Received: 8 March 2023 | Revised: 4 October 2023 | Accepted: 6 October 2023

DOI: 10.1002/spy2.351

SURVEY PAPER

WILEY

Personal health record storage and sharing using searchable encryption and blockchain: A comprehensive survey

Abhishek Bisht[1] | **Ashok Kumar Das**[1] | **Debasis Giri**[2]

[1]Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India

[2]Department of Information Technology, Maulana Abul Kalam Azad University of Technology, Nadia, West Bengal, India

Correspondence
Ashok Kumar Das, Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India.
Emails: iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in

Abstract

Personal Health Records (PHRs) allow patients to have full control over their health data. However, storage and sharing of PHRs still remains a difficult but necessary task, especially when health data is one of the major targets of cyber attacks worldwide. Searchable Encryption (SE) is a feasible solution for this problem and can be augmented by Blockchain to address some of its issues, such as verifiability. Therefore, SE using blockchain is a promising technologies to tackle the challenge of PHR storage and sharing. In this survey, we have explored the research works that use SE and blockchain technology for the same. The work starts with an introduction of cloud, searchable encryption and blockchain. Subsequently, we present a literature survey of the corresponding technologies. We then describe SE in detail and how it fits with blockchain. This is followed by description of noteworthy existing solutions for secure storage and sharing of PHRs. Even though there have been a number of surveys related to SE, none of them have surveyed the use of blockchain with SE or use of SE and blockchain in PHR sharing. The work concludes with a comparative study of these existing solutions and future scope in this direction.

KEYWORDS

blockchain, cloud computing, personal health records, searchable encryption, security, storage

1 | INTRODUCTION

Blockchain has been one of the most popular technologies over the past decade and has found applications in wide areas that are far from its first application–cryptocurrency. One of such areas is searchable encryption, which aims to provide an efficient and secure encryption mechanism to allow authorized user to search over the data encrypted by the data owner and stored on a remote storage such as the cloud. Searchable encryption itself has various areas of application, one of which is tied with health care. Storing Health Records of patients digitally is the preferred way by the hospitals because it allows efficient management and quick information retrieval. However, maintaining local storage for high volume data is non-trivial and expensive task, hence, outsourcing of the data to third party storage providers is the preferred way to store the records. But again, outsourcing the records as plain text to some storage service provider is a direct breach of privacy as the third party service provider cannot be trusted. Due to this, health records are first encrypted and then stored on the remote storage but at the expense of efficient information retrieval. Since direct query over encrypted data is not possible a major advantage of storing health records electronically is lost. This is where the role of searchable encryption comes into play. Additionally, cloud storage is the traditional medium that acts as remote storage provider, however, it has a

disadvantage that the remote server needs to be trusted against any dishonesty such as returning false data for saving costs and so on. To deal with this issue, many existing works have used blockchain as a mechanism to detect the dishonesty of the cloud server and in some cases the cloud sever has been completely replaced by the distributed blockchain network which stores the health records in the form of transactions. A transaction is immutable in the blockchain environment and the validity of any block can be easily verified which leads to the health records being tamper-proof. Next, we describe the various technologies mentioned above so that the subsequent discussion of application of SE and blockchain in healthcare becomes easier to understand.

1.1 | Cloud computing

Cloud computing is the practice of delivering computing resources such as computing, storage, databases, network, analytics, and so forth, through pay-as-you-go model over the internet without managing or setting up physical infrastructure. It is on demand provision of services by cloud service providers, such as "Amazon Web Service (AWS)", "Microsoft Azure", "Google Cloud Platform", and "IBM Cloud Services" to the users with a service level agreement(SLA) to ensure robust delivery of resources.

According to the "National Institute of Standards and Technology (NIST), USA", cloud computing can be defined as follows:

> *"A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."*

This cloud model consists of five necessary characteristics, four deployment models, and three service models. Cloud Computing is the cost-effective and reliable solution for those who want to reduce the hassle of physically setting up the whole infrastructure. It is highly scalable within no time, based on the consumption users can anytime increase the resources within a couple of minutes. Moreover, with cloud resources security is also taken care of by *Cloud Service Providers (CSPs)*.

Following are the characteristics of cloud computing:

1. **On demand services:** Users can pay for services according to need of there business and monitor those services to increase or decrease the resources according to requirement.
2. **Scalability and elasticity:** Cloud computing provides cost effective solution for resource utilization through scalable services which can be scaled up or down based on workloads.
3. **Multi tenancy:** Through cloud computing single program instance can be used to serve various user groups along with keeping data of users separate.
4. **High availability:** CSP's provides high availability of services through having backed up servers and resources in case of any failures.
5. **Resource pooling:** Resources can be better utilized through cloud service models, wherein CSPs can share resources among various clients. The resource allocation works on real time as a result no interruptions are there for end users.

Cloud computing can be divided into the following models based on the service provided by them.

1.1.1 | Infrastructure as a service

Infrastructure as a Service (IAAS) is also known as Hardware as a Service is the provision of cloud computing infrastructures like servers, storage, operating system, and network as a service. These services get available to users based on demand and can be scaled whenever required based on load. Through this service model companies can buys high processing power CPUs, memory, etc without spending much time and CapEx on setting up physical servers and infrastructure. Some of the IAAS offering examples are "Rackspace, Amazon Web Services (AWS) DigitalOcean, Cisco Metapod, Google Compute Engine (GCE), Linode, and Microsoft Azure".

1.1.2 | Platform as a service

Platform as a Service (PAAS) delivers cloud platform services, that is, runtime environment over which applications can be built, tested and deployed. PAAS services provides framework for developers to further develop their softwares on. These services includes both Infrastructure as a service and Platform as a service. Using PAAS offerings developers can directly concentrate on building and designing of applications without putting much efforts on runtime environment maintenance, software updates, underlined Operating System and infrastructure. PAAS providers, such as "Azure, Google App Engine, Force.com, and so forth." that provide Application frameworks, Databases, Programming Languages, and so forth.

1.1.3 | Software as a service

Software as a Service (SAAS) also known as "On Demand Software" offers cloud applications services which included end-to-end ready softwares to use. In these services cloud providers provide infrastructure, platforms and softwares and maintains all technical aspect of the offering such as data, middleware, storage and servers. Through SAAS services users can reduce the time spent on installing, updating and maintaining softwares on individual system, rather everything can be accessed and managed through central location on cloud. SAAS providers, such as Salesforce, Netflix, Google Workspace apps, and so forth. provide offerings like cloud based mail services, social networks, and collaborative documents sharing.

1.2 | Blockchain

Blockchain was first introduced in Satoshi Nakamoto's white paper[1] as the technology upon which Bitcoin (the first cryptocurrency) is based. Following that, researchers started exploring the applications of blockchain in other areas. In recent years, blockchain has found application in many areas, such as healthcare applications,[2-7] crowd sourcing system,[8] industrial Cyber-Physical Systems (CPS),[9] Internet of Drones (IoD),[10-14] Internet of Things (IoT)-enabled smart agriculture,[15-17] Internet of Vehicles (IoV) and Vehicular Adhoc Network (VANET),[18-21] Software-Defined Network (SDN),[22,23] Industrial Internet of Things (IIoT),[24,25] Internet of Everything (IoE),[26] smart-grid system,[27] supply chains,[28] Cognitive Internet of Things (CIoT),[29-31] and so on. The various applications of the blockchain technology are also illustrated in Figure 1.

Blockchain acts as a distributed ledger which records transactions that are immutable. However, it is not limited to financial transactions as in Bitcoin, it can record any other type of data that can be represented in the form of a transaction, such as the occurrence of an event of any type. The primary benefit of blockchain is that once a transaction has been recorded, it is immutable. To nullify the effect of a previous transaction, one will have to create a new transaction. These transactions are stored in the form of a block where a single block may contain an arbitrary number of transactions depending on the requirement and context in which the blockchain is being used. A blockchain network contains many independent nodes which interact with each other and make decisions based on an agreement algorithm, known as a consensus algorithm.

A block is nothing but binary data which is divided into two parts: (1) *header* and (2) *payload*. Header stores meta-data related to block one of which is the hash of the previous block and payload contains the data specific to a transaction. Since each block (except first one) has the hash of its predecessor, all the blocks together form of a chain and hence, the data structure got its name as blockchain. The first block which does not have any predecessor is known as *genesis* block. Each node participating in the blockchain network has its own copy of the blocks. The longest chain in the network is considered to be the valid chain by all the nodes. Any modification to a block requires that all the blocks ahead of it in the chain be modified. A malicious entity will have to alter the contents of majority of the nodes in the network by breaching into them which is almost impossible to do if the nodes are controlled by independent entities. Thus, when a malicious entity tries to alter the contents stored on a block, it gets detected very easily and other nodes involved in the maintenance of blockchain can take appropriate action.

When a node wants to add a new block to the chain it broadcasts this block to all the nodes in the network and each node verifies the integrity of this block before adding it to its local copy. Further, to decide upon the node that will be allowed to add the new block in the network, the consensus algorithm is used.

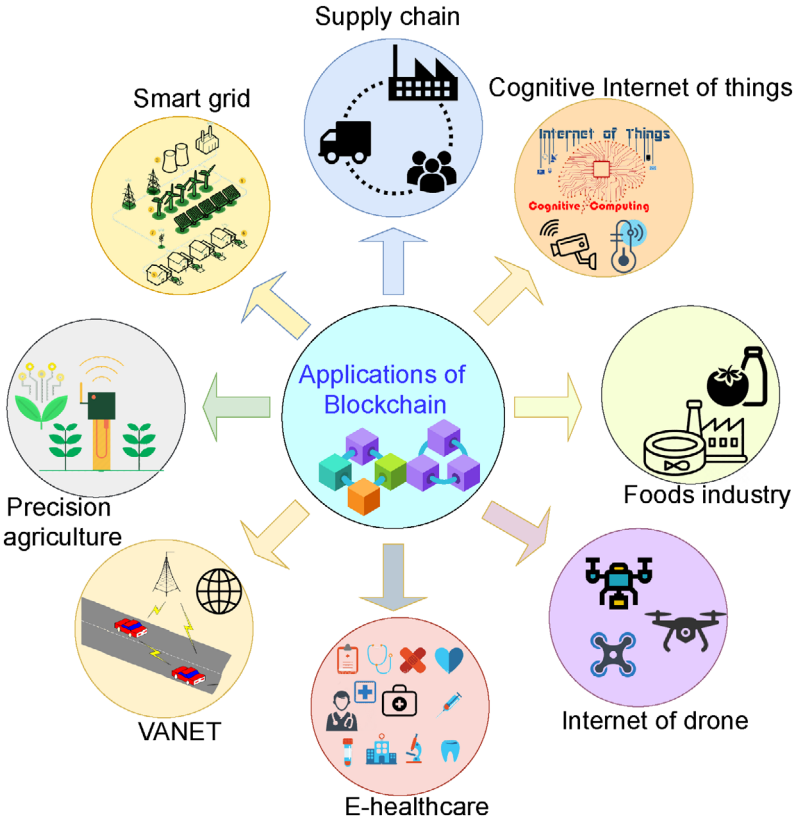

FIGURE 1 Various applications of blockchain.

1.2.1 | Consensus algorithms

Consensus is one of the fundamental problems associated with distributed systems. Many of the algorithms used in distributed system rely on the consensus algorithm for their correctness. Faulty nodes can also be of two types–Crash Faulty and Byznatine Faulty. Crash faulty process can crash in realtime however byzantine faulty processes can behave as a malicious entity that tries to prevent other precesses from reaching a consensus.

The popular ones are listed below:

1. *Proof of Work (PoW)*: In this algorithm,[32] a hard problem is selected and given to every node. The node that solves this problem earliest gets the chance to add new block to the chain. This algorithm is used in Bitcoin, however, it is very energy inefficient and hence alternative solutions to it are being considered now.
2. *Practical Byzantine Fault Tolerance (PBFT)* This algorithm[33] guarantees that a consensus is reached if the number of faulty nodes is at-most $\frac{n-1}{3}$, where n is the total number of nodes in the blockchain network. It is an energy efficient algorithm, however, it does not scale well to large number of nodes as the communication overhead is quite high.
3. *Proof of Elapsed Time (PoET)*: This algorithm was developed by Intel Corp. It follows a lottery system in which each node is randomly sleeps for a random period of time called wait time. The node with the smallest wait time wins the lottery and gets chance to add new block to the chain. This algorithm relies on secure computing as in the sleep algorithm must run in a tamper proof way so that a malicious node does not alter the randomness of sleep time for his own benefit.
4. *Raft*: Raft[34] is equivalent to an earlier class of algorithms called Paxos[35] but is easier to understand and is proven to be safe unlike Paxos. The algorithm selects a leader among the nodes in the network and the leader has a tenure after which election for a new leader are held. Once a leader is elected, it is responsible for coordination of all the nodes.
5. *Ripple Protocol Consensus Algorithm (RCPA)*: In the network running Ripple algorithm,[36] each server/node maintains a Unique Node List (UNL) which consist of the nodes that this node queries when determining consensus.

TABLE 1 Comparative study among various blockchain consensus protocols.[17]

Consensus protocol	Concept applied	Resource used	Application areas
PoW	Hashing	Computations	Ethereum[39] Bitcoin[1]
PoS	Digital signatures	Currency	SnowWhite[40] PeerCoin[41] Ouroboros[42]
PBFT	Voting	No resource	Tendermint[43]
Ripple	Voting occur in multiple rounds	No resource	XRP ledger[44]
PoET	Generation of random numbers	Intel SGX	Hyperledger sawtooth[45]

Each node runs the Ripple algorithm every few seconds and if 80% of a node's UNL agree on a transaction then it is applied.

6. *Proof of Stake (PoS)*: Under this algorithm,[37] each node stakes some amount of cryptocurrency in order to become a candidate for the validator in the network. An algorithm then selects one node out of the candidate set based on a combination of amount of cryptocurrency staked along with some other factors and assigns it as the validator for the next block. The validator then validates the next block and adds it to the network in exchange for a fee.
7. *Proof of Vote (PoV)*: This algorithm[38] is used in case of consortium blockchain 1.2.2. There are four major roles in this algorithm-"Commissioner", "Butler", "Butler-candidate" and "Ordinary User". The Commissioner is part of the consortium committee of the network and verify each block that is generated in the network. Butlers specialize in production of blocks and are elected by the commissioner for a specific tenure which expires after a certain period. The next role–Butler-candidate, participates in the election for becoming Butler. Ordinary users on the other hand can join or exit the network anytime without any authorization.

A comparative study among various blockchain consensus protocols is summarized in Table 1.

1.2.2 | Types of blockchain

Based on the relation between the nodes participating in the blockchain network, we can categorize blockchain into three types.

1. **Private blockchain:** In this type of blockchain all the nodes belong to a single organization and any external node cannot join the network without getting permission from the organization.
2. **Public blockchain:** In Public blockchain the nodes can belong to different individuals and organizations. The most popular blockchain network–Bitcoin, is an example of public blockchain.
3. **Consortium blockchain:** Consortium blockchain is a mixture of the first two types. It consists of a core set of nodes that have exclusive rights to validate a new transaction and add a new block to the chain. All other participants can only submit transactions and necessarily participate in block forwarding.

1.2.3 | Smart contract

As described earlier, blockchain stores data in the form of transactions that are immutable once committed. This data need not be passive, instead, it can also be active such as some piece of code. Because of the immutable property of a transaction, the code, cannot be modified by any malicious entity unless it controls the majority of the nodes in the blockchain. Further, this code runs only when certain conditions are met. These conditions are set by the creator of the contract and are embedded in the code itself. It is because of Smart Contracts that blockchain can be used for a wide variety of applications. Smart contracts originally were introduced by Ethereum[46] which is a public blockchain platform. Now there are even alternatives to smart contracts that are more efficient and flexible, one of those is the Hyperledger Sawtooth which uses Transaction Processors instead of Smart Contracts to achieve the same functionality.

1.2.4 | Advantages of blockchain

Blockchain has several advantages over cloud that are listed below:

1. **Transparency:** All the nodes participating in the blockchain network hold a copy of the blocks which makes the transactions transparent to all the nodes.
2. **Security:** Once a transaction is committed to a block, it cannot be mutated by anyone, that is, it is immutable. This property of blockchain ensures that the data stored in the blocks is secure from any malicious entity.
3. **Decentralization:** Blockchain is a distributed network in which each participating nodes has equal power. There is no central authority hence a central point of failure is eliminated.
4. **Traceability:** All the blocks are linked to in the form of a chain, this makes it very easy to trace the transactions that have occurred up to any depth. This property of blockchain which makes it very easy to trace the previous transactions has its applications in many areas a major one of which is supply chain management.
5. **Immutability:** The immutable property is achieved on the transactions that need to be agreed as well as shared across the blockchain network. Now, if the transactions are connected to the blockchain, it becomes difficult to modify or erase them.[47]

1.3 | Searchable encryption

With ubiquitous presence of cloud storage, to mitigate the risk of data leakage one would prefer to encrypt the data and then upload it to a cloud server. However, this results in loss of many desirable functionalities out of which a prominent one is search over the stored data. Searchable encryption mitigates this loss by encrypting data in such a way that one can search for keywords in the stored data without actually decrypting it. Research in the SE domain was pioneered by Song et al.[48] in the year 2000. They proposed the first searchable encryption scheme which used a linear search technique in which the entire encrypted documents needs to be processed during the search. However, it is inefficient when the size of document and document sets becomes large. A second approach was suggested in the work by Song et al. and later used by many subsequent works, which creates an index over the encrypted files that is then used during search to look for the files in which this keyword occurs. The latter approach is significantly faster than the former one and was therefore studied in more detail later on by other researchers. The original linear scan technique by Song et al., being very naive and inefficient is not considered for further study. Later, other forms of searchable encryption such as homomorphic encryption based searchable encryption, Fuzzy Keyword based search and Public Key encryption using conjunctive keyword search were also introduced. However, index based searchable encryption remains the most efficient one and a lot of work has been done in this direction.

Searchable encryption can be widely categorized as *static* and *dynamic*. *Static* SE refers to the case where the files are encrypted as a batch and after encryption files cannot be added or deleted without requiring a re-encryption of the modified set of files. *Dynamic* on the other hand allows efficient addition and deletion of files without requiring a re-encryption. SE can also be categorized as *Symmetric* or *Asymmetric* based on the encryption type that lies at its core. Moreover, depending on the number of keywords that can searched, SE is categorized as *Single-keyword* or *Multi-keyword*. Multi-keyword search is obviously more desirable, however, it is difficult to construct schemes supporting it without leaking more information.

1.4 | Personal health records

Personal Health Record or PHR is a digitized form of a patient's medical record that contains the entire medical history of the patient and is maintained by the patient. Ideally, a patient can grant access to other users such as medical institutions and insurance companies as per his or her requirements. PHRs make it easier to share and access medical history of a patient in an efficient manner that saves time and cost for both the patient and the medical institution. In case of an emergency PHRs can be very helpful as the previous medical history is often needed before giving any kind of treatment. PHRs often get confused with "Personal Health records (EMRs)" and "Personal Health Records (PHRs)". PHR and PHR differ in the sense that the former one is maintained by the patient himself and the latter is created and maintained by the hospital or medical institute. However, the difference between PHR and EMR can be a bit subtle. It can however be

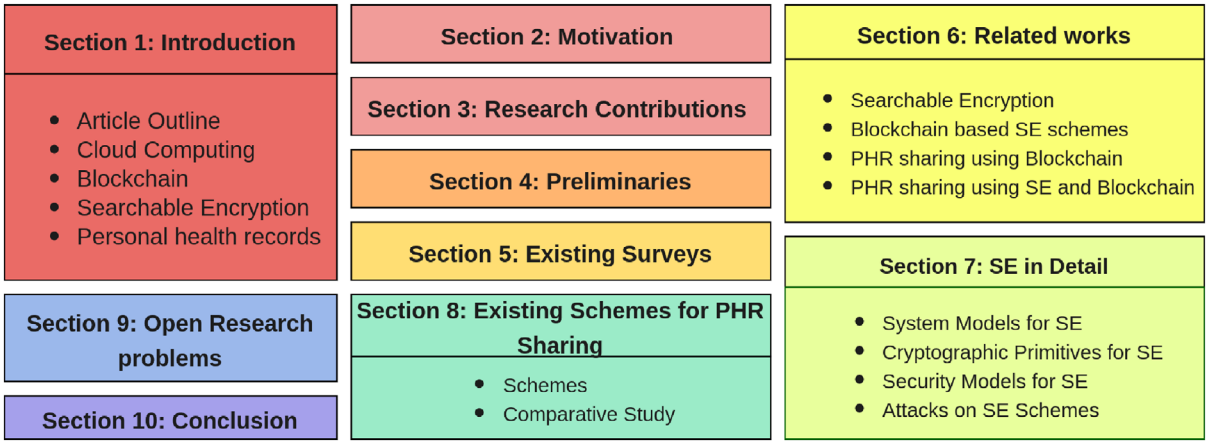

FIGURE 2 Paper outline.

clarified by considering the difference between the terms 'medical' and 'health'. An EMR is a narrow view of patient's medical history created by a doctor for diagnosis of the current disease. An PHR on the other hand is comprehensive medical history of the patient related to diagnosis of current disease as well as that of past diseases. However, the two terms are many times used interchangeably.

1.5 | Article outline

The rest of the article is organized as follows. We have also provided an outline of the entire paper in Figure 2.

1. In Section 2, we describe the motivation behind this paper and the research contributions made in this article in Section 3.
2. In Section 4, we discuss some mathematical primitives that are necessary to discuss the concept of SE and its application in healthcare applications.
3. In Section 5, we discuss the existing surveys in this area.
4. A detailed literature survey on the SE and blockchain, and their application in PHR storage and sharing has been provided in Section 6.
5. In Section 7, we discuss various system models, cryptographic primitives, security models, possible attacks, and also threat model on SE.
6. Various existing state of art solutions to PHR sharing using blockchain technology are discussed in Section 8.
7. We also discuss some open research problems associated with the searchable encryption on cloud storage for healthcare applications in Section 9.
8. Finally, the paper has been concluded in Section 10.

2 | MOTIVATION

Healthcare systems that are currently in use are mostly proprietary and hence do not allow efficient sharing of data with other institutions. Further, these systems are centralized and are vulnerable to variety of attacks and thus raise concern about the privacy of the patient. A simple example is the case where an employee managing the health data of patients is himself malicious. No efficient security mechanism exists for such a case other than deterrence of a legal action against the employee. However, blockchain technology along with searchable encryption has provided a solution for storing PHRs that addresses the problems above. By using the immutability property of blockchain the PHRs, can be made tamper-proof and since there is no third party involved all sort of internal attacks are thwarted. Further, to maintain the privacy of the patient the PHRs can be encrypted before storing them on blockchain and searchable encryption techniques can be

deployed to allow authorized users to search over encrypted data. Given the importance of work in this direction, we felt the need for a survey of existing solutions proposed by various researchers so that future researches in this direction can be carried out more easily.

3 | RESEARCH CONTRIBUTIONS

In this survey, we have mainly attempted to present the confluence of SE and blockchain and their application in PHR sharing. We first briefly describe some existing surveys and then show how SE and blockchain together can be used to create a robust system for storage and sharing of data. We discuss the system models, cryptographic primitives, security models and attacks that can happen on such a scheme. Following this, we describe the existing schemes based on SE and blockchain for PHR storage and sharing. We also provide a comparative study of these schemes. Finally, we present some open research problems that need to be solved for development of better PHR storage and sharing systems.

4 | PRELIMINARIES

In this section, we discuss the following basic mathematical preliminaries that are essential to describe and analyze the security protocols.

1. **Pseudo random generator:** Suppose f is a function from $\{0,1\}^n \rightarrow \{0,1\}^{l(n)}$, where it takes an n-bit string $x \in \{0,1\}^n$ as an input and produces an $l(n)$-bit string, say $y = f(x) \in \{0,1\}^{l(n)}$ as an output. Now, f is called a pseudo random generator (PRG), if the following are valid:[49]

   - *Expansion*: For every n, $l(n) > n$. In other words, the length of the output must be greater than that of input.
   - *Indistinguishability*: For any "probabilistic polynomial time distinguisher, $\mathcal{D}$", there is a negligible function such that

   $$|Pr[\mathcal{D}(G(s)) = 1] - Pr[\mathcal{D}(r) = 1]| \leq negl(n).$$

   Here, the seed $s \in \{0,1\}^n$ chosen uniformly randomly and $r \in \{0,1\}^{l(n)}$ chosen uniformly randomly. Moreover, the probabilities are considered over the "random coins used by $\mathcal{D}$ and the choice of r and s". In addition, the function $l(\cdot)$ is known as the "expansion factor of G".

   A pseudo random generator is then considered as a *deterministic* algorithm that takes a "short truly random seed" and then stretches it into a "long string that is pseudo random".
2. **Pseudo random function:** Suppose that $Func_n$ is the set of all functions from $\{0,1\}^n \rightarrow \{0,1\}^n$ and F: $\{0,1\}^* \rightarrow \{0,1\}^*$ is a keyed function that is also length preserving. F is called a pseudo random function (PRF), if there exists a negligible function *negl*, for all probabilistic polynomial time distinguishers $\mathcal{D}$, such that the following property holds:[49]

   $$|Pr[D^{F_k(\cdot)}(1^n) = 1] - Pr[D^{f(\cdot)}(1^n) = 1]| \leq negl(n)$$

   Here, key k is chosen uniformly randomly from the set $\{0,1\}^n$ and f is chosen uniformly randomly from $Func_n$.

   In cryptography, the pseudo random functions become a very important building block for a number of various cryptographic constructions. For instance, there are efficient primitives known as *block ciphers* that are assumed to act as pseudo random functions.
3. **Bilinear maps:** Let G_1 and G_2 represent an "additive cyclic group" and a "multiplicative cyclic group" with a prime order p. Assume that g_1 is a generator of G_1. A mapping $e : G_1 \times G_1 \rightarrow G_2$ is called a "bilinear map", if it fulfils the following properties:[50,51]

   - **Bi-linearity**: $\forall u, v \in G_1$ and $\forall a, b \in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$ holds, where $Z_p = \{0, 1, 2, \ldots, p-1\}$.
   - **Non-degeneracy**: $e(g_1, g_1) \neq 1$, where the identity in G_2 is 1.

- **Computability**: $\forall u, v \in G_1$, $e(u, v)$ is efficiently computable.

The map e is called as an admissible bilinear map if $e(g_1, g_1)$ generates G_2 as well as e is also efficiently computable. We have the following computational problems in the bilinear pairing setting.

Definition 1 (Bilinear Diffie-Hellman Problem (BDHP)). If G is a "multiplicative cyclic group that is generated by g of prime order p", and given a tuple (a, g^a, g^b, g^c) for $a, b, c \in Z_p$, compute $e(g, g)^{abc}$.

Definition 2 (Weak Diffie-Hellman Problem (WDHP)). Assume that G is a "multiplicative cyclic group generated by g of prime order p". If (g, h, g^a) for $a \in Z_p$ are given, compute h^a. The problem is called a weak Diffie-Hellman problem due to the fact that $e(g^a, h) = e(g, h)^a = e(g, h^a)$.

4. **Collision resistant hash functions:** It is a "cryptographic function" that takes an input of a variable length and produces an output of a fixed length, called a hash value or message digest. Mathematically, it can be expressed as a mapping as h: $A \rightarrow B$, where the set $A = \{0,1\}^*$ consists of all the strings of variable length and the set $B = \{0,1\}^{l_h}$ consists of the strings of fixed length l_h.[52]

The formal definition of a one way hash function is as follows.[53]

Definition 3 (Collision-resistant One-way Hash Function). A collision resistant one-way hash function h: $\{0,1\}^* \rightarrow \{0,1\}^{l_h}$ is a deterministic function which takes an arbitrary length binary string $\alpha \in \{0,1\}^*$ as input and returns a fixed-length binary string of l_h bits $\beta = h(\alpha) \in \{0,1\}^{l_h}$ as output. The advantage $Adv_{\mathcal{A}}^{HASH}(t_p)$ of an adversary $\mathcal{A}$ in finding the hash collision in polynomial time t_p is given by

$$Adv_{\mathcal{A}}^{HASH}(t_p) = Pr[(\alpha, \alpha') \leftarrow \mathcal{A} : \alpha \neq \alpha' \text{ and } h(\alpha) = h(\alpha')]$$

where $Pr[X]$ denotes probability of random event X, and $(\alpha, \alpha') \leftarrow \mathcal{A}$ represents that the pair (α, α') is randomly picked by $\mathcal{A}$. By an "(ψ, t_p)-adversary $\mathcal{A}$ attacking the collision resistance of $h(\cdot)$", we mean that the "runtime of $\mathcal{A}$ is bounded by t_p and that $Adv_{\mathcal{A}}^{HASH}(t_p) \leq \psi$."

A hash function has the following properties:[52]

- One way hash functions are easy to compute, that is, given an input $\alpha \in A$ it is easy to compute $h(\alpha) \in B$, but computing the inverse of the function is computationally infeasible. Thus, given the hash value $h(\alpha)$ of any input α, the computation of the original input α is computationally infeasible. This property is known as *pre-image resistant*.
- For any given $\alpha \in A$, there exists no feasible way in polynomial time to find out another input $\alpha' \in A$ such that $h(\alpha) = h(\alpha')$. This property is called *weak collision resistant property*.
- It is computationally infeasible to find any two inputs $\alpha, \alpha' \in A$, such that $\alpha \neq \alpha'$ with $h(\alpha) = h(\alpha')$. This property is known as *strong collision resistant property*.

Some examples of secure and collision-resistant hash functions include "Secure Hash Algorithm (SHA-1) and its variants SHA-224, SHA-256, SHA-384, and SHA-512.[54]"

5. **Bloom filters:** A Bloom filter[55] is a datastructre that is used to test set membership of an element. It can give false-positive results with some probability but never False-negatives. It consists of a set of n elements $S = \{s_1, \ldots, s_n\}$ represented by array of m bits. All the bits of array are initially 0. A set of r independent hash functions $h_1, \ldots, h_r$ are used by the filter where $h_i : \{0,1\}^* \rightarrow [1, m] \forall i \in [1, r]$. The bits at position $h_1(s), \ldots, h_r(s)$ of the array are set to 1 for each element $s \in S$. We can set a location to 1 multiple times, however, only the first one is significant. Further, to determine if set S contains an element a, we check whether all the bits at positions $h_1(a), \ldots, h_r(a)$ are 1 or not. If they are all 1s, then $a \in S$. With some probability false positive may occur because some other element other than a may also set the same location.
6. **Standard blockchain model:** To formally analyze security schemes that use blockchain, Kosba et al.[56] proposed the standard blockchain model. It treats blockchain as a trusted entity that is guarantees correctness and availability but cannot be trusted for privacy. This is because all the transactions can be read by every participating node thus privacy cannot be guaranteed by the blockchain unless the transactions are encrypted in some form. However, without the consensus of majority of nodes a transaction cannot be modified, therefore, cannot be modified by them.

TABLE 2 Abbreviations.

Abbreviation	Full form
ABKS	Attribute based encryption with keyword search
DO	Data owner
CS	Cloud server
DU	Data user
P	Patient
PEKS	Public key encryption with keyword search
PHR	Personal health record
PRES	Proxy re-encryption with keyword search
SE	Searchable encryption
SSE	Searchable symmetric encryption

To make this article easier to read, we have provided a list of abbreviations used in this paper in Table 2.

5 | EXISTING SURVEYS

Over the years, several survey papers over searchable encryption have been published. We present the findings of these survey papers in this section and also compare them with the our survey.

1. **Bosch et al.**[57] **(2014):** This survey gives an overview of the works in searchable encryption done till 2014. Its target audience consists of non-specialists and researchers who only have a basic background in security. For this reason the authors have omitted security proofs and other mathematical details. They have categorized the different SE schemes under four categories–(1) "*Single Writer/Single Reader*", (2) "*Single Writer/Multiple Reader*", (3) "*Multi Writer/Single Reader*", and (4) "*Multi Writer/Multi Reader*". Here, writer refers to data owner who generates the data to be secured and reader refers to data users. This is a detailed survey of SE schemes, however, it does not include any SE schemes that make use of blockchain. The main reason for this is that till 2014 there were hardly any works in the direction of SE and blockchain.
2. **Wang et al.**[58] **(2016):** This survey is mainly focussed on two of the standard SE techniques–"*Searchable Symmetric Encryption (SSE)*" and "*Public Key Encryption with Keyword Search (PEKS)*". The authors have compared the schemes based on index size, search time, security and dynamism (for SSE), and security assumptions (for PEKS). A basic security background is suffice to read this paper.
3. **Poh et al.**[59] **(2017):** This is a detailed survey that gives a good insight into the workings of SE schemes. The authors have categorized the existing schemes based on principals involved, operations, security models, entity setups, query functionalities, cryptographic primitives, scheme structure, performance and characteristics. It is targeted towards readers having intermediate background in security. It indeed covers most of the literature of SE, however, it does include schemes that make use of blockchain.
4. **Zhang et al.**[60] **(2018):** In this survey, the authors explore the application of SE in healthcare industry. They have presented four scenarios in which SE can be used to secure health records, namely, owner as reader/writer, Single reader/Single writer, One writer/many readers, authorization delegation. Reader and Writer refer to health data producer and health data consumer. They also provide an overview of the standard SE techniques, namely, "*Searchable Symmetric Encryption (SSE)*", "*Attribute Based Encryption with Keyword Search (ABKS)*", "*Public Key Encryption with Keyword Search (PEKS)*" and "*Proxy Re-encryption with Keyword Search (PRES)*". This work, however, lacks a discussion and comparison of existing SE schemes for healthcare and also does not describe schemes that use blockchain. To read this work we recommend an intermediate security background.
5. **Pham et al.**[61] **(2019):** In this work, the authors have provided a general overview of the system architecture involved in a SE scheme. They have categorized the schemes, based on the type of search, into–(1) "*Keyword Search*", (2) "*Regular Expression Search*", and (3) "*Semantic Search*". On the basis of security levels they have categorized the schemes

TABLE 3 Existing surveys.

Year	Survey	Scope	Requirement
2014	Bosch et al.[57]	SE	Basic security background
2016	Wang et al.[58]	SSE and PEKS	Basic security background
2017	Poh et al.[59]	SE	Intermediate security background
2018	Zhang et al.[60]	SE for healthcare	Intermediate security background
2019	Pham et al.[61]	SE system architecture	Basic security background
2022	Andola et al.[62]	SE with cloud storage	Basic security background

into–(1) *Somewhat secure*, (2) *Semi secure*, (3) *Secure* and (4) *Fully Secure*. This work also explores the application areas of SE such as healthcare, law enforcement and text editors. The target audience for this work consists of readers with basic background in security. However, this work too does not include SE schemes that use blockchain.

6. **Andola et al.[62] (2022):** This work is geared towards survey of application of SE in storage in cloud. On the basis of the search type the authors have classified SE on the cloud into "sequential search, index-based search, rank based search, fuzzy keyword search, conjunctive keyword search and access-control-based search". Further, they have also discussed cryptographic primitives used in these categories of SE. However, on downside, no SE schemes using blockchain have been covered in this survey. We note that this paper requires reader to have basic security background.

Finally, the summary of all the above existing surveys with their scopes and requirements is provided in Table 3.

6 | RELATED WORKS

6.1 | Searchable encryption

Song et al.[48] proposed the first searchable encryption scheme in 2000. Their scheme encrypts each word in the file using a deterministic algorithm. Later during search operation, a linear scan is performed over the encrypted file to search for keywords. However, it is not efficient enough for most practical purposes because of linear search time in terms of number of words in the encrypted document.

To overcome these limitations, Goh et al.[63] in 2003 proposed an alternative solution using an index based approach. They provided construction of a secure index in which the data owner generates a secure index over the files that are to be encrypted. This secure index is uploaded to the cloud server along with the data files which are encrypted using some standard symmetric encryption algorithm such as AES. They also introduced a security model known as "semantic security against adaptive chosen keyword attack (IND-CKA) and its stronger variant IND2-CKA." In IND-CKA, A challenger C gives an adversary A two documents V_0, V_1 such that their lengths are equal, along with an encrypted index. The adversary A is then challenged to determine which document is encoded in the index. Hardness of deducing whether the index is of V_0 or V_1 implies the hardness of using the index to find a word such that V_0 and V_1 do not have it in common. In the same work they also proposed Z-IDX, which is an efficient IND-CKA secure index construction, using "Pseudo Random Functions" and "Bloom Filters". To make search time linear in terms of number of documents, a bloom filter is created for each document in the proposed scheme.

Searchable encryption was extended to public key setting in 2004 by Boneh et al.[64] who took a mail server as the reference for their scheme and proposed the first public key encryption with keyword search (PEKS) scheme. PEKS uses Identity based encryption (IBE)[65] with the keyword as the identity. The sender encrypts his/her message using any standard public key system such as RSA or EC and then appends the PEKS of each keyword $w_1, w_2, w_3, \ldots, w_m$ with it. To obtain the PEKS, a publicly known string is encrypted using public key associated with keyword as identity. The ciphertext obtained is of the form shown below:

$$E_{K_{pub}} \| C_1 = PEKS(K_{pub}, w1) \| \ \ldots \ \| C_m = PEKS(K_{pub}, w_m)$$

To search for a keyword the receiver derives a secret key from the keyword it wants to search using the master key. This secret key is the trapdoor that is sent to the server, which then attempts to decrypt the IBE cipher-texts. If the decryption results in a publicly known string, then it is successful and it can be concluded that the attacked encrypted message contains the keyword.

Previous works in SE lacked a robust security model so in 2005 Chang et al.[66] proposed a real-ideal simulation model which is now widely used in subsequent works over searchable encryption.

Later, in 2006, Curtmola et al.[67] proved that IND-CKA and IND2-CKA security models were not strong enough and introduced two new security models for SSE, namely, *non-adaptive model* and *adaptive model* and also provided two schemes: SSE-1 and SSE-2 against the proposed security models respectively.

Next major breakthrough in SE was in 2012, when Kamara et al.[68] extended the work of Curtmola et al. and introduced "*dynamic searchable encryption (DSE)*" scheme which allows addition and deletion of new files efficiently.

Even though a lot of research on SE had been done by that time, Islam et al.[69] pointed out that there had been no proper study of attacks on SE schemes. Their work showed that various attacks were possible due access pattern disclosure in SE schemes and emphasised on the need of *forward security* on SE schemes. Forward security refers to the notion that an adversary should not learn anything new when a new file is added Based on this Stefanov et al.[70] presented the first forward secure SE scheme in 2013. However, due heavy dependency on client side computation, its practical applications were limited. Advancing in the direction of efficient searchable encryption Kamara et al.[71] proposed a Parallel and "dynamic searchable encryption scheme". In this they have used a *Red-Black tree* to create a *keyword Red Black (KRB)* tree data structure.

Cash et al.[72] in 2015 provided formalization of leakages in SE schemes along with possible attacks due to these leakages. They classified SE schemes in four levels based on the leakages. Prior to that, no other work had focused on the study of attacks on SE schemes. Later in 2016 the notion of *verifiability* was introduced by Bost et al.[73] by improving Stefanov et al. scheme. Verifiability refers to whether the results returned to the end user by the cloud server are correct or not. For example, during lack of verifiability a server can return partial results to the end user to save computation costs.

Next, in 2017 the first scheme to support *backward privacy* was introduced by Bost et al.[74] Backward privacy refers to the leakage that can occur after deletion of a file from the server that was previously queried. Simultaneously, a DSE scheme with focus on data deletion in SE schemes was proposed by Kim et al.[75] In the following year, an efficient, forward secure and parallelizable DSE scheme was proposed by Etemad et al.[76] which has performance on par with previous non-forward secure DSE schemes. However, a security flaw in backward privacy for a special case was discovered in Etemad's scheme in 2022 by Watanabe et al.[77] which was also patched by them.

Based on these works we observed that there is a trade-off between security and efficiency in searchable encryption schemes. These trade-offs have been presented in the work by Ashrov et al.[78] Many of the subsequent works[75,79-85] have focused on providing schemes that guarantee forward and backward secrecy but also efficiency at the same time. Some of the recent works have tried to implement SE in mobile devices for ease of users.[86] Many of the existing searchable encryption schemes use the public-key cryptography that rely on bilinear pairings. However, recent works[87,88] are pairing-free, and they do not suffer from computational cost problems that arise due to use of bilinear-pairings. Additionally, there are schemes[89-91] that make use of attribute-based encryption technique to provide a fine-grained access control over the encrypted data.

6.2 | Blockchain-based SE schemes

As discussed earlier, for practical applications of searchable encryption, there is a need to verify whether the server returns correct results or not. A malicious server could return the wrong results in order to save costs and still receive full payment from the user. Many of the works since 2018 have incorporated blockchain into SE schemes to provide verifiability. In 2018, Hu et al.[92] proposed a blockchain based solution in which the server is replaced by a blockchain network. By utilizing smart contracts they provided a solution in which the results returned to the user were always correct and all the parties received the correct payment. Around the same time Li et al.[93] also proposed a solution to the same problem using blockchain but without eliminating the central server. Continuing in this direction, in 2020, Guo et al.[94] suggested a verifiable and forward secure encrypted search which also leveraged the blockchain technology. In the same year Du et al.[95] also published a similar work that provided forward as well as backward secrecy along with verified results. In addition, there are also some works[96] that combine blockchain and attribute based encryption to provide escrow-free schemes.

6.3 | PHR sharing using blockchain

After the success of Bitcoin which was proposed in 2008,[1] the researchers began to find the application of blockchain in other areas, one of which was healthcare. An early work for application of blockchain in healthcare was proposed by Azaria et al.[97] in 2016. It focused on taking advantage of transparency and immutability of blockchain share data in a way that is integrity and verifiability is maintained. Another work proposed by Yue et al.[98] focused on an app called "Healthcare Data Gateway (HDG)" which uses blockchain to let users exercise full control over their medical records. Subsequently, one the early works that specifically targeted secure sharing of health records by using immutability and autonomy of blockchain was proposed by Xia et al.[99] in 2017. Next, Fan et al.[100] improved upon the scheme proposed by Xia et al. by providing efficient access and retrieval mechanisms and addressed the problems related to privacy originated by sharing health data with third parties.

6.4 | PHR sharing using SE and blockchain

Zhang et al.[101] in 2018 proposed a scheme for sharing of health records using SE and blockchain. Their scheme, called "Blockchain-based Secure and Privacy-preserving PHI sharing (BSPP)", that makes use of both private blockchain and consortium blockchain along with PEKS. Next in 2021, Wang et al.[102] made use of a decentralized storage called *Inter-planetary File System (IPFS)* to store the PHRs. However, these schemes rely on bi-linear pairing which increases their computation time. In the past few years, there have been works[103,104] that make use of proxy re-encryption based SE for health record sharing. Yet again, these schemes rely on bilinear pairings thereby requiring higher computation power. On the basis of SSE there have been lesser number of for health record sharing such as that of Chen et al.[105] and the one proposed by Tang et al.[106] We have described these scheme in detail later in Section 8.

7 | SEARCHABLE ENCRYPTION IN DETAIL

7.1 | System models for SE

Here we discuss the different types of system models employed by researchers for searchable encryption. Broadly, there are two types of systems that have been used in the literature. First one is the traditional SE system model which involves a data owner, cloud server and data user. Second one additionally uses blockchain to provide more features such as verifiability.

In the following, we describe both models: (1) SE without blockchain and (2) SE with blockchain, and the benefits of using SE with blockchain over SE without blockchain.

1. **SE without blockchain:** Figure 3 shows the general system model for searchable encryption when blockchain is not involved. There are three entities in this model–(1) Data Owner (DO), (2) Data User(DU) and (3) Cloud Server (CS). First, the data owner generates an encrypted index and encrypted files from the plaintext files and sends them to the could server. The encrypted index is used by the cloud server to search for documents containing a given keyword. To search for a keyword, the data user requests the data owner to for a token/trapdoor, which it then sends to the cloud server. The cloud server upon receiving the trapdoor searches for any document containing the keyword hidden in the trapdoor and returns the results to the user.
2. **SE with blockchain:** By utilizing the immutability property of blockchain, searchable encryption schemes which have intrinsic verifiability can be constructed. There are multiple ways in which blockchain has been used along with searchable encryption. However, here we have shown two of those ways. Figure 4 shows the first method in which the cloud server is entirely replaced with blockchain. The encrypted documents as well as the encrypted index are all stored on the blockchain. Hu et al.[92] were the first to use this technique.

   The second method has been show in Figure 5. In this method, the data is stored on a storage server which can be cloud or a distributed file system such as IPFS.[107] Blockchain is used to store metadata such as encrypted index. This reduces the overhead of processing large files on the blockchain. The data owner first encrypts the files/documents and uploads them to the storage server. The storage server provides him the unique file identifiers for the uploaded files. The DO then uses these identifiers and the original files to create an encrypted index. This encrypted index is then uploaded to the blockchain.

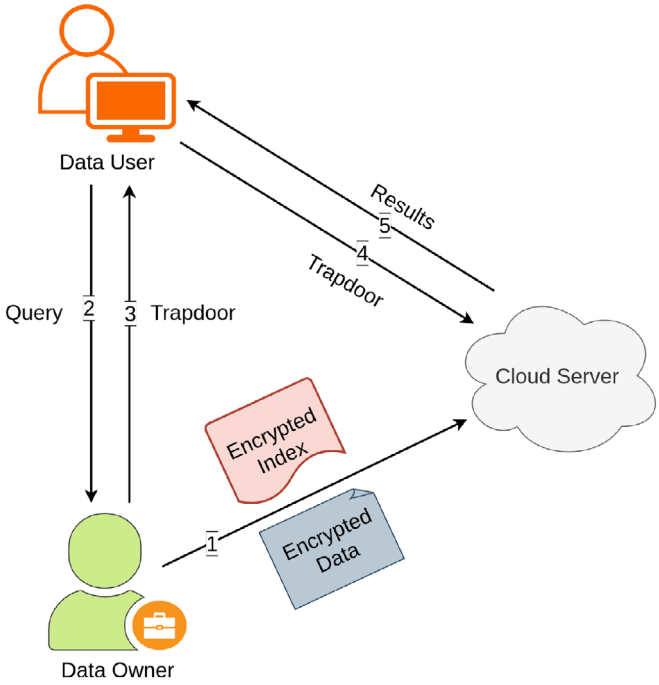

FIGURE 3 Searchable encryption without blockchain.

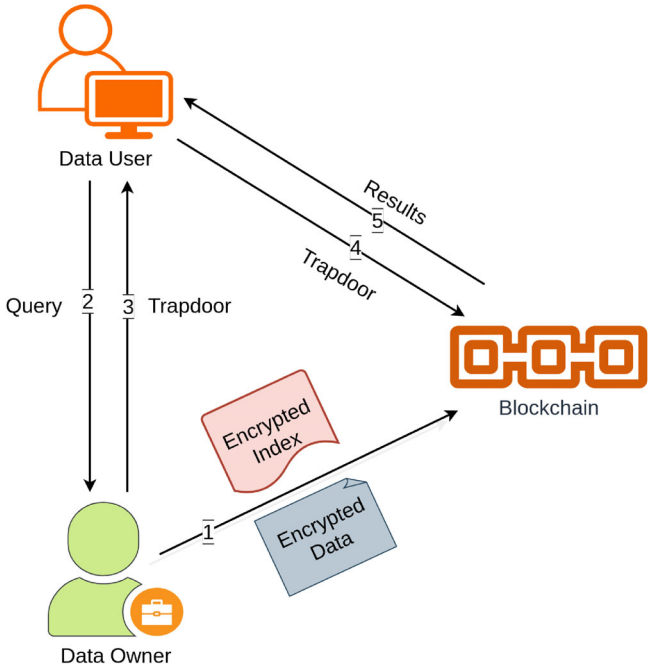

FIGURE 4 Searchable encryption with blockchain (type 1).

When a user wants to search for a keyword, he requests the data owner for a trapdoor corresponding to his/her query. The user then sends this trapdoor to the blockchain to obtain the file ids which contains the keyword. Finally, the user uses the received file ids to download the files from the storage server. A typical use of this technique can be seen in the paper by Guo et al.[94]

7.2 | Cryptographic primitives for SE

Based on the cryptographic primitives used, SE schemes can be categorized into two categories:

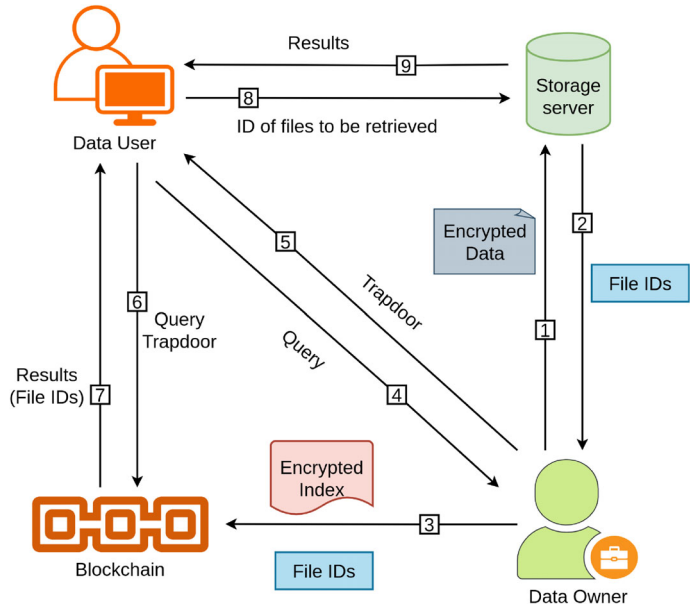

FIGURE 5 Searchable encryption with blockchain (type 2).

1. **Searchable symmetric encryption (SSE):** This is the most widely studied form of searchable encryption pioneered by Song et al.[48] in 2000. It relies on symmetric key cryptography, hence the name. There are many variants of SSE such as sequential search based, inverted index based and tree based. However, the most popular one is inverted index based due its simplicity and sub-linear search time.
2. **Public key encryption with keyword search (PEKS):** This searchable encryption technique allows keyword search over data encrypted using asymmetric cryptography. It was first introduced by Boneh et al.[64] to allow encrypted search over mails sent to a user. In usual case, multiple people can send mails to the user by encrypting them using his/her public key and the user will have to encrypt each of the mails to filter out relevant content. However, PEKS allows the user to filter his/her mails without actually decrypting them, thereby enabling the user to decrypt and read only those mails that he/she is interested in.
3. **Attribute based encryption with keyword search (ABKS):** Attribute based encryption allows encryption of data in a manner such that only a person who has attributes that satisfy certain set of conditions (know as a policy) can decrypt it. For example, in an organization, attributes can be given to employees on the basis of their rank in the organization and the policy can be made in a way that higher ranked employees can read the encrypted messages sent to lower ranked but not vice-versa. The first attribute based encryption scheme was proposed by Zhao et al.[108] in 2011.
4. **Proxy re-encryption with keyword search (PRES):** Proxy re-encryption is a technique which allows an entity, called proxy, to re-encrypt a message encrypted using public key of original recipient, into a message which can only be decrypted by a person chosen by the original recipient. In short, the original recipient delegates his decryption capability to another person on his behalf. To re-encrypt the original message, the proxy entity needs a re-encryption key and the public key of the delegate. The re-encryption is generated using the secret key of the original receiver but cannot be used to disclose the original receiver's secret key. An example where this is useful is when an official of an organization wants to delegate some of his messages to his/her secretary.

   This technique was combined with encrypted keyword search for the first time by Shao et al.[109] In the use case presented by them, the proxy entity was a mail server which could re-encrypt the message intended for a recipient into message for a delegate.

7.3 | Security models for SE

The first formal threat model for SE schemes was proposed by Goh.[63] He proposed semantic security against chosen-keyword-attack through a threat model which he termed IND-CKA. This model aimed to cover the notion of

security in which an adversary cannot deduce any partial information about the contents of a document from its index other than what it already knows from external sources. IND-CKA provided security only when the size of all the documents involved in creation of index is equal. This shortcoming was overcome by proposing a stronger model IND-CKA2 in which the size of the documents need not be equal. However, none of the two security models required trapdoors to be secure and therefore failed to model real world scenario for searchable encryption.

In 2006, Curtmola et al.[67] provided an alternative security model for SE, which since then has become a standard. They proposed two security models: (1) *Non-Adaptive Semantic Security* and (2) *Adaptive Semantic Security*. The setting is that of a client and server, where the client generates a secure index and upload it to server along with the encrypted files. The adversary is assumed to have the access to all the data that is leaked to the server during operations. Before understanding the security models in detail, we need to define some terminologies that will be used in to define them. We use informal definitions to describe them for the ease of understanding. To read the formal definitions we suggest the reader to go through the original paper.[67] Note that the security model proposed by Curtmola et al. was designed for *Searchable Symmetric Encryption (SSE)*, however, it can be modified for other searchable encryption techniques.

1. *History*: It is defined as a tuple of collection of documents and set of queries that a user has made over them.
2. *Access Pattern*: For a given history access pattern is the set of document identifiers that were revealed by the queries present in the history.
3. *Search Pattern*: For a given history containing q queries, search pattern tells whether any two documents contain the same keyword or not. It is represented by using a binary matrix of size $q \times q$ where value at i^{th} row and j^{th} column is 1, if $w_i = w_j$; else, it is 0.
4. *Trace*: The length of each document in the collection along with *search pattern* and *access pattern* for the trace of a given history.
5. *Non-singular History*: A history is said to be non-singular if at least one history exists such that trace of both the histories is equal and such a history can be found in polynomial time. Trace is the information that we can allow to be leaked to

We now describe the security models presented by Curtmola et al. In both the models, history must be non-singular.

1. **Non-adaptive semantic security:** In this security model, the adversary is allowed to generate histories but only at once. Moreover, it is not allowed access to the secure index, document collection or the trapdoors of any of the queries until it has generated the histories. The scheme is considered secure if an adversary cannot distinguish between two histories that it has generated after getting access to secure index, trapdoors and encrypted documents. The authors have also provided a simulation based definition for non-adaptive security and proved it to be equivalent to the indistinguishability based definition.
2. **Adaptive semantic security:** The adaptive security model is similar to the non-adaptive one except that the adversary can choose the histories adaptively. Specifically, the challenger randomly selects a bit b and asks the adversary to provide two documents D_0 and D_1. Following which, the challenger generates secure index I_b and provides it to the adversary. Next, the challenger asks the adversary to provide two keywords w_0 and w_1 and then returns trapdoor for the word w_b to the adversary. The adversary is allowed to repeat this query process polynomial number of times. After that, the adversary is challenged to output a bit b'. The scheme is considered secure if the adversary cannot distinguish between b and b' except with some negligible probability.

It is clear from the description above that Adaptive Semantic security is harder to achieve but guarantees greater security.

7.4 | Attacks on SE schemes

The main security threats in searchable encryption scheme arise from the leakage that occurs during operations such as build and search. The first work to demonstrate a successful attack on SE schemes was presented by Islam et al.[69] in 2012. Their work exploited the leakage pattern and some prior knowledge to successfully disclose sensitive information. It demonstrated the need for a formalisation of leakages by SE schemes. Working in this direction Cash et al.[72] presented a formalisation of leakages in SE schemes They categorized SE schemes into four levels–L_1, L_2, L_3 and L_4, where L_1 denotes least amount of leakage and L_4 denotes highest.

Attacks on SE due to leakages can be divided into the following categories:[58]

1. **Keyword guessing attack (KGA):** In this attack the adversary tries to decipher the token/trapdoor in order to recover the keyword that is being searched. This is possible because in practice the words in a document are from a low entropy message space and brute force can be used to encrypt each of the words in message space and compare with the received encrypted keyword. Based on the nature of the attacker, it can further be classified into the following two categories:
   - *Inside KGA*: In inside KGA[110] the cloud server on which the encrypted data and encrypted index is stored is itself malicious. Since the attacker has access to all the search queries, it is easy for them to launch the KGA attack.
   - *Outside KGA*: In outside KGA the attacker is an external entity and has no relation with the user or cloud server. The attacker captures the data while it is being uploaded to the server and then later on tries to guess the keyword contained in a search query by comparing with word selected from the message space.
2. **Inference attacks:** In this attack the attacker tries to use the encrypted indexes to decipher keyword plaintext. Following are the commonly used methods.
   - *Frequency analysis attack*: In this attack the frequency of keywords present in the cipher-text and encrypted ranking information[111] in the outsourced data is analysed by the attacker to get the keyword plaintext.
   - *Sorting attack*: This type of attack is extremely effective for dense distribution. The attacker uses "keyword in the ciphertext to get plaintext from the encrypted indexes and tries to compute the encrypted ciphertext frequency."
   - *Counting attack*: This attack is based on the fact that a large fraction of keywords will match against a unique number of documents.[72] The adversary can then count the number of documents returned corresponding to a keyword and match it with number of documents matched by the query. The pattern of keywords across returned documents can be deducted if multiple keywords return same number of documents.
3. **Access pattern attack:** Any information that can be used by an attacker to determine the frequency at which files are accessed or their association with the query is known as access pattern.[61] In this type of attack the attacker uses access pattern to guess the keywords from the trapdoor.
4. **Search analysis attack:** Any information that can be used by the attacker to determine if random queries are related to keywords is known as search pattern.[61] In this type of attack the attacker usually tries to determine if a new trapdoor and previous trapdoor are derived from the same keyword.

7.5 | Threat model

We follow the widely-accepted Dolev-Yao (DY) threat model[112] as it was used in security protocols for other networking environments. Under the DY-model, an adversary $\mathcal{A}$ will have an opportunity to tamper with the communicating data, such as reading, modifying, deleting or even inserting fake data during the communications of the transmitted messages among the entities in the network. Additionally, as discussed in Section 7.4, $\mathcal{A}$ can launch various attacks apart from traditional attacks (replay, man-in-the-middle and impersonation attacks), like KGA, inference, access pattern and search analysis attacks. Furthermore, based on the DY-model, the end point entities are not fully trusted. We assume that once the data in put into the blockchain, $\mathcal{A}$ can not tamper with the data inside blocks, because of inherent properties of the blockchain technology (decentralization, transparency and immutability). Finally, we assume that the cloud servers are treated as semi-trusted nodes in the network.

8 | EXISTING SCHEMES FOR PHR SHARING

In this section we describe various schemes for PHR sharing that leverage searchable encryption and blockchain.

Since computers became mainstream, researchers and industrialists have been trying to digitize the entire record keeping of health records and like every other digital information, PHRs have been prone to cyber attacks of various kinds especially due to the high value of the health data that they store. However, with the advent of blockchain many

paths have opened up to tackle the shortcomings of the previous health data storage and sharing schemes. Application of searchable encryption over blockchain in healthcare is a new and active area of research. Most of the schemes using searchable encryption and blockchain, that have been proposed till date, have assumed their own model of the health care infrastructure. However, all of them follow one of the following described approaches. The first approach uses blockchain as a storage for PHRs as well as secure indexes. It is similar to the system model described in Figure 4. The second one uses blockchain only for storing secure indexes and metadata related to search and leave the storage of PHRs to a cloud server or a distributed database. It is similar to the system model described in Figure 5.

Most of the schemes have three main entities–patient, hospital and blockchain network. A patient visits a hospital and registers himself to receive his unique id and security keys depending on the type of encryption used. The hospital creates an PHR for the patient based on the diagnosis provided by the doctor. The PHR is then encrypted and uploaded to the blockchain network after the required processing for generating secure indexes is done. It is important to note that the doctor may access the medical history of the patient by asking for a token from the patient that allows him to query the blockchain network for the past medical records of the patient. This is possible because the PHRs are encrypted using searchable encryption scheme that allows authorized users to access the encrypted data. The exact details of the scheme vary depending on the assumptions that have been made about the network and storage infrastructure of the hospitals. Many of the searchable encryption scheme also use *Attribute Based Encryption (ABE)*[113] in order to have a fine-grained access control over the encrypted data. In those schemes access control structures such as AND-gate structure, tree-based structure and threshold structure are used.

8.1 | Existing schemes

We now briefly describe the existing schemes for PHR sharing using SE and blockchain.

1. **Zhang et al.**[101] **(2018):** The first specific solution to the PHR problem using blockchain and searchable encryption was proposed in 2018 by Zhang et al.[101] It is based on "*Public Key Encryption with Keyword Search (PEKS)*". Their scheme, called *Blockchain based Personal Health Information Sharing (BSPP)*, targets efficient sharing of health information of patients for improvement in diagnosis. They have used public key encryption which allows other users to access the patient's data with the help of a search token. This scheme however allows only single keyword search. Both the PHRs and the meta-data related to searching is stored on blockchain. However, they have used two blockchain networks, one of which is private and the other is consortium. Private blockchain stores the PHRs in encrypted form, while consortium blockchain stores the secure indexes and other metadata of the encrypted PHRs.

   The proposed scheme contains three phases–(1) "*System Setup*", (2) "*Data Generation and Storage*", and (3) "*Data Search and Access*". During the system setup phase, the user and doctors executes key generation algorithm to generate their public private key pairs and *GlobalSetup* algorithm is run to generate system parameter $GP = (P_1, P_2, \hat{e}, H_0, H_1, H_2, H_3, H_4, H_5, H_6)$. P_1 and P_2 are generators of cyclic groups of prime order, $\hat{e}$ is bilinear mapping and $H_i \forall i \in 0, ..., 6$ are hash functions. When a user i registers himself with the hospital k, he receives $\beta \in \{0,1\}^*$ through a secure channel. The hospital server also selects a doctor j for the user and sends $\mu = H_1(\beta)$ to the doctor. The doctor uses this value to authorize the user when he/she visits him/her. The doctor j generates a health record $m \in \{0,1\}^*$ for the patient i and also selects $w \in \{0,1\}^*$ from the standard keyword set. m and w are encrypted by doctor using public key pk_i of the user. The encryption algorithm gives $c = (c_{i_0}, c_{i_1}, c_{i_2})$ as output, where c_{i_1} is the ciphertext for searchable keyword and c_{i_2} is the evidence required for proof of conformance algorithm proposed in this paper. c_{i_0} is stored in the hospital server and the hash value of c_{i_0} is stored in blockchain. To search the user generates two trapdoors, the first one is identity searching trapdoor T_d and second one is keyword trapdoor T_w for word w. T_d is required to search for the specific user and T_w is required to search for the specific word. Both are sent to the doctor.
2. **Chen et al.**[105] **(2019):** In 2019, Chen et al.[105] proposed an Ethereum based searchable encryption scheme for PHRs. It uses *Symmetric Searchable Encryption (SSE)* and allows queries with complex expressions. This is a bit different from the traditional searchable encryption scheme as the documents that are being encrypted are of MongoDB and the queries are not keywords, but instead they are expressions used to search the MongoDB. In this scheme, all the PHRs are scanned and the records that satisfy condition expression $X = \{X_1, X_2, ..., X_m\}$ (X_i is a conditional expression that can be used to search over a Mongo DB document) are extracted and id_{ij} is computed corresponding to document D_{ij} that satisfy X_i. Based on these extracted identifiers the index I is built which is uploaded to the blockchain and he encrypted documents are outsourced to a decentralized file system such IPFS.

To search over an PHR a user authenticates himself/herself to the data owner and obtains a trapdoor/search-token. The user sends this trapdoor to the smart-contract on blockchain which returns the file identifiers on which the corresponding keyword is present. The user can then contact the storage server on which the encrypted files are stored and use the identifiers received from the smart contract to download them.

The authors describe their goals as fairness, soundness and confidentiality. Fairness ensures that user will get accurate results if he pays for it. Soundness ensures that a dishonest party will be detected and will obtain no rewards. Confidentiality ensures that the PHRs are secure.

3. **Niu et al.**[114] **(2019):** In 2019, Niu et al.[114] proposed a scheme using *Attribute Based Encryption using Keyword Search (ABKS)* and blockchain. It is based on permissioned blockchain model and uses private key encryption and supports multiple keyword search. The scheme consists of three phases–(1) "*System Setup*", (2) "*Data Generation and Storage*", and (3) "*Data Search and Access*".

   In the system setup phase, the system parameters $PP = (p, e, g_1, g_2, g_2^{\alpha}, g_2^{\beta}, g_1^{\frac{\beta}{\alpha}}, G_1, G_2, H_1, H_2, H_3, H_4)$ are generated along with master secret key $msk = (\alpha, \beta)$ Here, G_1 and G_2 are multiplicative cyclic groups with generators p, g_1 and g_2 are two generators of G_1, $e : G_1 \times G_1 \rightarrow G_2$ is an admissible bilinear map. $\alpha, \beta \in Z_p^*$ are randomly selected by the system and H_i is a hash function for $i \in 1, 2, 3, 4$.

   In data generation phase, the patient P visits hospital and is randomly assigned $h \in Z_p^*$ through a secure channel and also assigned a doctor D. Next, $\mu = H_3(h)$ is generated and reserved by the server. The patient P chooses an access control structure Γ which is in the form of a tree and runs $Share(\Gamma, h)$ for each leaf node to obtain secret value $h_v(0)$ for the leaf node v and calculates $A_v = g_2^{h_v(0)}$, $B_v = g_2^{H_1(s_v)h_v(0)}$ for each leaf node attribute s_v. These values are sent to the doctor D, which runs an encryption algorithm to compute ciphertext C and keyword index I. The encryption algorithm $Enc(PP, \Gamma, W, M) \rightarrow (I, C)$ takes public parameters PP, access control structure Γ of the patient, a keyword set $W = w_1, w_2, ..., w_m$ of the shared PHRs and PHRs M as input and returns the keyword index I and ciphertext C as output. The doctor D then uploads C and I to the hospital database server.

   In data search and access phase, data users construct the trapdoor of keywords $W' = \{w_1', w_2', ..., w_n'\}$ where $n \leq m$. The trapdoor algorithm $Trapdoor(PP, W', sk) \rightarrow T_{W'}$ takes the public parameters, search keywords set W' and secret key sk as input and outputs the trapdoor $T_{W'}$. Next, the participants in the permissoned blockchain run the search algorithm $Search(I, T_{W'})$ which takes keyword index I and trapdoor $T_{W'}$ and outputs ciphertext C. Finally the user upon receiving C decrypts the ciphertext using $Decrypt(PP, sk, C) \rightarrow M$ algorithm to obtain the message M.
4. **Wang et al.**[103] **(2019):** In 2019 Wang et al. proposed a electronic health record sharing scheme using Proxy Re-encryption with Keyword Search (PRES) and blockchain. It consists of five entities-"*Data Owner(DO)*", "*Data provider (DP)*", "*Cloud Server (CS)*", "*Blockchain (BS)*" and "*Data Requester (DR)*". The proposed scheme has three layers–(1) "*Data Generation Layer*", (2) "*Data Storage Layer*" and (3) "*Data Sharing Layer*".

   A patient requires to create an account for consortium blockchain when visiting a hospital. During registration, a patient i receives an account address A_i and a private key from the consortium blockchain. The doctor is the data provider and he/she generates PHR m for the patient and extracts a list of keywords w_i from it. He/She then encrypts m using public key pk_i of the patient, private key x_k of the doctor and keyword w_i to generate cipher text C_m. The doctor also encrypts the keywords w_i using his/her public key X_k to generate the keyword ciphertext C_w. Next, $v_1 = (C_m||C_w||A_i)$ is uploaded to the cloud server and the receives file location F_i as the response from the server, which is then send to the patient. The doctor also sends data packet $v_2 = (C_w||A_i||C_k)$ to the blockchain. Here, C_k is doctor's signature for proof of conformance for the blockchain.

   In this work, the search query is approved by the doctor instead of the patient unlike in most other works. When a doctor receives a keyword for search, he/she generates a trapdoor T_Q corresponding to it. Using the trapdoor, the data requester (DR) can search for the required PHR and account address A_i of the patient on the blockchain. Next DR sends $V_3 = (I_j||pk_j||X_k||A_j)$ to the data owner, that is, patient. Here, I_j is identity of data requester, pk_j is his/her public key, A_j is his/her account address, and X_k is doctor's public key. Upon receiving, the request from DR, patient authorizes it and sends the file location F_i and keyword set w_i to DR. Moreover, the patient also sends a re-encryption key rk to CS which carries out proxy re-encryption for the ciphertext required. Finally, the re-encrypted ciphertext is decrypted by the DR using his private key sk_j.
5. **Sun et al.**[115] **(2020):** In 2020, Sun et al.[115] proposed a solution for the problem of secure storage and sharing of Personal Health Records(PHRs) using *Attribute based Encryption with Keyword Search (ABKS)*. In this work they have used *Inter Planetary File System (IPFS)*[107] to store the records. IPFS is a distributed file system that assigns a unique identifier to each file that is stored on it. The hash of the record is stored on Ethereum based blockchain network which ensures

authenticity and integrity of the data. The encryption type used is private key, however, this scheme does not support multi-keyword search.

The scheme consists of three parts–(1) *System Establishment*, (2) *Medical Data Generation and Storage*, and (3) *Data Search and Access*. In the system establishment phase, the doctor generates public key *PK* for the system using algorithm *GlobalSetup*(λ). Here λ is the security parameter. The algorithm selects $\alpha, \beta \in Z_p^*$ and a hash function $H : \{0,1\}^* \rightarrow G_0$ and computes $A = e(g_0, g_0)^\alpha, B = g_0^\beta$ where $e : G_0 \times G_0 \rightarrow G_1$ is a symmetric bilinear map, G_0, G_1 are multiplicative cyclic groups and g_0 is generator of G_0. Further $\xi \in Z_p^*$ is selected randomly and the search private key is set as $sk_u = \{\xi\}$. Additionally, each patient $i \in U$ selects $y_i \in Z_p^*$ randomly and sets its as his/her private key sk_i and announces public key as $pk_i = g_0^{y_i}$. Similarly, each doctor $j \in D$ selects $x_j \in Z_p^*$ randomly and sets it as his/her private key and announces public key as $pk_j = g_0^{x_j}$.

During data generation and storage phase, for each patient i, a value $\varphi_i \in Z_p^*$ is randomly selected by hospital k and is then sent to patient i securely along with assigning a doctor j for the patient. The server calculates $\mu_i = H(\varphi_i)$ which is used to authorize the patient. The doctor then generates PHR m and a set of keywords W_m associated with m. He/She then encrypts m and W_m based on the access policy negotiated with the patient and generates ciphertext CT and encrypted index *index*.

When a data user wants access to the medical records of the patient, he/she sends a request to access the records along with his Ethereum public key to the doctor. The doctor assigns some attributes to the user and adds his account address to the list of authorized users after verifying his/her identity.

6. **Tang et al.[106] (2021):** Tang[106] proposed another blockchain based *Symmetric Searchable Encryption (SSE)* scheme for medical record sharing in 2021. First, the authors have presented a solution in which the entire medical record of the patient is retrieved on searching i.e without fine grained access control and then have presented a modification to the original scheme to allow fine-grained access over the medical records of the patient. Their scheme consists of four entities–(1) *hospital collection*, (2) *patient collection*, (3) *Blockchain system* and (4) *authority*. It has the following phases–(1) *Index-building*, (2) *Data retrieval*, (3) *Integrity verification*, (4) *Data addition*, and (5) *Data deletion* phases. The data addition and deletion property make this scheme dynamic in nature. It also provides forward security which ensures that previous search queries do not leak any information about the PHRs that are being added later.

   During system initialization, smart contracts are deployed and the entities involved generate their keys and security parameters. During index building phase, a patient O_i visits hospital H_j where a doctor generates the medical records denoted by $PHR_{i,j,c}$ for the patient. Here, c denotes the serial number of medical record. Hospital H_j summarizes the records for the patient O_i when they reach a certain number and generate a secret key using global identifier of the patient O_i. This secret key is used to build encrypted search index and verification index which are uploaded to the blockchain network using smart contract. During data retrieval phase, a doctor employed at another hospital H_k tries to retrieve medical history of O_i for diagnosis. O_i gives the required data for trapdoor generation to the hospital which then retrieves the PHR identifiers from the encrypted index using smart contract. The hospital then requests the corresponding hospital to send the encrypted medical records. The received records are decrypted and checked for integrity and accuracy using the verification index. After diagnosis is over new PHRs after generated by H_k and corresponding entries are added to the encrypted search index and verification index. The patient O_i also updates his locally stored information so that a new trapdoor could be generated in the future. This is important for forward security of the scheme.

7. **Wang et al.[102] (2022):** This scheme proposed by Wang et al.[102] is based on *ABKS*. They have combined consortium blockchain with a distributed file system IPFS for secure storage and sharing of PHRs. They have used attribute base encryption to allow a fine-grained access control mechanism over the PHRs and have used zero-knowledge proof for storage evidence of PHRs. The proposed scheme consists of five main entities–(1) *Data Source (DS)*, (2) *Data Owner (DO)*, (3) *Data User (DU)*, (4) *Blockchain Client (BC)* and a (5) *Blockchain Network (BN)*. Their main goal are–(1) *Data Security*, (2) *Secure Search*, (3) *Collusion-Resistant Privacy Preservation* and (4) *Personalized Access Control*.

   The scheme proposed by the authors consists of the following phases (1) *System Initialization*, (2) *Data Generation* and (3) *Data Sharing* phases. Data generation further has the following sub-phases–(1) *Cipher-text Generation*, (2) *Storage Proof Generation*, (3) *Keyword Ciphertext Generation*, (4) *Symmetric Key Cipher-text Generation* and (5) *Smart Contract Generation*. During System Initialization phase, system parameters such as master public key and master secret key are setup. In Cipher-text Generation phase, DO encrypts the PHR using symmetric key encryption. This is followed by storage proof generation where, DO uploads encrypted PHR to IPFS and receives a unique identifier

TABLE 4 Features comparison of various blockchain-based EHR sharing schemes.

Year	Scheme	Cryptographic primitive	Blockchain type	Keyword search type	Dynamic	System model
2018	Zhang et al.[101]	PEKS	Private + Consortium	Single	No	Type-1
2019	Chen et al.[105]	SSE	Public	MongoDB query expressions	No	Type-2
2019	Niu et al.[114]	ABKS	Private	Multiple	No	Type-2
2019	Wang et al.[103]	PRES	Consortium	Single	No	Type-2
2020	Sun et al.[115]	ABKS	Public	Single	No	Type-2
2021	Tang et al.[106]	SSE	Public	Single (for fine-grained search)	Yes	Type-2
2022	Wang et al.[102]	ABKS	Consortium	Single	No	Type-2

which is used as input for zero-knowledge proof to compute storage proof. In verify and consensus phase, the storage proof is sent as a transaction to the blockchain network which uses *Practical Byzantine Fault Tolerance (PBFT)* as the consensus algorithm. Next, DO chooses a set of keywords from the PHR and a set of attributes consisting of personal information of the owner and uses them to generate keyword ciphertext. This is followed by encryption of symmetric key used to generate the ciphertext of PHR. Finally, DO writes symmetric key ciphertext, identifier of PHR ciphertext and required incentive fee into a smart contract.

In the data sharing phase, DU requests for attribute key from attribute agency by providing his/her attributes. This attribute key can be used to generate trapdoor for a set of keywords that the DU wants to search for. This is done by sending the attribute key and trapdoor to the smart contract which validates both and returns PHR identifier and the symmetric key to decrypt the PHR. DU can then download the encrypted PHR from IPFS using the identifier and decrypt it using the key received from smart contract.

8.2 | Comparative study

In this section, a comparison of the schemes discussed above has been provided. Generic parameters have been chosen for comparison of features that are offered by the schemes considered above. It consists of type of cryptographic primitive used, type of blockchain model used, type of keyword search and dynamism. There are four types of cryptographic primitives as discussed in Section 7.2, namely-"*Searchable Symmetric Encryption (SSE)*, *Public key Encryption with Keyword Search (PEKS)*, *Attribute based Encryption with Keyword Search (ABKS)* and *Proxy Re-encryption with Keyword Search (PRES)*". We refer *Type-1* and *Type-2* for the system model as shown in Figures 4 and 5, respectively. The type of blockchain model can be public, private or consortium as described in Section 1.2.2. The type of keyword search refers to the type of queries allowed in the search. Single keyword search and multiple keyword search are two possible types of keyword searches. However, some schemes also use database query expressions such as that of Chen et al.[105] instead of the standard keyword search.

Table 4 shows the comparison of various features supported by the schemes under consideration as per the criteria discussed above.

9 | OPEN RESEARCH PROBLEMS

Searchable encryption with blockchain has opened up avenues for construction of health care systems that not only allow secure storage but also secure sharing of health records under the control of the data owner. The goal of research in this domain should be construct a universal healthcare infrastructure that is easily adaptable and is convenient to use for the users apart from being secure. However, there are still many open challenges that need to be tackled before we reach this goal. We discuss some of them as follows:

1. There is a need for a universal standard for the format of personal health records (PHRs). It will be easier to share PHRs between different medical institutions and other third parties involved.

2. We emphasis on the need for SE schemes that will allow more expressive queries rather than a single keyword search and still maintain the security of the PHRs.
3. There is also need for more SE schemes that should leverage the advantage of parallel computation as well.

10 | CONCLUSION

In this article, we described searchable encryption and blockchain which are currently leading technologies. We explained how search over encrypted data can be achieved and provided an understanding of the blockchain technology which has been steadily growing in use over the past decade. Additionally, we provided a description of Personal Health Records (PHRs) and the benefits of using them. Furthermore, we described how we can combine searchable encryption (SE) and blockchain together, and use them to create secure systems that are capable of secure storage and sharing of PHRs. We also discussed the latest works that have used SE and blockchain for PHRs storage and sharing. Finally, we conclude that both of these technologies show great promise for the future advancements in healthcare technologies.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable feedback.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Ashok Kumar Das https://orcid.org/0000-0002-5196-9589
Debasis Giri https://orcid.org/0000-0003-3033-3036

REFERENCES

1. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. Cryptography Mailing List at. https://metzdowd.com 2009.
2. Bera B, Mitra A, Das AK, Puthal D, Park Y. Private Blockchain-based AI-envisioned home monitoring framework in IoMT-enabled COVID-19 environment. *IEEE Consumer Electron Mag*. 2021;12(3):62-71. doi:10.1109/MCE.2021.3137104
3. Wazid M, Das AK, Park Y. Blockchain-enabled secure communication mechanism for IoT-driven personal health records. *Trans Emerg Telecommun Technol*. 2022;33(4):e4421.
4. Bera B, Das AK, Das SK. Search on encrypted COVID-19 healthcare data in blockchain-assisted distributed cloud storage. *IEEE Internet Things Mag*. 2021;4(4):127-132. doi:10.1109/IOTM.001.2100125
5. Das AK, Bera B, Giri D. AI and blockchain-based cloud-assisted secure vaccine distribution and tracking in IoMT-enabled COVID-19 environment. *IEEE Internet Things Mag*. 2021;4(2):26-32. doi:10.1109/IOTM.0001.2100016
6. Son S, Lee J, Kim M, Yu S, Das AK, Park Y. Design of secure authentication protocol for cloud-assisted telecare medical information system using blockchain. *IEEE Access*. 2020;8:192177-192191. doi:10.1109/ACCESS.2020.3032680
7. Wazid M, Bera B, Mitra A, Das AK, Ali R. Private Blockchain-Envisioned Security Framework for AI-Enabled IoT-Based Drone-Aided Healthcare Services. 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and beyond. London, United Kingdom. 2020:37-42.
8. Wazid M, Das AK, Hussain R, Kumar N, Roy S. BUAKA-CS: Blockchain-enabled user authentication and key agreement scheme for crowdsourcing system. *J Syst Arch*. 2022;123:102370.
9. Das AK, Bera B, Saha S, Kumar N, You I, Chao HC. AI-envisioned blockchain-enabled signature-based key management scheme for industrial cyber-physical systems. *IEEE Internet Things J*. 2022;9(9):6374-6388. doi:10.1109/JIOT.2021.3109314
10. Wazid M, Bera B, Das AK, Garg S, Niyato D, Hossain MS. Secure communication framework for blockchain-based internet of drones-enabled aerial computing deployment. *IEEE Internet Things Mag*. 2021;4(3):120-126. doi:10.1109/IOTM.1001.2100047
11. Bera B, Wazid M, Das AK, Rodrigues JJPC. Securing internet of drones networks using AI-envisioned smart-contract-based blockchain. *IEEE Internet Things Mag*. 2021;4(4):68-73. doi:10.1109/IOTM.001.2100044
12. Bera B, Das AK, Sutrala AK. Private blockchain-based access control mechanism for unauthorized UAV detection and mitigation in internet of drones environment. *Comput Commun*. 2021;166:91-109.
13. Bera B, Chattaraj D, Das AK. Designing secure blockchain-based access control scheme in IoT-enabled internet of drones deployment. *Comput Commun*. 2020;153:229-249.
14. Mitra A, Bera B, Das AK. Design and Testbed Experiments of Public Blockchain-Based Security Framework for IoT-Enabled Drone-Assisted Wildlife Monitoring. IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). 2021:1-6.

15. Bera B, Vangala A, Das AK, Lorenz P, Khan MK. Private blockchain-envisioned drones-assisted authentication scheme in IoT-enabled agricultural environment. *Comput Stand Interf*. 2022;80:103567.
16. Vangala A, Sutrala AK, Das AK, Jo M. Smart contract-based blockchain-envisioned authentication scheme for smart farming. *IEEE Internet Things J*. 2021;8(13):10792-10806. doi:10.1109/JIOT.2021.3050676
17. Vangala A, Das AK, Kumar N, Alazab M. Smart secure sensing for IoT-based agriculture: blockchain perspective. *IEEE Sensors J*. 2021;21(16):17591-17607. doi:10.1109/JSEN.2020.3012294
18. Chattaraj D, Bera B, Das AK, Saha S, Lorenz P, Park Y. Block-CLAP: blockchain-assisted certificateless key agreement protocol for internet of vehicles in smart transportation. *IEEE Trans Veh Technol*. 2021;70(8):8092-8107. doi:10.1109/TVT.2021.3091163
19. Vangala A, Bera B, Saha S, Das AK, Kumar N, Park Y. Blockchain-enabled certificate-based authentication for vehicle accident detection and notification in intelligent transportation systems. *IEEE Sensors J*. 2021;21(14):15824-15838. doi:10.1109/JSEN.2020.3009382
20. Bagga P, Sutrala AK, Das AK, Vijayakumar P. Blockchain-based batch authentication protocol for internet of vehicles. *J Syst Arch*. 2021;113:101877.
21. Bera B, Saha S, Das AK, Kumar N, Lorenz P, Alazab M. Blockchain-envisioned secure data delivery and collection scheme for 5G-based IoT-enabled internet of drones environment. *IEEE Trans Veh Technol*. 2020;69(8):9097-9111. doi:10.1109/TVT.2020.3000576
22. Chattaraj D, Bera B, Das AK, Rodrigues JJPC, Park Y. Designing fine-grained access control for software-defined networks using private blockchain. *IEEE Internet Things J*. 2022;9(2):1542-1559. doi:10.1109/JIOT.2021.3088115
23. Shashidhara R, Ahuja N, Lajuvanthi M, Akhila S, Das AK, Rodrigues JJPC. SDN-chain: privacy-preserving protocol for software defined networks using blockchain. *Secur Privacy*. 2021;4(6):e178.
24. Banerjee S, Bera B, Das AK, Chattopadhyay S, Khan MK, Rodrigues JJ. Private blockchain-envisioned multi-authority CP-ABE-based user access control scheme in IIoT. *Comput Commun*. 2021;169:99-113.
25. Saha S, Chattaraj D, Bera B, Kumar DA. Consortium blockchain-enabled access control mechanism in edge computing based generic internet of things environment. *Trans Emerg Telecommun Technol*. 2021;32(6):e3995.
26. Bera B, Das AK, Obaidat MS, Vijayakumar P, Hsiao KF, Park Y. AI-enabled blockchain-based access control for malicious attacks detection and mitigation in IoE. *IEEE Consum Electron Mag*. 2021;10(5):82-92. doi:10.1109/MCE.2020.3040541
27. Bera B, Saha S, Das AK, Vasilakos AV. Designing blockchain-based access control protocol in IoT-enabled smart-grid system. *IEEE Internet Things J*. 2021;8(7):5744-5761. doi:10.1109/JIOT.2020.3030308
28. Jangirala S, Das AK, Vasilakos AV. Designing secure lightweight blockchain-enabled RFID-based authentication protocol for supply chains in 5G Mobile edge computing environment. *IEEE Trans Industr Inform*. 2020;16(11):7081-7093. doi:10.1109/TII.2019.2942389
29. Qian Y, Jiang Y, Hu L, Hossain MS, Alrashoud M, Al-Hammadi M. Blockchain-based privacy-aware content caching in cognitive internet of vehicles. *IEEE Netw*. 2020;34(2):46-51.
30. Saghiri AM, Vahdati M, Gholizadeh K, Meybodi MR, Dehghan M, Rashidi H. A Framework for Cognitive Internet of Things Based on Blockchain. 4th International Conference on Web Research (ICWR'18). Tehran, Iran. 2018:138-143.
31. Vahdati M, Gholizadeh HamlAbadi K, Saghiri AM, Rashidi H. A self-organized framework for insurance based on internet of things and Blockchain. IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud'18), Barcelona, Spain. 2018:169-175.
32. Jakobsson M, Juels A. Proofs of Work and Bread Pudding Protocols: 258–272; Secure Information Networks: Communications and Multimedia Security IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS'99) September 20–21, 1999. Leuven, Belgium. 1999.
33. Castro M, Liskov B. Practical byzantine fault tolerance and proactive recovery. *ACM Trans Comput Syst*. 2002;20(4):398-461.
34. Ongaro D, Ousterhout J. In Search of an Understandable Consensus Algorithm. USENIX Annual Technical Conference (Usenix ATC 14), Philadelphia, PA, USA. 2014:305-319.
35. Pease M, Shostak R, Lamport L. Reaching agreement in the presence of faults. *J ACM*. 1980;27(2):228-234.
36. Schwartz D, Youngs N, Britto A. The ripple protocol consensus algorithm. *Ripple Labs Inc White Paper*. 2014;5(8):151.
37. Saleh F. Blockchain without waste: proof-of-stake. *Rev Financ Stud*. 2021;34(3):1156-1190.
38. Li K, Li H, Hou H, Li K, Chen Y. Proof of Vote: A High-Performance Consensus Protocol Based on Vote Mechanism & Consortium Blockchain. IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Bangkok, Thailand. 2017:466-473.
39. Wood G. Ethereum: a secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*. 2014;2014(151):1-32.
40. Bentov I, Pass R, Shi E. Snow white: provably secure proofs of stake. *IACR Cryptol ePrint Arch*. 2016;2016(919):1-65.
41. King S, Nadal S. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. 2012.
42. Kiayias A, Russell A, David B, Oliynykov R. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. Advances in Cryptology (CRYPTO'17), Santa Barbara, CA, USA. 2017:357-388.
43. Kwon J. Tendermint: consensus without mining. *Self-Published Paper (Draft v06)*. 2014;1(11):1-11. https://tendermint.com/static/docs/tendermint.pdf
44. The XRP. Ledger. 2014. https://xrpl.org/consensus-principles-and-rules.html
45. Hyperledger Sawtooth Architecture Guide. Intel Corporation. 2020. https://sawtooth.hyperledger.org/docs/1.2/
46. Buterin V. A next-generation smart contract and decentralized application platform. *White Paper*. 2014;3(37):1-2.
47. Golosova J, Romanovs A. The advantages and disadvantages of the Blockchain technology. 6th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE). Vilnius, Lithuania. 2018:1-6.

48. Song DX, Wagner D, Perrig A. Practical Techniques for Searches on Encrypted Data. IEEE Symposium on Security and Privacy (S&P'00). Berkeley, California, USA. 2000:44-55.
49. Katz J, Lindell Y. *Introduction to Modern Cryptography*. 2nd ed. Chapman & Hall/CRC; 2014.
50. Boneh D. Pairing-based cryptography: past, present, and future. International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'12), Beijing, China. 2012:1.
51. Menezes A. An Introduction to Pairing-Based Cryptography. 2013. https://www.math.uwaterloo.ca/~ajmeneze/publications.
52. Stallings W. *Cryptography and Network Security: Principles and Practice*. 5th ed. Prentice Hall; 2010.
53. Sarkar P. A simple and generic construction of authenticated encryption with associated data. *ACM Trans Inf Syst Secur*. 2010;13(4).
54. May WE. Secure Hash Standard. 2015 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
55. Bloom BH. Space/time trade-offs in hash coding with allowable errors. *Commun ACM*. 1970;13(7):422-426.
56. Kosba A, Miller A, Shi E, Wen Z, Papamanthou C. Hawk: the Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. 2016 IEEE Symposium on Security and Privacy (SP). San Jose, CA, USA: IEEE. 2016:839-858.
57. Bösch C, Hartel P, Jonker W, Peter A. A survey of provably secure searchable encryption. *ACM Comput Surv*. 2014;47(2):1-51. doi:10.1145/2636328
58. Wang Y, Wang J, Chen X. Secure searchable encryption: a survey. *J Commun Inf Netw*. 2016;1:52-65.
59. Poh GS, Chin JJ, Yau WC, Choo KKR, Mohamad MS. Searchable symmetric encryption: designs and challenges. *ACM Comput. Surv.* 2017;50(3):1-37.
60. Zhang R, Xue R, Liu L. Searchable encryption for healthcare clouds: a survey. *IEEE Trans Serv Comput*. 2018;11(6):978-996. doi:10.1109/TSC.2017.2762296
61. Pham H, Woodworth J, Amini SM. Survey on secure search over encrypted data on the cloud. *Concurrency Comput: Pract Exper*. 2019;31(17):e5284.
62. Andola N, Gahlot R, Yadav VK, Venkatesan S, Verma S. Searchable encryption on the cloud: a survey. *J Supercomput*. 2022;78(7):9952-9984.
63. Goh E. Secure indexes. *IACR Cryptol ePrint Arch*. 2003;216:1-18. http://eprint.iacr.org/2003/216
64. Boneh D, Di Crescenzo G, Ostrovsky R, Persiano G. Public key encryption with keyword search. International Conference on the Theory and Applications of Cryptographic Techniques – Advances in Cryptology (EUROCRYPT'04), Interlaken, Switzerland. 2004:506-522.
65. Boneh D, Franklin M. Identity-Based Encryption from the Weil Pairing. 21st Annual International Cryptology Conference – Advances in Cryptology (CRYPTO'01). Santa Barbara, California, USA. 2001:213-229.
66. Chang YC, Mitzenmacher M. Privacy Preserving Keyword Searches on Remote Encrypted Data. International Conference on Applied Cryptography and Network Security (ACNS'05). New York, NY, USA. 2005:442-455.
67. Curtmola R, Garay J, Kamara S, Ostrovsky R. Searchable symmetric encryption: improved definitions and efficient constructions. *J Comput Secur*. 2011;19(5):895-934.
68. Kamara S, Papamanthou C, Roeder T. *Dynamic Searchable Symmetric Encryption. ACM Conference on Computer and Communications Security (CCS'12)*. Raleigh, North Carolina, USA. 2012:965-976.
69. Islam MS, Kuzu M, Kantarcioglu M. Access pattern disclosure on searchable encryption: ramification, attack and mitigation. Network and distributed system security symposium. Citeseer. 2012:1-12.
70. Stefanov E, Papamanthou C, Shi E. Practical dynamic searchable encryption with small leakage. *Cryptol ePrint Arch, Paper 2013/832*. 2013.
71. Kamara S, Papamanthou C. Parallel and Dynamic Searchable Symmetric Encryption. International Conference on Financial Cryptography and Data Security. Okinawa, Japan: Springer. 2013:258-274.
72. Cash D, Grubbs P, Perry J, Ristenpart T. Leakage-Abuse Attacks against Searchable Encryption. 22nd ACM Conference on Computer and Communications Security (CCS). Denver, Colorado, USA. 2015:668-679.
73. Bost R, Fouque PA, Pointcheval D. Verifiable dynamic symmetric searchable encryption: optimality and forward security. *Cryptol ePrint Arch*. Paper 2016/062, 2016; https://eprint.iacr.org/2016/062
74. Bost R, Minaud B, Ohrimenko O. Forward and Backward Private Searchable Encryption from Constrained Cryptographic Primitives. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas, Texas, USA. 2017:1465-1482.
75. Kim KS, Kim M, Lee D, Park JH, Kim WH. Forward Secure Dynamic Searchable Symmetric Encryption with Efficient Updates. ACM Conference on Computer and Communications Security (CCS). Dallas, TX, USA. 2017:1449-1463.
76. Etemad M, Küpçü A, Papamanthou C, Evans D. Efficient dynamic searchable encryption with forward privacy. *Proc Privacy Enhanc Technol*. 2017;2018:20-25.
77. Watanabe Y, Ohara K, Iwamoto M, Ohta K. Efficient Dynamic Searchable Encryption with Forward Privacy under the Decent Leakage. Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy. Baltimore, MD, USA. 2022:312-323.
78. Asharov G, Segev G, Shahaf I. Tight tradeoffs in searchable symmetric encryption. *J Cryptol*. 2021;34(2):1-37.
79. Zuo C, Sun SF, Liu JK, Shao J, Pieprzyk J. Dynamic searchable symmetric encryption schemes supporting range queries with forward (and backward) Security. European Symposium on Research in Computer Security (ESORICS'18). Springer, Barcelona, Spain. 2018:228-246.
80. Song X, Dong C, Yuan D, Xu Q, Zhao M. Forward private searchable symmetric encryption with optimized I/O efficiency. *IEEE Trans Depend Secure Comput*. 2018;17(5):912-927.
81. Ghareh Chamani J, Papadopoulos D, Papamanthou C, Jalili R. New Constructions for Forward and Backward Private Symmetric Searchable Encryption. ACM Conference on Computer and Communications Security (CCS). Toronto, Canada. 2018:1038-1055.

82. Li J, Huang Y, Wei Y, et al. Searchable symmetric encryption with forward search privacy. *IEEE Trans Depend Secure Comput*. 2019;18(1):460-474.
83. Zuo C, Sun SF, Liu JK, Shao J, Pieprzyk J. Dynamic Searchable Symmetric Encryption with Forward and Stronger Backward Privacy. European Symposium on Research in Computer Security (ESORICS'19). Luxembourg: Springer. 2019:283-303.
84. Chen L, Li J, Li J. Toward forward and backward private dynamic searchable symmetric encryption supporting data deduplication and conjunctive queries. *IEEE Internet Things J*. 2023;10(19):17408-17423. doi:10.1109/JIOT.2023.3274390
85. Lu Y, Li J, Zhang Y. Secure channel free certificate-based searchable encryption withstanding outside and inside keyword guessing attacks. *IEEE Trans Serv Comput*. 2021;14(6):2041-2054. doi:10.1109/TSC.2019.2910113
86. Lu Y, Li J. Lightweight public key authenticated encryption with keyword search against adaptively-chosen-targets adversaries for mobile devices. *IEEE Trans Mob Comput*. 2022;21(12):4397-4409. doi:10.1109/TMC.2021.3077508
87. Lu Y, Li J, Zhang Y. Privacy-preserving and pairing-free multirecipient certificateless encryption with keyword search for cloud-assisted IIoT. *IEEE Internet Things J*. 2020;7(4):2553-2562. doi:10.1109/JIOT.2019.2943379
88. Lu Y, Li J, Wang F. Pairing-free certificate-based searchable encryption supporting privacy-preserving keyword search function for IIoTs. *IEEE Trans Industr Inform*. 2021;17(4):2696-2706. doi:10.1109/TII.2020.3006474
89. Qian H, Li J, Zhang Y, Han J. Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation. *Int J Inf Secur*. 2015;14(6):487-497. doi:10.1007/s10207-014-0270-9
90. Li J, Lin X, Zhang Y, Han J. KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage. *IEEE Trans Serv Comput*. 2017;10(5):715-725. doi:10.1109/TSC.2016.2542813
91. Li J, Zhang Y, Ning J, Huang X, Poh GS, Wang D. Attribute based encryption with privacy protection and accountability for CloudIoT. *IEEE Trans Cloud Comput*. 2022;10(2):762-773. doi:10.1109/TCC.2020.2975184
92. Hu S, Cai C, Wang Q, Wang C, Luo X, Ren K. Searching an Encrypted Cloud Meets Blockchain: A Decentralized, Reliable and Fair Realization. IEEE INFOCOM 2018-IEEE Conference on Computer Communications. Honolulu, HI, USA. 2018:792-800.
93. Li H, Tian H, Zhang F, He J. Blockchain-based searchable symmetric encryption scheme. *Comput Electr Eng*. 2019;73:32-45.
94. Guo Y, Zhang C, Jia X. Verifiable and Forward-Secure Encrypted Search Using Blockchain Techniques. IEEE International Conference on Communications (ICC). IEEE. 2020:1-7.
95. Du R, Wang Y. Verifiable Blockchain-Based Searchable Encryption with Forward and Backward Privacy. 16th International Conference on Mobility, Sensing and Networking (MSN). Tokyo, Japan: IEEE. 2020:630-635.
96. Guo Y, Lu Z, Ge H, Li J. Revocable blockchain-aided attribute-based encryption with escrow-free in cloud storage. *IEEE Trans Comput*. 2023;72(7):1901-1912. doi:10.1109/TC.2023.3234210
97. Azaria A, Ekblaw A, Vieira T, Lippman A. Medrec: Using Blockchain for Medical Data Access and Permission Management. 2016 2nd International Conference on Open and Big Data (OBD). Vienna, Austria: IEEE. 2016:25-30.
98. Yue X, Wang H, Jin D, Li M, Jiang W. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *J Med Syst*. 2016;40(10):1-8.
99. Xia Q, Sifah EB, Smahi A, Amofa S, Zhang X. BBDS: blockchain-based data sharing for electronic medical records in cloud environments. *Information*. 2017;8(2):44.
100. Fan K, Wang S, Ren Y, Li H, Yang Y. Medblock: efficient and secure medical data sharing via blockchain. *J Med Syst*. 2018;42(8):1-11.
101. Zhang A, Lin X. Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain. *J Med Syst*. 2018;42(8):1-18.
102. Wang Y, Zhang A, Zhang P, Qu Y, Yu S. Security-aware and privacy-preserving personal health record sharing using consortium blockchain. *IEEE Internet Things J*. 2022;9(14):12014-12028. doi:10.1109/JIOT.2021.3132780
103. Wang Y, Zhang A, Zhang P, Wang H. Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain. *Ieee Access*. 2019;7:136704-136719.
104. Alsayegh M, Moulahi T, Alabdulatif A, Lorenz P. Towards secure searchable electronic health records using consortium blockchain. *Network*. 2022;2(2):239-256.
105. Chen L, Lee WK, Chang CC, Choo KKR, Zhang N. Blockchain based searchable encryption for electronic health record sharing. *Future Gener Comput Syst*. 2019;95:420-429.
106. Tang X, Guo C, Choo KKR, Liu Y, Li L. A secure and trustworthy medical record sharing scheme based on searchable encryption and blockchain. *Comput Netw*. 2021;200:108540. doi:10.1016/j.comnet.2021.108540
107. Benet J. Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561; 2014.
108. Zhao F, Nishide T, Sakurai K. Multi-User Keyword Search Scheme for Secure Data Sharing with Fine-Grained Access Control. Information Security and Cryptology-ICISC 2011: 14th International Conference, Seoul, Korea, November 30-December 2, 2011. Revised Selected Papers 14. Springer. 2012:406-418.
109. Shao J, Cao Z, Liang X, Lin H. Proxy re-encryption with keyword search. *Inform Sci*. 2010;180(13):2576-2587.
110. Huang Q, Li H. An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks. *Inform Sci*. 2017;403:1-14.
111. Wang C, Cao N, Li J, Ren K, Lou W. Secure ranked keyword search over encrypted cloud data. IEEE 30th International Conference on Distributed Computing Systems (ICDCS'10), Genoa, Italy. 2010:253-262.
112. Dolev D, Yao A. On the security of public key protocols. *IEEE Trans Inf Theory*. 1983;29(2):198-208.
113. Sahai A, Waters B. Fuzzy identity-based encryption. Annual International Conference on the Theory and Applications of Cryptographic Techniques – Advances in Cryptology (EUROCRYPT'05). Springer, Aarhus, Denmark. 2005:457-473.

114. Niu S, Chen L, Wang J, Yu F. Electronic health record sharing scheme with searchable attribute-based encryption on blockchain. *IEEE Access*. 2019;8:7195-7204.
115. Sun J, Ren L, Wang S, Yao X. A blockchain-based framework for electronic medical records sharing with fine-grained access control. *PloS One*. 2020;15(10):e0239946.

How to cite this article: Bisht A, Das AK, Giri D. Personal health record storage and sharing using searchable encryption and blockchain: A comprehensive survey. *Security and Privacy*. 2024;7(2):e351. doi: 10.1002/spy2.351