g)^q$, where the value of q is selected from $\mathbb{Z}^*_p$. The benefit of $\mathcal{B}$ in breaching the DBDH assumption is 2ε if

$$\left|\begin{array}{l}\Pr\left[\mathcal{A}\left(g, g^m, g^n, g^o, e(g,g)^{mno}\right)=1 : g \in \mathbb{G},\ m,n,o \in \mathbb{Z}^*_p\right] - \\ \left[\mathcal{A}\left(g, g^m, g^n, g^o, e(g,g)^{q}\right)=1 : g \in \mathbb{G},\ m,n,o \in \mathbb{Z}^*_p\right]\end{array}\right| \geq 2\varepsilon \quad (23)$$

The DBDH assumption is valid if no Probabilistic Polynomial Time (PPT) attacker can solve the issue with a non-zero advantage [16].

A3. Pedersen Secret Sharing (k, n) Protocol

In a restricted space $\mathbb{G}_p$ (where p is a significant prime value), n contributors $\mathrm{P} = (P_1, P_2, \ldots, P_n)$ execute the following procedure to disclose the Primary Secret [39].

(1) Compute the Primary Secret $S = \sum_{u=1}^{n} S_u$ where $S_u \in \mathbb{Z}^*_p$ is chosen autonomously by each participant P_u.

(2) Compute and Disseminate Sub-shares: Each participant P_u randomly selects polynomial function $\mathbb{f}_u(x)$ of a degree $k-1$ such that $\mathbb{f}_u(x) = S_u$. Subsequently, it computes n sub-shares $\mathbb{s}_{uv} = \mathbb{f}_u(x_v)$ for $u = 1, 2, ..., n$ and transfers $\mathbb{s}_{uv}$ to P_v via a secure medium.

(3) Generate Master-shares: Once the n sub-shares $\mathbb{s}_{uv}$ ($v = 1, 2, ..., n$) had been obtained, P_u computes its master-share $\mathbb{S}_u = \sum_{u=1}^{n} \mathbb{S}_{vu}$.

(4) Reconstruction of Master Secret: Let $P_T \subset P$ be the subset of participants ($k \leq |P_T| \leq n$) collaborate to restore the master secret:

$$S = \sum_{P_u \in P_T} \mathbb{S}_i \prod_{P_u, P_v \in P_T, u \neq v} \frac{v}{v-u} (\mathrm{mod}\ p). \quad (24)$$

In this work, blockchain nodes collaboratively work with Pedersen Protocol to produce the key to achieve the public search and the search trapdoor. The Pedersen Protocol involves a collection of blockchain nodes, with the master secret key serving as the target secret.

A4. Hyperledger Fabric Platform - Permissioned Blockchain

Hyperledger Fabric [40] is an open-source blockchain platform developed for corporate use in a permissioned network. We chose Hyperledger Fabric because of its privacy features, scalability, transaction efficiency, interoperability, and fine-grained access permissions to EMR information. This option considerably decreases turnaround time for EMR storage and sharing, improves medical decision-making, and saves overall costs. Nodes on a permissioned blockchain, such as Hyperledger Fabric, must be verified before entering the network. Nodes in such a blockchain are categorized into two forms: user nodes and consensus nodes. User nodes start transactions, which are subsequently routed to consensus nodes. These consensus nodes initiate the automated execution of smart contracts, and the output is saved in the ledger once the consensus is reached. We can also use smart contracts to perform a majority of tasks, such as data tracking, access control and behavior logging, while the consensus mechanism assures that ledger data is tamper-proof. In our approach, the blockchain serves three important functions: recording, tracking, and creating trapdoors.

References

[1] CHEN, C. M., LIU, S., LI, X., et al. A provably-secure authenticated key agreement protocol for remote patient monitoring IoMT. *Journal of Systems Architecture*, 2023, vol. 136, p. 1–11. DOI: 10.1016/j.sysarc.2023.102831

[2] WU, G., WANG, H., YANG, Z., et al. Electronic health records sharing based on consortium blockchain. *Journal of Medical Systems*, 2020, vol. 48, p. 1–23. DOI: 10.1007/s10916-024-02120-9

[3] ZHANG, G., YANG, Z., LIU, W. Blockchain-based privacy preserving e-health system for healthcare data in cloud. *Computer Networks*, 2022, vol. 203, p. 1–9. DOI: 10.1016/j.comnet.2021.108586

[4] XIANG, X., ZHAO, X. Blockchain-assisted searchable attribute-based encryption for e-health systems. *Journal of Systems Architecture*, 2022, vol. 124, p. 1–9. DOI: 10.1016/j.sysarc.2022.102417

[5] YAN, X., ZHENG, C., TANG, Y., et al. Dynamic forward secure searchable encryption scheme with phrase search for smart healthcare. *Journal of Systems Architecture*, 2023, vol. 144, p. 1–10. DOI: 10.1016/j.sysarc.2023.103003

[6] XIONG, Y., LUO, M. X. Searchable encryption scheme for large data sets in cloud storage environment. *Radioengineering*, 2024, vol. 33, no. 2, p. 223–235. DOI: 10.13164/re.2024.0223

[7] WANG, B. Y., LI, H., LIU, X. F., et al. Preserving identity privacy on multi-owner cloud data during public verification. *Security and Communication Networks*, 2014, vol. 7, no. 12, p. 2104–2113. DOI: 10.1002/sec.922

[8] BENALOH, J., CHASE, M., HORVITZ, E., et al. Patient controlled encryption: Ensuring privacy of electronic medical records. In *Proceedings of 2009 ACM Cloud Computing Security Workshop (CCSW '09).* Chicago (USA), 2009, p. 103–114. DOI: 10.1145/1655008.1655024

[9] CHENAM, V. B., ALI, S. T. A designated cloud server-based multi-user certificateless public key authenticated encryption with conjunctive keyword search against IKGA. *Computer Standards & Interfaces*, 2022, vol. 81, p. 1–21. DOI: 10.1016/j.csi.2021.103603

[10] YANG, N., ZHOU, Q., HUANG, Q., et al. Multi-recipient encryption with keyword search without pairing for cloud storage. *Journal of Cloud Computing*, 2022, vol. 11, no. 1, p. 1–12. DOI: 10.1186/s13677-022-00283-9

[11] FU, X., WANG, H., SHI, P., et al. Teegraph: A blockchain consensus algorithm based on TEE and DAG for data sharing in IoT. *Journal of Systems Architecture*, 2022, vol. 122, p. 1–9. DOI: 10.1016/j.sysarc.2021.102344

[12] KHAN, M. A., SALAH, K. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 2018, vol. 82, p. 395–411. DOI: 10.1016/j.future.2017.11.022

[13] YU, G., WANG, X., YU, K., et al. Survey: Sharding in blockchains. *IEEE Access*, 2020, vol. 8, p. 14155–14181. DOI: 10.1109/ACCESS.2020.2965147

[14] LI, H., TIAN, H., ZHANG, F., et al. Blockchain-based searchable symmetric encryption scheme. *Computers and Electrical Engineering*, 2019, vol. 73, p. 32–45. DOI: 10.1016/j.compeleceng.2018.10.015

[15] KUO, T. T., KIM, J., GABRIEL, R. A. Privacy-preserving model learning on a blockchain network-of-networks. *Journal of the American Medical Informatics Association*, 2020, vol. 27, no. 3, p. 343–354. DOI: 10.1093/jamia/ocz214

[16] BONEH, D., DI CRESCENZO, G., OSTROVSKY, R., et al. Public key encryption with keyword search. In *International Conference on the Theory and Applications of Cryptographic Techniques*. 2004, p. 506–522. DOI: 10.1007/978-3-540-24676-3_30

[17] YANG, L., WANG, G., LI, J. Keyword guessing attacks on a public key encryption with keyword search scheme without random oracle and its improvement. *Information Sciences*, 2019, vol. 479, p. 270–276. DOI: 10.1016/j.ins.2018.12.004

[18] HUANG, Q., LI, H. An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks. *Information Sciences*, 2017, vol. 403-404, p. 1–14. DOI: 10.1016/j.ins.2017.03.038

[19] QIN, B., CHEN, Y., HUANG, Q., et al. Public-key authenticated encryption with keyword search revisited: Security model and constructions. *Information Sciences*, 2020, vol. 516, no. C, p. 515–528. DOI: 10.1016/j.ins.2019.12.063

[20] LI, J., WANG, M., YANG, L., et al. ABKS-SKGA: Attribute-based keyword search secure against keyword guessing attack. *Computer Standards & Interfaces*, 2021, vol. 74, p. 1–7. DOI: 10.1016/j.csi.2020.103471

[21] YANG, L., LI, J., ZHANG, Y. Secure channel free certificate-based searchable encryption withstanding outside and inside keyword guessing attacks. *IEEE Transactions on Services Computing*, 2019, vol. 14, no. 6, p. 2041–2054. DOI: 10.1109/TSC.2019.2910113

[22] YANG, L., LI, J. Lightweight public key authenticated encryption with keyword search against adaptively-chosen-targets adversaries for mobile devices. *IEEE Transactions on Mobile Computing*, 2021, vol. 21, no. 12, p. 4397–4409. DOI: 10.1109/TMC.2021.3077508

[23] PAN, X., LI, F. Public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability. *Journal of Systems Architecture*, 2021, vol. 115, p. 1–8. DOI: 10.1016/j.sysarc.2021.102075

[24] ZHANG, X., XU, C., WANG, H., et al. FS-PEKS: Lattice-based forward secure public-key encryption with keyword search for cloud-assisted industrial Internet of Things. *IEEE Transactions on Dependable and Secure Computing*, 2021, vol. 18, no. 3, p. 1019–1032. DOI: 10.1109/TDSC.2019.2914117

[25] LEE, C. Y., LIU, Z. Y., TSO, R., et al. Privacy-preserving bidirectional keyword search over encrypted data for cloud-assisted IIoT. *Journal of Systems Architecture*, 2022, vol. 130, p. 1–11. DOI: 10.1016/j.sysarc.2022.102642

[26] XU, P., TANG, S., XU, P., et al. Practical multi-keyword and boolean search over encrypted e-mail in cloud server. *IEEE Transactions on Services Computing*, 2021, vol. 14, no. 6, p. 1948–1960. DOI: 10.1109/TSC.2019.2903502

[27] LIU, X., YANG, G., SUSILO, W., et al. Privacy-preserving multi keyword searchable encryption for distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 2021, vol. 32, no. 3, p. 561–574. DOI: 10.1109/TPDS.2020.3027003

[28] SHAMSHAD, S., MINAHIL, MAHMOOD, K., et al. A secure blockchain-based e-health records storage and sharing scheme. *Journal of Information Security and Applications*, 2020, vol. 55, p. 1–17. DOI: 10.1016/j.jisa.2020.102590

[29] ZHANG, L., ZHANG, T., WU, Q., et al. Secure decentralized attribute-based sharing of personal health records with blockchain. *IEEE Internet of Things Journal*, 2022, vol. 9, no. 14, p. 12482 to 12496. DOI: 10.1109/JIOT.2021.3137240

[30] ZHANG, Y., XU, C., NI, J., et al. Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage. *IEEE Transactions on Cloud Computing*, 2019, vol. 9, no. 4, p. 1335–1348. DOI: 10.1109/TCC.2019.2923222

[31] JIANG, P., QIU, B., ZHU, L., et al. SearchBC: A blockchain-based PEKS framework for IoT services. *IEEE Internet of Things*

Journal, 2020, vol. 8, no. 6, p. 5031–5044. DOI: 10.1109/JIOT.2020.3036705

[32] YANG, Y., HU, M., CHENG, Y., et al. Keyword searchable encryption scheme based on blockchain in cloud environment. In *Proceedings of the 3rd International Conference on Smart Blockchain (SmartBlock)*. Zhengzhou (China), 2020, p. 1–4. DOI: 10.1109/SmartBlock52591.2020.00013

[33] NIU, S., CHEN, L., WANG, J., et al. Electronic health record sharing scheme with searchable attribute-based encryption on blockchain. *IEEE Access*, 2019, vol. 8, p. 7195–7204. DOI: 10.1109/ACCESS.2019.2959044

[34] YU, J., LIU, S., XU, M., et al. An efficient revocable and searchable MA-ABE scheme with blockchain assistance for C-IoT. *IEEE Internet of Things Journal*, 2023, vol. 10, no. 3, p. 2754 to 2766. DOI: 10.1109/JIOT.2022.3213829

[35] SHEN, F., SHI, L., ZHANG, J., et al. BMSE: Blockchain-based multi-keyword searchable encryption for electronic medical records. *Computer Standards & Interfaces*, 2024, vol. 89, p. 1–10. DOI: 10.1016/j.csi.2023.103824

[36] WU, Q., LAI, T., ZHANG, L., et al. Blockchain-enabled multi-authorization and multi-cloud attribute-based keyword search over encrypted data in the cloud. *Journal of Systems Architecture*, 2022, vol. 129, p. 1–12. DOI: 10.1016/j.sysarc.2022.102569

[37] HE, B., FENG, T., FANG, J., et al. A secure and efficient charitable donation system based on Ethereum blockchain and searchable encryption. *IEEE Transactions on Consumer Electronics*, 2023, vol. 70, no. 1, p. 263–276. DOI: 10.1109/TCE.2023.3323356

[38] BAO, F., DENG, R. H., ZHU, H. Variations of Diffie-Hellman problem. In *Proceedings of Information and Communications Security (ICICS)*. 2003, p. 301–312. DOI: 10.1007/978-3-540-39927-8_28

[39] PEDERSEN, T. P. A threshold cryptosystem without a trusted party. In *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*. 1991, p. 522–526. DOI: 10.1007/3-540-46416-6_47

[40] ANDROULAKI, E., BARGER, A., BORTNIKOV, V., et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*. Porto (Portugal), 2018, p. 1–15. DOI: 10.1145/3190508.3190538

About the Authors

RAJESH KUMAR Suyamburaj (corresponding author) was born in India. He received his ME degree in Computer Science and Engineering from Apollo College of Engineering, Tamilnadu, India, in the year 2014. His research interests are cloud computing, cryptography, Deep Learning, and soft computing. Currently, he is working as an Assistant Professor in the Department of Computer Science and Engineering at KPR Institute of Engineering and Technology, Tamilnadu, India.

GOMATHI Velusamy is the Head of the Department of Computer Science and Engineering at the National Engineering College, Kovilpatti, Tamil Nadu. She earned her Ph.D. from Anna University Chennai in 2011, M.Tech. from the Indian Institute of Technology Madras in 2005, and B.E. from Thiagarajar College of Engineering, Madurai, in 1997. Dr. Gomathi has an extensive publication record, including 32 journal articles, 5 books, and 28 conference proceedings. She has guided multiple doctoral theses and led eight research projects. Her contributions have been recognized with several awards, such as the Best Paper Award at the eTOTAL International Conference in 2023 and the NVIDIA DLI Ambassador accolades in 2021 and 2022. Dr. Gomathi's research interests lie in Computer Science and Artificial Intelligence.

VIVEKRABINSON Kanagamani was born in India. He received both his B.E. and M.E. in Computer Science and Engineering from Kalasalingam Institute of Technology, Tamilnadu, India, in the years 2014 and 2016, respectively. In 2021, he received his Ph.D. in Information and Communication Engineering from Anna University, Tamil Nadu, India. His research interests are cloud computing, cryptography, blockchain and the internet of things. Currently, he is working as an Assistant Professor in the Department of Computer Science and Engineering at Kalasalingam Academy of Research and Education, Tamilnadu, India.