SeCoSe: Toward Searchable and Communicable Healthcare Service Seeking in Flexible and Secure EHR Sharing

Zhihuang Liu, Ling Hu, Zhiping Cai, *Member, IEEE*, Ximeng Liu, *Senior Member, IEEE*, and Yanhua Liu

***Abstract*— Cloud-assisted electronic health record (EHR) sharing plays an important role in modern healthcare systems but faces threats of distrust and non-traceability. The advent of blockchain offers an attractive solution to overcome this issue. Many efforts are devoted to promoting secure, flexible, and multi-featured blockchain-based EHR sharing. Yet, the problem of seeking out suitable healthcare providers and communicating information beyond the EHR has unfortunately been ignored. In this paper, we propose SeCoSe, a novel EHR sharing scheme to address these concerns. SeCoSe enables patients and their general practitioners to autonomously seek out and stay in touch with their preferred healthcare professionals. Specifically, a searchable and repeatable transformation identity-based encryption (SRTIBE) is proposed to achieve dynamic and flexible authorization updates. Moreover, we design attribute-identity mapping contracts and evidence-based contracts on the blockchain to enable on-demand retrieval of anonymous identities and ensure tamper resistance and traceability of system transactions. Furthermore, we employ the advanced messages on-chain protocol (AMOP) to facilitate the online communication of off-chain messages. Detailed security analysis and extensive evaluations demonstrate that SeCoSe is privacy-secure, traceable, and attack-resistant. SeCoSe has lower overhead for repeated authorization and transformation, on-chain transactions can be responded to within seconds, and online communication can handle the transmission of 49,000 messages in about 6 seconds.**

***Index Terms*— Electronic health records, healthcare service seeking, blockchain, smart contract, identity-based encryption.**

I. Introduction

ELECTRONIC health records (EHRs) are an essential component of modern healthcare systems [1]. EHR sharing further enhances collaboration and information exchange among healthcare practitioners, patients, and other stakeholders. This is crucial for improving medical treatment decisions and enhancing patient experiences [2], [3], [4]. Also, EHR sharing plays an important role in combating pandemics such as COVID-19 and facilitating remote healthcare services [5], [6].

Outsourcing EHRs to cloud servers is a common practice to facilitate storage, access, and management [7]. However, cloud-based data sharing technologies raise concerns regarding security and privacy, posing risks of compromising the integrity and confidentiality of EHRs containing patient-sensitive information [8]. To address this issue, encryption-based solutions serve a vital role in safeguarding the privacy of EHRs [9], [10]. Many encryption schemes with fine-grained access control have been proposed to enhance the security and flexibility of cloud-based EHR sharing systems [11], [12], [13]. Nevertheless, cloud-based data sharing systems still suffer from the difficulty of tracing and auditing when subjected to tampering attacks or data breaches, while additional auditing schemes incur significant computational and communication overhead [2], [14].

Recently, blockchain technology with features such as tamper-proofing, anonymity, and traceability has emerged as an attractive solution for ensuring the security of systems like healthcare services. Blockchain-based EHR sharing systems have the capability to anonymize user identities, resist data tampering attacks, and provide trustworthy auditing [1], [15], [16]. Furthermore, by combining blockchain with off-chain encryption techniques, EHR sharing can not only enhance security but also achieve additional features such as search flexibility and public verifiability [3], [17].

A. Related Work and Motivation

Most of the existing work related to blockchain-based EHR sharing can be classified into four application scenarios. TABLE I presents a comparative summary of some schemes within each scenario.

Type I focuses on the transfer of patients' EHRs between different hospitals, discussing collaborative diagnosis or scientific research involving multiple hospitals or medical institutions [4], [7], [18]. Zaghloul et al. [4] utilize Attribute-Based Encryption (ABE) and smart contracts to allow patients to share different parts of their medical records with m different data users and describe a method for access permission

Manuscript received 2 November 2023; revised 10 March 2024; accepted 16 April 2024. Date of publication 22 April 2024; date of current version 7 May 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFF1203001; in part by the National Natural Science Foundation of China under Grant 62172155, Grant 62072465, Grant 62102425, and Grant U22B2005; and in part by the Science and Technology Innovation Program of Hunan Province under Grant 2022RC3061. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Qinghua Li. *(Corresponding author: Zhiping Cai.)*

Zhihuang Liu, Ling Hu, and Zhiping Cai are with the College of Computer, National University of Defense Technology, Changsha 410082, China (e-mail: lzhliu@nudt.edu.cn; linghu50@nudt.edu.cn; zpcai@nudt.edu.cn).

Ximeng Liu and Yanhua Liu are with the College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China (e-mail: snbnix@gmail.com; lyhwa@fzu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIFS.2024.3391914, provided by the authors.

Digital Object Identifier 10.1109/TIFS.2024.3391914

TABLE I
COMPARISON WITH RELATED WORK ABOUT BLOCKCHAIN-BASED EHR SHARING

Scheme	Scenario	Off-chain privacy-preserving	Fine-grained access control	Authorization update	Changes in allowed access	Finding the suitable providers	Online communication
[4]	Type I	ABE	✓	Revocation	$0 \rightarrow m$	×	×
[7]	Type I	ABE	✓	×	$0 \rightarrow m$	×	×
[18]	Type I	PRE	✓	Re-encryption	$1 \rightarrow 1$	×	×
[3]	Type II	ABE	✓	Revocation	$0 \rightarrow m$	×	×
[2]	Type II	ABE	✓	Revocation	$0 \rightarrow m \rightarrow m'$	×	×
[17]	Type II	SE	✓	×	$0 \rightarrow 1$	×	×
[15]	Type III	ZKP	×	N/A	N/A	×	×
[19]	Type III	FL	×	N/A	N/A	×	×
[20]	Type IV	N/A	N/A	N/A	N/A	×	✓
[21]	Type IV	Diagnosis key	N/A	N/A	$0 \rightarrow 1$	×	×
[14]	Type IV	SEnc	✓	×	$0 \rightarrow 1$	×	×
Ours	Type IV	IBE	✓	Dynamic update	$1 \rightarrow m \rightarrow m'$	✓	✓

"N/A" denotes not comparable. m: Number of authorized users for the initial authorization; m': Number of authorized users for repeated authorization.

revocation. Wang et al. [7] also design an ABE scheme to achieve fine-grained access control and embed access policies into the blockchain. However, [7] does not consider how to support authorization updates. Lin et al. [18] employ Proxy Re-Encryption (PRE) to achieve selective sharing of patient records between hospital A and hospital B, granting access to new users through the generation of re-encryption keys.

Type II is initiated by data users such as hospitals or research institutes, requesting access to patients' EHR data [2], [3], [17]. Liu et al. [3] propose a hybrid blockchain-backed searchable proxy signcryption scheme, utilizing search contracts to determine whether data users have access permissions, and also supporting tracing and user revocation. Xu et al. [2] also uses ABE to support flexible EHR fine-grained access control, including dynamic revocation of ciphertexts. Chen et al. [17] allow users to perform efficient ciphertext search and public verification through Searchable Encryption (SE) and blockchain, although they do not involve dynamic updates of access permissions.

Type III involves the remote monitoring of patients' conditions by medical institutions and making treatment decisions or emergency rescues [15], [19]. Using the Zero-Knowledge Proof (ZKP) protocol for authentication, Aujia and Jindal [15] propose a blockchain model for medical monitoring that can securely store EHR data in the cloud, including emergency situations. Singh et al. [19] propose a blockchain-based healthcare privacy protection architecture that analyzes health data such as EHR through Federated Learning (FL) and sends alerts to healthcare providers. However, access control for EHRs has received less attention in Type III scenarios.

Type IV is a proactive patient-initiated request for medical treatments to healthcare providers [14], [20], [21]. Kordestani et al. [20] proposes a blockchain-based remote consultation framework called Hapicare, which includes an interface for communication between patients and doctors. However, [20] assumes that patients have appropriate doctors, such as their general practitioners [22], and it is only at the framework design stage without a concrete implementation. Similarly, [21] assumes that patients already have suitable doctors when requesting diagnoses. Li et al. [14] introduces a secure EHR traceability mechanism based on blockchain and Symmetric Encryption (SEnc), where the data flow starts with the patient initiating an appointment and being assigned a doctor by the hospital. However, [14] only discusses the single-doctor case, omitting the problem of doctor allocation.

Unfortunately, we observe that the aforementioned works have overlooked the following two issues:

*(1) How to proactively seek suitable healthcare providers (**searchable**):* Even in Type IV, where patients proactively initiate appointments, the default assumption is that doctors or medical institutions are assigned to request the patients' EHRs. In fact, patients autonomously seeking preferred healthcare providers is a crucial aspect of patient-centered care and plays a vital role in enhancing the patient experience [23], [24], [25]. However, the challenge lies in *how patients can find healthcare providers with anonymous identities within the blockchain system that align with their specific needs.* It is worth noting that the *searchability* problem addressed is not the retrievability of EHR ciphertexts as referred to in searchable encryption schemes [3], [17], [26], [27], [28], [29].

*(2) How to maintain communication with suitable healthcare providers (**communicable**):* Existing work assumes that patients and healthcare providers have a secure communication channel outside the system, such as face-to-face or telephone communication. However, in remote healthcare services or anonymous service systems, the challenge is *how to deliver messages to healthcare providers who are unwilling to disclose their real contact information and how to receive private responses from them.* Achieving *communicability* additionally involves considering the efficient transmission of on-chain messages to overcome the effects of the blockchain's consensus mechanism [30], [31].

In fact, addressing these two challenges is crucial for the widespread adoption of blockchain-based EHR sharing systems, especially in future applications of remote healthcare services. The comprehensive motivating scenario that helps to understand is shown in Fig. 1, which depicts an application scenario of this paper. Due to space limitations, a detailed description of Fig. 1 can be found in **Supplementary Material A**. Our work belongs to Type IV, and it is similar to [3] in that general practitioners of resource-limited patients are empowered to authorize other healthcare

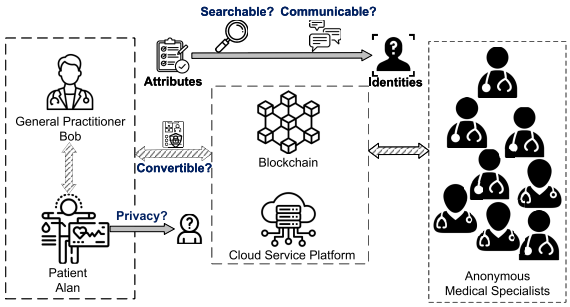


Fig. 1. Application scenario and challenges.

providers. However, we also find that existing access control techniques in Type IV are relatively rough, with insufficient flexibility in granting authorization to data users. Nevertheless, from Type I to Type III, it can be observed that fine-grained access control and flexible authorization updates are important in the EHR sharing process.

B. Our Contributions

To address the above challenging problems, in this paper, we propose a searchable and communicable proactive healthcare service seeking scheme (SeCoSe) in a blockchain-based EHR sharing system.

The main contributions of this paper are summarized below:

1) The proposed SeCoSe is designed to refine the details of EHR sharing systems for practical applications. SeCoSe is a novel scheme within blockchain-based EHR sharing systems that considers patient-centric retrieval of available healthcare providers (**searchable**) and facilitates anonymous communication with them (**communicable**). SeCoSe can support anonymous identity mapping and online communication in blockchain-based data sharing systems to bridge the gap between off-chain and on-chain.
2) A searchable and repeatable transformation identity-based encryption (SRTIBE) method is proposed to achieve dynamic multiple updates of access permissions for ciphertext data. By integrating with on-chain evidence-based contracts, SRTIBE enables SeCoSe searchable, verifiable, and fine-grained access control for encrypted EHR sharing, with data tamper resistance and transaction traceability.
3) Smart contracts are designed on the blockchain to implement attribute-identity mapping, enabling the discovery of suitable healthcare providers and facilitating the off-chain SRTIBE encryption for designated identities. By leveraging the AMOP technique to forward off-chain user messages to anonymous on-chain nodes, healthcare providers can be contacted while maintaining anonymity.
4) We provide a formal security analysis and detailed theoretical analysis to demonstrate the security and efficiency of our scheme. Experimental simulations and implementations show that SeCoSe saves the cost of repeated updates of ciphertext authorization. Lightweight on-chain transactions can be controlled within seconds, and message communication can be completed in milliseconds. Extensive evaluations show that SeCoSe effectively realizes a secure and flexible EHR sharing system with service searchable and communicable capabilities.

II. Preliminary

Our construction is based on Identity-Based Encryption (IBE) [32], [33], smart contracts, and AMOP [34] technology. This section primarily introduces the bilinear map, the foundational scheme IBET [35] of SRTIBE, and AMOP technology. The adoption of IBE for EHR sharing aims to better emphasize patient-centeredness and enable data owners to have clearer information about authorized identities (anonymously). Another popular encryption scheme, ABE, often suffers from the issue of increased overhead as the number of attributes grows [10], [36].

A. Bilinear Map

Let $\mathcal{G}_p$ be a bilinear group generating algorithm that takes a security parameter λ as input and outputs $(p, g, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative cyclic groups of prime order p, g is a generator of $\mathbb{G}$. The bilinear map e is defined as $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.

B. IBET

The identity-based encryption transformation (IBET) [35] scheme consists of the following algorithms: Setup, Register, Encrypt, Authorize, Transform, and Decrypt.

- $\mathsf{Setup}(1^\lambda, m) \rightarrow (PP, MSK)$. The authority entity as a registry runs the Setup algorithm, taking as input the security parameter λ and the maximum number of data users m authorized to the same data, and outputs the system's primary public parameters PP and the registry's own primary key MSK.
- $\mathsf{Register}(PP, MSK, ID) \rightarrow SK_{ID}$. The registry takes the PP, MSK, and the user's identity ID as input and outputs a private key SK_{ID} for the user.
- $\mathsf{Encrypt}(PP, M, ID) \rightarrow CT_{ID}$. The data owner runs the Encrypt algorithm to encrypt the message M using PP and the authorized identity ID as input, and outputs an IBE-formatted ciphertext CT_{ID}.
- $\mathsf{Authorize}(PP, SK_{ID}, S) \rightarrow TK_{ID \rightarrow S}$. This algorithm is run by the entity with identity ID. It takes SK_{ID}, PP, and a set of identities S for authorized visitors as inputs, and outputs an authorization token $TK_{ID \rightarrow S}$.
- $\mathsf{Transform}(PP, TK_{ID \rightarrow S}, CT_{ID}) \rightarrow CT_S$. The cloud service provider responsible for storing ciphertexts runs the Transform algorithm using PP, $TK_{ID \rightarrow S}$, and IBE-formatted ciphertext CT_{ID} as input. It outputs a transformed IBBE-formatted ciphertext CT_S.
- $\mathsf{Decrypt}(PP, CT_{ID}/CT_S, SK_{ID'}) \rightarrow M/\perp$. This algorithm is run by a data user with identity ID'. It takes PP, $SK_{ID'}$, and an IBE ciphertext CT_{ID} or IBBE ciphertext CT_S as input. For CT_{ID}, if $ID' = ID$ then it outputs the message M; otherwise it outputs $\perp$. For CT_S, if $ID' \in S$ then it outputs the message M; otherwise it outputs $\perp$.

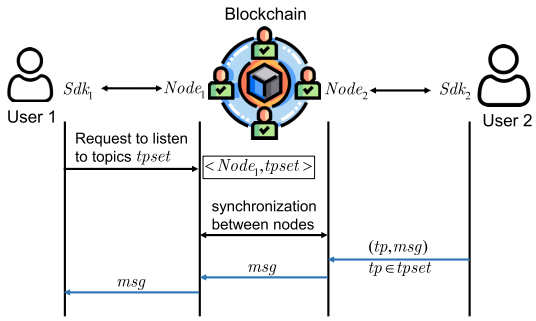


Fig. 2. An example of using AMOP to deliver messages.

We note that when an IBE ciphertext is transformed into an IBBE ciphertext, Authorize uses the $SK_{ID^\star}$ of $ID^\star$ to authorize new users, which is consistent with the $ID^\star$ in Encrypt. Otherwise, the correct transformation cannot be completed in Transform. Therefore, in an EHR share using IBET, Alan completes Encrypt, where $ID^\star$ belongs to Bob, the first authorized user to access the ciphertext. Then Bob extends the permission for this ciphertext through Authorize and hands it over to the cloud platform for Transform. This makes IBET a kind of Identity-Based Proxy Re-Encryption (IB-PRE) [37]. Therefore, it is necessary to clarify the relationship between data owner Alan and IBE ciphertext visitor Bob, that is, Bob is trusted by Alan. In addition, IBET does not pay attention to details such as ciphertext retrieval matching and multiple updates of authorization. Therefore, based on the aforementioned observations, we propose a searchable and repeatable transformation identity-based encryption (SRTIBE).

C. AMOP

The Advanced Messages On-chain Protocol (AMOP) [34] aims to provide a secure and efficient communication channel for consortium blockchains. AMOP messages are transmitted in real-time between nodes with millisecond-level latency, independent of blockchain transactions and consensus. Additionally, all communication links in AMOP are encrypted using SSL and support identity authentication mechanisms. AMOP has two topic modes: public topic and private topic. The private topic provides an identity authentication mechanism to prevent irrelevant recipients from listening to the topic. However, due to the signature and verification processes involved, the cost of using a private topic is higher. Therefore, in SeCoSe, only the public topic mode of AMOP is utilized. Moreover, by utilizing the secret value in the ciphertext as the topic name, it can also achieve the effect of private message transmission.

Fig. 2 shows an example of AMOP usage. User 1 connects to the blockchain node $Node_1$ through the Software Development Kit (SDK) interface sdk_1, while user 2 connects to blockchain node $Node_2$ through sdk_2. User 1 sends a collection of topics $tpset$ to $Node_1$ through sdk_1, and $Node_1$ synchronizes the mapping relationship $<Node_1, tpset>$ to the blockchain nodes. When user 2 sends a message msg to the topic tp through sdk_2, where $tp \in tpset$, sdk_2 forwards the message to $Node_2$, which in turn sends msg to $Node_1$, as $Node_1$ has subscribed to the tp topic. $Node_1$ then forwards the message to user 1's sdk_1 to notify user 1.

III. Problem Formulation

In this section, we present the system model, threat model, SRTIBE definition proposed, SeCoSe definition proposed, and the design goals of the scheme. In this paper, two scheme names, SeCoSe and SRTIBE, are proposed. SeCoSe is a more comprehensive overall solution, while SRTIBE, included within SeCoSe, is an improvement on IBET [35] designed to implement SeCoSe. Moreover, SeCoSe and SRTIBE involve two different concepts of 'searchable'. In SeCoSe, the main problem that SeCoSe addresses in terms of searchability refers to the patient-centric retrieval of available healthcare providers, rather than the retrievability of EHR ciphertexts as indicated in most related works. In SRTIBE, 'searchable' indeed refers to the retrievability of ciphertexts. SRTIBE uses hash values as indices for ciphertexts to efficiently find matching ciphertexts. Given that there is already mature research on finding corresponding ciphertexts based on hash values [4], [11], [17], [38], the search details are simplified and included in the specific implementation of SRTIBE's algorithms.

A. System Model

The system model of SeCoSe shown in Fig. 3 consists of six entities, namely Registration Authority (RA), Patients (Data Owners, DOs), General Practitioners (Delegators, Dgs), Medical Specialists (Data Users, DUs), Cloud Service Platform (CSP), and Consortium Blockchain (CB). There are multiple DOs, Dgs and DUs in the system. The details of each entity are given below in combination with the steps in Fig. 3, aiming to describe the workflow of the system model.

- The RA is responsible for initializing the system, deploying smart contracts, registering users, and generating private keys in the system.
- DOs are patients with limited resources. They suffer from some rare diseases that cannot be cured by their Dgs (Step ①). To relieve DOs' computational and storage pressure, they do not directly participate in the healthcare blockchain. Instead, they entrust their corresponding Dgs to share EHR and seek medical advice, thus also protecting the privacy of DOs' personal identity.
- Dgs understand their DOs' medical history and assist them in completing preliminary examinations and treatments for their rare diseases (Step ①). In Step ②, Dgs are responsible for retrieving suitable DUs for DOs and granting corresponding DUs access to the latest EHR ciphertext, then publishing digest information on CB. Meanwhile, Dgs notify the corresponding DUs about the request for diagnosis through the messaging channel. In Step ⑤, Dgs receive notifications from DUs regarding the completion of the diagnosis, and through Step ⑥, they access and provide feedback on the diagnosis results. Dgs can update the encrypted EHR and access permissions based on the consultation and treatment progress, repeating the diagnosis and treatment process multiple times (Step ⑦)

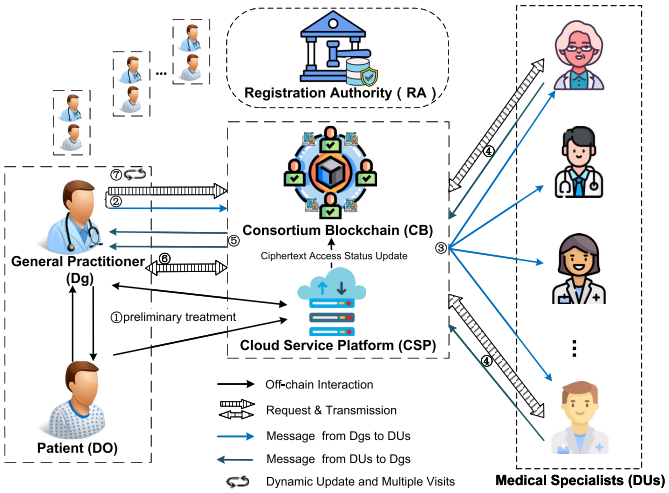


Fig. 3. System model.

- DUs receive diagnosis requests from DOs in Step ③, and some DUs respond to these requests. In Step ④, these DUs retrieve the EHR ciphertext for diagnosis and upload encrypted treatment plans to the CSP while also uploading digest information to CB. At the same time, the DUs also issue notifications upon completing the diagnosis and treatment.
- The CSP is a data storage platform capable of high capacity and high performance in encrypted form. The CSP provides retrieval services and sends the ciphertext to entities with permissions. In addition, the CSP is responsible for uploading the access status of encrypted data to CB.
- CB is a permissioned blockchain with an admission mechanism that enables secure data sharing within a relatively controlled scale. CB only stores some lightweight data such as hash values of encrypted data, which is tamper-proof and traceable. CB is also tasked with realizing online message transmission between entities.

In the system, DOs, Dgs, and DUs are collectively referred to as users. When registering users, the RA transforms their real identities into anonymous identities. Within the system, an incentive mechanism based on interests and reputation is employed to attract specialists proficient in various domains of rare diseases.

B. Threat Model

In our system, the RA and CB are fully trusted entities. Each Dg associated with a DO is trustworthy and acts as an agent for his/her DO to complete the EHR sharing process and obtain benefits. Different Dgs are semi-trusted, and DUs and the CSP are also semi-trusted. They follow predefined system rules and scheme rules, but curiosity about transmitted information such as EHR and diagnostic information in the system may cause sensitive information leakage. Moreover, the CSP may return incorrect data to other entities in order to save computing resources.

C. SRTIBE Definition

A searchable and repeatable transformation identity-based encryption scheme can be denoted as $\mathbf{\Pi} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}_{\mathsf{DO}}, \mathsf{Authorize}_{\mathsf{Dg}}, \mathsf{Transform}, \mathsf{ReAuth}_{\mathsf{Dg}}, \mathsf{ReTran}_{\mathsf{CSP}}, \mathsf{Request}_{\mathsf{DU}}, \mathsf{DecVer}_{\mathsf{DU}})$. Compared with IBET, SRTIBE adapts to the proposed system model and specifies the usage permissions of the algorithms. SRTIBE can support ciphertext retrieval and correctness verification in the decryption phase. Moreover, SRTIBE considers algorithms for multiple updates of authorization and ciphertext transformation. In this paper, 'repeated authorization and transformation' is used to fully describe the two processes of $\mathsf{ReAuth}_{\mathsf{Dg}}$ and $\mathsf{ReTran}_{\mathsf{CSP}}$. The input of Setup is the same as that of $\mathsf{IBET.Setup}$, but the output PP is different. KeyGen, $\mathsf{Authorize}_{\mathsf{Dg}}$, and $\mathsf{Transform}$ align with $\mathsf{IBET.Register}$, $\mathsf{IBET.Authorize}$, and $\mathsf{IBET.Transform}$, respectively. The remaining algorithms are described below.

- $\mathsf{Enc}_{\mathsf{DO}}(PP, ID_{Dg}, EHR) \rightarrow (CT_l, H_l)$. The DO takes the identity ID_{Dg} of his/her Dg and the EHR as inputs, and outputs an EHR ciphertext CT_l and a hash value H_l used as a retrieval and verification token.
- $\mathsf{ReAuth}_{\mathsf{Dg}}(PP, S') \rightarrow c_{new}$. The Dg runs $\mathsf{ReAuth}_{\mathsf{Dg}}$ to perform repeated authorization. The input is PP, a new set of authorized users S'. The output is a partial ciphertext c_{new} that will be uploaded to the CSP for replacement.
- $\mathsf{ReTran}_{\mathsf{CSP}}(CT_l, c_{new}) \rightarrow CT_S$. The CSP replaces a portion of the stored ciphertexts CT_l with c_{new} and outputs the updated ciphertext CT_S.
- $\mathsf{Request}_{\mathsf{DU}}(H_l, ID_i, S) \rightarrow CT_S/\perp$. The algorithm inputs the retrieval token H_l, the requester's identity ID_i, and a set of authorized users S. The CSP retrieves data based on H_l and outputs ciphertext CT_S if ID_i is in the permission set S; otherwise, it outputs $\perp$.
- $\mathsf{DecVer}_{\mathsf{DU}}(PP, CT_S, H_l, S, SK_{DU_i}) \rightarrow (M, EHR)/\perp$. The DU runs $\mathsf{DecVer}_{\mathsf{DU}}$ algorithm with input PP, CT_S, H_l, and his/her private key SK_{DU_i}. If M recovered from decryption matches H_l after an operation with the partial ciphertext, the algorithm successfully decrypts the correct EHR; otherwise, it outputs $\perp$.

D. SeCoSe Definition

A SeCoSe scheme can be denoted as $\mathbf{\Lambda} = (\mathbf{\Pi}, \mathbf{\Gamma}, \overleftarrow{\mathbf{\Pi}}, \overleftarrow{\mathbf{\Gamma}})$. Thereinto, $\mathbf{\Pi}$ refers to the proposed SRTIBE, $\mathbf{\Gamma} = (\mathtt{Deploy}, \mathtt{Reg}, \mathtt{Search}, \mathtt{Issue}_{\mathtt{Dg}}, \mathtt{Amop}_{\mathtt{Dg}}, \mathtt{Sta}_{\mathtt{CT_S}}, \mathtt{UpAuth}_{\mathtt{CT_S}}, \mathtt{Close}_{\mathtt{CT_S}})$ represents the main smart contract algorithms deployed on CB, $\overleftarrow{\mathbf{\Pi}} = (\mathsf{Enc}_{\mathsf{DU}}, \mathsf{Request}_{\mathsf{Dg}}, \mathsf{DecVer}_{\mathsf{Dg}})$ comprises algorithms derived from $\mathbf{\Pi}$ that enables the DU-to-Dg direction, and $\overleftarrow{\mathbf{\Gamma}} = (\mathtt{Issue}_{\mathtt{DU}}, \mathtt{Amop}_{\mathtt{DU}}, \mathtt{Sta}_{\mathtt{CT_r}}, \mathtt{FeBack}_{\mathtt{Dg}})$ comprises smart contract algorithms in the DU-to-Dg direction that are derived from $\mathbf{\Gamma}$. The algorithms in $\mathbf{\Gamma}$ and $\overleftarrow{\mathbf{\Gamma}}.\mathtt{FeBack}_{\mathtt{Dg}}$ are described below. In particular, most of the return values of the algorithms have been omitted to save on algorithm call costs, as the call details and results of the algorithms in $\mathbf{\Gamma}$ and $\overleftarrow{\mathbf{\Gamma}}$ can be known by nodes on the blockchain.

- $\mathtt{Deploy}(PK_{RA}, SK_{RA}) \rightarrow \mathtt{Sc}$. This algorithm is run by the RA and takes the RA's public key PK_{RA} and private key SK_{RA} as input, and outputs the deployed smart contract address $\mathtt{Sc}$.

- $\texttt{Reg}(\texttt{Sc}, ID, AttS)$. The algorithm consists of $\texttt{Reg}_{\texttt{Dg}}$, $\texttt{Reg}_{\texttt{DU}}$ and $\texttt{Reg}_{\texttt{CSP}}$. The RA registers the on-chain address ID of each Dg, DU, and CSP via $\texttt{Sc}$, respectively, each with attribute $AttS$ and a certain amount of tokens (called coins in this paper).
- $\texttt{Search}(\texttt{Sc}, dis, cds) \rightarrow S$. The Dg inputs the DO's potential disease type dis and the required attribute constraints cds via $\texttt{Sc}$, and the algorithm outputs the set of IDs S of DUs that satisfy the specified constraints.
- $\texttt{Issue}_{\texttt{Dg}}(\texttt{Sc}, H_l, S, Tm)$. A Dg runs this algorithm via $\texttt{Sc}$ to publish the digest information of the encrypted EHR. It includes the retrieval and verification token H_l, the set of authorized DUs S, and the allowance for the first Tm DUs to access.
- $\texttt{Amop}_{\texttt{Dg}}(Tops, Cont)$. This algorithm is run by a Dg to broadcast a message with topic name $Tops$ and content $Cont$.
- $\texttt{Sta}_{\texttt{CT}_\texttt{S}}(\texttt{Sc}, H_l, IDs) \rightarrow Tm/false$. The CSP updates the access status of the ciphertext CT_S to the CB via $\texttt{Sc}$, taking H_l and the set of accessed IDs as inputs. This algorithm verifies the input's validity, returning $false$ if it is not valid; otherwise, it outputs the remaining number Tm of DUs allowed to access after updating.
- $\texttt{UpAuth}_{\texttt{CT}_\texttt{S}}(\texttt{Sc}, H_l, S')$. This algorithm is run by the Dg with the token H_l via $\texttt{Sc}$ to store the updated authorized user set S' of the ciphertext CT_S.
- $\texttt{Close}_{\texttt{CT}_\texttt{S}}(\texttt{Sc}, H_l)$. This algorithm is run by the Dg with the token H_l via $\texttt{Sc}$ to set the ciphertext CT_S as no longer accepting DUs' access.
- $\texttt{FeBack}_{\texttt{Dg}}(\texttt{Sc}, H_r)$. This algorithm is run by the Dg via $\texttt{Sc}$ to pay for consultation fee corresponding to H_r.

E. Design Goals

1) *Privacy Preservation.* This includes three aspects: the integrity and validity of shared data in the system are guaranteed and verifiable; shared EHR and diagnostic information should not be accessed by unauthorized users, and private notification messages sent from DUs to specified Dg should not be disclosed to other users; users' anonymous identities in the system should be difficult to associate with their real identities, especially for DOs.
2) *Fine-grained Access Control.* Dgs can find a specific set of identities based on attributes for repeatable access authorizations, and Dgs can also terminate the open access to EHR ciphertext. Only authorized DUs or Dgs that meet the requirements can retrieve and access the corresponding files.
3) *Traceability.* Data owners should be able to trace the access history of their data; any transactions performed on CB should be traceable; malicious operations and errors within the system should be auditable and traceable.
4) *Attack Resistance.* Misuse of smart contracts and frequent resource consumption behaviors should be prevented, as well as block tampering attacks, collision attacks, 51% attacks, etc.

TABLE II
SUMMARY OF NOTATIONS

Notation	Description
PP, MSK	public parameter, master secret key
H_0, H_1, H_2, H_3	four strong collision-resistant hash functions
PK, SK	a public key, secret key pair of a user
ID_{off}/ID	off-chain identity / on-chain address
$\texttt{Sc}$	smart contract address on CB
$AttS, S$	user attributes, DUs set matching the requirements
S'	Updated set of authorized DUs
$\mathbb{R}$	constraint attributes for searching
$CT_l/CT_S/CT_r$	ciphertext from DO to Dg / Dg to DUs / DUs to Dg
C_R/C_G	ciphertext of EHR/DTG after symmetric encryption
H_l/H_r	retrieval and verification token of EHR/DTG ciphertext
Tm	number of DUs allowed to access CT_S
Cls	whether CT_S access is closed in the system
$d_{Dg \rightarrow S}$	authorization token for DUs set S
$Tops, Cont$	topic names of AMOP, message content of AMOP

IV. The Proposed SeCoSe

In this section, we give the overview and concrete construction of SeCoSe. Important notations are summarized in Table II.

A. Overview of SeCoSe

A highlight of SeCoSe is the design of a holistic off-chain and on-chain integration solution, enabling close collaboration between off-chain encryption algorithms and blockchain functionality. Blockchain enables flexible EHR sharing with additional features, including user anonymity, searchability and communicability of healthcare providers, and traceability and tamper resistance of system transactions.

The high-level description of SeCoSe is provided below.

1) The RA performs the initialization phase to obtain system public parameters and deploy smart contracts. The RA also registers users, specifying their attributes and values, assigning anonymous addresses, and issuing private keys. DUs also complete the initialization of AMOP topic subscriptions.
2) The DO and the Dg maintain the transmission of EHRs and information through IBE encryption, with the Dg completing preliminary diagnosis and treatment.
3) Based on the negotiated requirements, the Dg finds the set of IDs of eligible DUs through smart contracts and converts the original IBE ciphertext into an IBBE ciphertext that supports access by multiple DUs. Then, the Dg stores the digest information of the EHR ciphertext on the chain and sends a message seeking diagnosis to DUs proficient in the corresponding diseases through AMOP. Upon receiving the message, DUs obtain the specified EHR ciphertext through the CSP and decrypt it for verification. The overview of this part is shown in Fig. 4.
4) DUs provide diagnostic and treatment guidance based on the EHR content, then encrypt the guidance content using IBE and store the digest information on the chain. DUs also specify an AMOP topic to send a notification of completed diagnosis to a specific Dg. The Dg obtains the ciphertext of the guidance content, decrypts it for

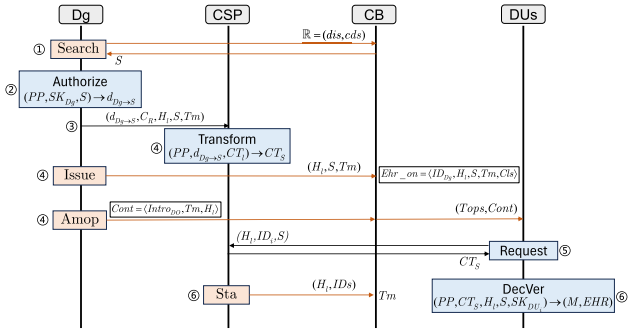


Fig. 4. The workflow of searchable and communicable sharing.

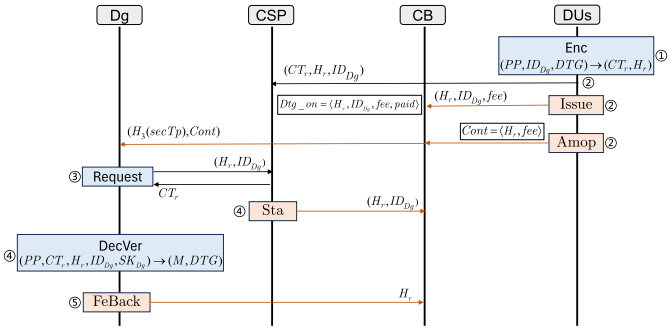


Fig. 5. The workflow of responsive designated sharing.

verification, and then treats his/her DO based on the guidance of DUs. The overview of this part is shown in Fig. 5.

5) Based on the latest demand, Dgs can repeatedly update the access permissions of the EHR ciphertext or terminate the open state of the EHR ciphertext.

B. System Initialization

This phase consists of the RA initializing the system and registering the users.

- $\mathsf{Setup}(1^\lambda, m) \rightarrow (PP, MSK)$. Input the security parameter λ to obtain the bilinear group $(p, g, \mathbb{G}, \mathbb{G}_T, e)$, where g is a generator of group $\mathbb{G}$. The RA randomly selects $\alpha \in \mathbb{Z}_p^*$ and $h, u \in \mathbb{G}$, and then computes $Y = e(g, h)$ and $g^\alpha, u^\alpha, h^\alpha, h^{\alpha^2}, \ldots, h^{\alpha^m} \in \mathbb{G}$, where m is the maximum number of users that can access the same ciphertext. The RA also defines four strong collision-resistant hash functions: $H_0 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $H_1 : \mathbb{G}_T \rightarrow \mathbb{G}$, $H_2 : C_M \times \mathbb{G}_T \rightarrow \{0,1\}^*$, and $H_3 : \{0,1\}^* \rightarrow \{0,1\}^\varsigma$. Here, $M \in \mathbb{G}_T$, C_M represents the ciphertext of symmetric encryption with symmetric key M, and ς denotes the length of the H_3 values. The primary public parameters is $PP = (g^\alpha, u, u^\alpha, h, h^\alpha, h^{\alpha^2}, \ldots, h^{\alpha^m}, Y, H_0, H_1, H_2, H_3)$, and the primary secret key is $MSK = (g, \alpha)$. The PP in the system is embedded in the first block of the CB, i.e., the genesis block, as public information.
- $\mathtt{Deploy}(PK_{RA}, SK_{RA}) \rightarrow \mathtt{Sc}$. The RA creates a public-private key pair PK_{RA}, SK_{RA} that is known only to the RA, which is converted by algorithms such as hash to obtain a unique address string ID_{RA} that represents the RA's on-chain account. The RA uses this ID_{RA} to deploy a designed smart contract on CB, which becomes the first transaction on the blockchain. The deployed smart contract gets a unique address on CB called contract account $\mathtt{Sc}$. Subsequent transactions on CB will be performed by loading $\mathtt{Sc}$.
- $Register(PP, MSK, ID_{off}, AttS, \mathtt{Sc}) \rightarrow (SK, ID)$. The $Register$ consists of $\mathtt{Reg}_{\mathtt{Dg}}$, $\mathtt{Reg}_{\mathtt{DU}}$, $\mathtt{Reg}_{\mathtt{CSP}}$, and $\mathsf{KeyGen}(PP, MSK, ID) \rightarrow SK$. When a user requests to join the system from the RA, the legitimacy of the user's identity ID_{off} and attribute $AttS$ are checked first. If the check passes, the CSP and users other than DOs will be assigned an on-chain address ID with attributes $AttS$ and a certain number of coins, where ID serves as a unique identifier. In particular, the RA also assigns an anonymous address to a DO but will not put the ID on the chain. At the same time, the user's private key is obtained by calculating $SK = g^{\frac{1}{\alpha + H_0(ID)}}$ through KeyGen. The RA then sends SK, ID, and $\mathtt{Sc}$ (DOs do not have $\mathtt{Sc}$, and the CSP does not have SK) to the requester over a secure channel.

It is worth noting that user identities are sensitive information that includes names, contact information, and even assets and addresses. These pieces of information are reviewed by RA and kept confidential by RA to ensure privacy, and are not input into the smart contract algorithm.

As for attribute $AttS$, different types of users have different uniform standards. For Dgs, attributes include professional qualifications, years of practice, etc. As constraints for subsequent retrieval, the attributes of DUs are more abundant and standardized to facilitate Dgs finding corresponding DUs based on on the requested constraint attributes. Specifically, each DU has $\vartheta + 1$ attributes as $types$ and $\{att_1, att_2, \ldots, att_i, \ldots, att_\vartheta\}$. Thereinto, $types$ is an array that stores the mapping of specialist proficiency in disease categories such as *1: cardiovascular diseases* and *2: respiratory diseases*. These mapping relationships are common knowledge in the system. Moreover, DUs will subscribe to topics corresponding to their proficient fields based on AMOP. Any attribute att_i has a definite meaning and range. For example, att_1 denotes the age of practicing in the range of [3,60], att_2 takes values in [1,1000], representing the number of difficult cases cured, and att_3 represents the reputation value of the specialist with a range of [60,100]. The explicit DU attribute definition in the system helps Dgs initiate searches based on attributes to find specialists that match their needs more easily.

In addition, we introduce the concept of coins during registration. Subsequent publishing or responding transactions through smart contracts require a certain amount of coin consumption. Further design regarding reputation, coins, and incentive mechanisms can be referred to [16], [39], [40], and [41].

C. Off-Chain Interaction Between DOs and Dgs

In this phase, we describe a universal IBE and the interaction between DOs and Dgs.

1) Universal IBE.

- $\mathsf{Enc}_{\mathsf{IBE}}(PP, ID, msg) \rightarrow CT_{IBE}$. Randomly select $s \in \mathbb{Z}_p^*$ and $M \in \mathbb{G}_T$ to generate the IBE ciphertext of

message msg for ID.

$$C_1 = M \cdot Y^s, C_2 = h^{s(\alpha + H_0(ID))},$$
$$C_{msg} = \mathsf{SEnc_M}(msg \| 0^\xi).$$

Thereinto, $\mathsf{SEnc_M}$ is a symmetric encryption algorithm like AES, and 0^ξ indicates that an ξ-bit string of 0 is concatenated. The algorithm returns $CT_{IBE} = (C_1, C_2, C_{msg})$.

- $\mathsf{Dec_{IBE}}(PP, SK_{ID}, CT_{IBE}) \rightarrow (M, msg)$. After receiving the IBE ciphertext CT_{IBE}, use the private key SK_{ID} corresponding to ID to decrypt it. The recipient first computes M and then recovers msg using the symmetric encryption algorithm SDec.

$$M = C_1 / e(SK_{ID}, C_2), msg' = \mathsf{SDec_M}(C_{msg}).$$

One can get the msg by checking whether there are redundant zeros after the plaintext, and if $msg' = msg \| 0^\xi$, the msg can be obtained by removing ξ-bit 0.

2) The DO and the Dg maintain short message transmission through bidirectional $\mathsf{Enc_{IBE}}$ and $\mathsf{Dec_{IBE}}$ in a direct connected channel $\mathfrak{C}$, while files such as EHR are stored and transmitted through the CSP after encryption. When a DO is unwell and needs to seek a diagnosis from his/her Dg, the encryption algorithm $\mathsf{Enc_{DO}}$ is run.

- $\mathsf{Enc_{DO}}(PP, ID_{Dg}, EHR) \rightarrow (CT_l, H_l)$. It uses the EHR as the msg for $\mathsf{Enc_{IBE}}$ to get $C_1, C_2, C_R = \mathsf{SEnc_M}(EHR \| 0^\xi)$, it also computes

$$C_3 = u^{s(\alpha + H_0(ID_{Dg}))}, H_l = H_2(C_R, M).$$

Then, $CT_l = (C_1, C_2, C_3, C_R)$, H_l and the authorized access ID_{Dg} are outsourced to the CSP. The CSP stores CT_l using H_l as the index for CT_l, enabling subsequent search and identification of stored EHR ciphertexts based on H_l. The DO also encrypts H_l and passes it to his/her Dg via $\mathfrak{C}$.

3) The Dg requests access to CT_l from the CSP as ID_{Dg} with retrieval condition H_l, then decrypts CT_l using $\mathsf{Dec_{IBE}}$. By comparing the two H_l from the CSP and $\mathfrak{C}$, the Dg can determine whether the CSP has returned an incorrect ciphertext. Finally, the Dg diagnoses and treats the DO.

D. Searchable and Communicable Sharing

The workflow for this phase, as depicted in Fig. 4, is specifically described as follows.

1) The DO's condition has not improved after a period of treatment and has therefore entrusted the Dg to seek diagnosis from specialists in the relevant fields.

- $\mathtt{Search}(\mathtt{Sc}, dis, cds) \rightarrow S$. The DO and his/her Dg agree that the specialist they are looking for should satisfy the constraint attributes $\mathbb{R} = (dis, cds)$, containing the attributes and attribute values they wanted to search for. Thereinto, dis is a one-dimensional array representing the possible diseases that the DO's current condition may belong to, and cds is an array consisting of v two-element arrays $[[s_{11}, s_{12}], \ldots, [s_{i1}, s_{i2}], \ldots, [s_{v1}, s_{v2}]]$, where $[s_{i1}, s_{i2}]$ denotes the lower and upper bounds of the constraints on the i-th attribute of the specialists. Then, the Dg runs the algorithm `Search` on `Sc` to return a set of IDs $S = \{ID_i\}_{i=1}^{v \le m}$ that match the attribute requirements, where v is the number of eligible specialists. The description of `Search` is shown in Algorithm 1. Due to space limitations, the detailed descriptions of the `Contains`, `CheckConditions`, and `Slice` functions in the smart contract, along with $\mathtt{Reg}_{\mathtt{DU}}$, are provided in **Supplementary Material B**.

Algorithm 1 Searching From Constraint Attributes ($\mathbb{R}$) to Matched IDs (S) by Using the Smart Contract

Input: $\mathbb{R} = (dis, cds)$.
Output: $S = \{ID_i\}_{i=1}^{v \le m}$.

```
Initialize matIDs to an address array with length the
  number of DUs (DUsNum), initialize count to 0;
for i = 0 to DUsNum do
    bool isMatch = true;
    if Contains(DUs[mapadd[i]].types, dis) then
        isMatch = true;
    else isMatch = false; continue;
    for j = 0 to cds.length do
        if CheckCds(i, j, cds[j]) = false then
            isMatch = false; break;
    end
    if isMatch = true then
        matIDs[count] = mapadd[i];
        count++;
end
return S = Slice(matIDs, count).
```

2) The Dg authorizes access to the ciphertext CT_l for $S = \{ID_i\}_{i=1}^{v \le m}$.

- $\mathsf{Authorize_{Dg}}(PP, SK_{Dg}, S) \rightarrow d_{Dg \rightarrow S}$. The Dg randomly chooses $t, r \in \mathbb{Z}_p^*$, and generates authorization token $d_{Dg \rightarrow S} = (d_1, d_2, d_3, d_4)$ for $S = \{ID_i\}_{i=1}^{v \le m}$ with the Dg's SK_{Dg}, where

$$d_1 = (g^\alpha)^{-t}, d_2 = h^{t \prod_{i=1}^{v} (\alpha + H_0(ID_i))},$$
$$d_3 = H_1(Y^t) \cdot h^r, d_4 = SK_{Dg} \cdot u^{-r}.$$

The Dg then sends the latest C_R, H_l, S, Tm and $d_{Dg \rightarrow S}$ to the CSP, where $Tm \le v$ indicates the number of DUs allowed access. In particular, the latest EHR plaintext contains an $H_3(secTp)$, which is subsequently used as a specific topic name for the DUs passing back on-chain messages to privately notify only the Dg.

3) After receiving the authorization token from the authorized Dg, the CSP retrieves the corresponding ciphertext CT_l based on H_l, updates the corresponding C_R, S, and Tm, and transforms CT_l into the ciphertext CT_S to support multiple DUs access.

- $\mathsf{Transform}(PP, d_{Dg \rightarrow S}, CT_l) \rightarrow CT_S$. This algorithm outputs $CT_S = (c_1, c_2, c_3, c_4, c_5, C_R)$, where

$$c_1 = d_1, c_2 = d_2, c_3 = d_3, c_4 = C_3,$$
$$c_5 = C_1 / e(C_2, d_4) = M \cdot e(h^{s(\alpha + H_0(ID_{Dg}))}, u^r).$$

4) While sending the authorization token to the CSP, the Dg also deposits the digest information of the data stored in the CSP on CB and sends a message to notify the designated DUs through AMOP.

- $\texttt{Issue}_{\texttt{Dg}}(\texttt{Sc}, H_l, S, Tm)$. The Dg publishes H_l, S, and Tm to CB through Sc. $\texttt{Issue}_{\texttt{Dg}}$ is only allowed to be called by Dgs, and if H_l has already been uploaded, it will be interrupted to avoid abusing on-chain transactions. Otherwise, the algorithm first collects a predefined number of coins, then creates a digest structure $Ehr\_on = \langle ID_{Dg}, H_l, S, Tm, Cls\rangle$ of the ciphertext CT_S stored on CB, where H_l serves as the unique identifier and Cls indicates whether CT_S access is closed in the system.
- $\texttt{Amop}_{\texttt{Dg}}(Tops, Cont)$. The Dg sends messages to DUs nodes that have subscribed to a specific topic, $Tops$, in a broadcast manner. Here, $Tops$ is consistent with the array dis, and $Cont$ is formatted content $\langle Intro_{DO}, Tm, H_l\rangle$ that represents 'The basic introduction of patients and diseases is $Intro_{DO}$, requiring responses from the first Tm DUs and verifying that the token is H_l'.

5) Some DUs that receive AMOP notifications from the Dg will request the ciphertext data from the CSP, and then decrypt and verify it locally.

- $\mathsf{Request}_{\mathsf{DU}}(H_l, ID_i, S) \rightarrow CT_S/\perp$. The CB nodes that subscribe to any topic in $Tops$ will receive broadcast notifications and push messages to the SDKs of these DUs. This will promptly notify the wiring departments of the DUs through bulletin boards or website pop-ups. Subsequently, a DU with identity ID_i requests encrypted data with specified H_l from the CSP. After retrieving the corresponding data, the CSP verifies whether Tm of the data is greater than 0 and whether ID_i is in the S list of the data. If the conditions are not satisfied, $\perp$ is returned; otherwise, the CSP returns CT_S to the DU. In addition, the CSP updates the access status of CT_S within a certain period of time by invoking the $\texttt{Sta}_{\texttt{CT}_\texttt{S}}$ algorithm.
- $\texttt{Sta}_{\texttt{CT}_\texttt{S}}(\texttt{Sc}, H_l, IDs) \rightarrow Tm/false$. The algorithm first checks whether H_l has been uploaded and whether each ID_i in $IDs = \{ID_i\}_{i=1}^{sn\leq Tm}$ has been registered. If not, it outputs false. Otherwise, the algorithm reduces Tm by sn and outputs Tm, which corresponds to the ciphertext of H_l. DUs who wish to request the ciphertext of H_l from CSP can check the remaining number of allowed accesses to DUs in advance by calling the view type function CheckTm.
- $\mathsf{DecVer}_{\mathsf{DU}}(PP, CT_S, H_l, S, SK_{DU_i}) \rightarrow (M, EHR)/\perp$. For the ciphertext $CT = (c_1, c_2, c_3, c_4, c_5, C_R)$ and S with length v, a DU with identity $ID_i \in S$ computes

$$B = \left(e(c_1, h^{\rho_{i,S}(\alpha)}) \cdot e(SK_{DU_i}, c_2)\right)^{\frac{1}{\prod_{j=1, j\neq i}^{v} H_0(ID_j)}},$$

where

$$\rho_{i,S}(\alpha) = \frac{1}{\alpha} \cdot \Big(\prod_{j=1, j\neq i}^{v} (\alpha + H_0(ID_j)) - \prod_{j=1, j\neq i}^{v} H_0(ID_j) \Big).$$

Then obtain $h^r = c_3/H_1(B)$ and $M = c_5/e(h^r, c_4)$. The DU checks whether $H_l = (C_R, M)$ holds. If it is, the DU can recover plaintext $EHR' = \mathsf{SDec}_{\mathsf{M}}(C_R)$ using symmetric key M. The DU also checks whether there are redundant zeros at the end of the plaintext. If $EHR' = EHR\|0^{\xi}$, EHR can be obtained by removing the ξ-bit zeros. Otherwise, the DU can claim that CSP has returned an incorrect CT_S and output $\perp$.

E. Responsive Designated Sharing

The workflow for this phase, as depicted in Fig. 5, is specifically described as follows.

1) A DU provides diagnostic and treatment guidance (DTG) for the DO, which can include multimedia files in audio and video formats. The DU then runs the following algorithm:

- $\mathsf{Enc}_{\mathsf{DU}}(PP, ID_{Dg}, DTG) \rightarrow (CT_r, H_r)$. DTG is encrypted using $\mathsf{Enc}_{\mathsf{IBE}}$ to obtain $CT_r = (C_1, C_2, C_G)$, and it also computes $H_r = H_2(C_G, M)$. CT_r, H_r, and authorized ID_{Dg} are then uploaded to the CSP. The CSP stores CT_r using H_r as the index for CT_r, enabling subsequent search and identification of stored DTG ciphertexts based on H_r.
- $\texttt{Issue}_{\texttt{DU}}(\texttt{Sc}, H_r, ID_{Dg}, fee)$. Similar to the $\texttt{Issue}_{\texttt{Dg}}$ process, the DU publishes H_r and ID_{Dg} on the chain to generate a digest structure $Dtg\_on = \langle H_r, ID_{Dg}, fee, paid\rangle$ of the CT_r, where $paid$ indicates whether the fee has been paid.
- $\texttt{Amop}_{\texttt{DU}}(H_3(secTp), Cont)$. The DU sends a message containing $Cont$ to the topic name $H_3(secTp)$ previously specified by the Dg, ensuring that only a specific Dg can receive this message. $Cont$ contains 'Diagnostics completed, token requested and verified is H_r, fee to be paid is fee'.

2) Upon receiving an AMOP notification, the Dg requests the ciphertext data corresponding to a specific H_r from the CSP and then decrypts and verifies it locally. The Dg also responds and pays the consultation fee through Sc.

- $\mathsf{Request}_{\mathsf{Dg}}(H_r, ID_{Dg}) \rightarrow CT_r/\perp$. The Dg requests CT_r data from the CSP based on H_r. The CSP retrieves and verifies the corresponding data and calls $\texttt{Sta}_{\texttt{CT}_\texttt{r}}$.
- $\texttt{Sta}_{\texttt{CT}_\texttt{r}}(\texttt{Sc}, H_r, ID_{Dg})$. This algorithm verifies whether H_r has been uploaded and whether ID_{Dg} is valid. If it passes the verification, H_r is marked as accessed, indicating that the corresponding ciphertext CT_r has been requested by the Dg.
- $\mathsf{DecVer}_{\mathsf{Dg}}(PP, CT_r, H_r, ID_{Dg}, SK_{Dg}) \rightarrow (M, DTG)/\perp$. The Dg runs algorithm $\mathsf{Dec}_{\mathsf{IBE}}$ to recover symmetric key M and verifies its validity by $H_r \stackrel{?}{=} H_2(C_G, M)$. If it is valid, DTG plaintext can be obtained; otherwise, $\perp$ is output.
- $\texttt{FeBack}_{\texttt{Dg}}(\texttt{Sc}, H_r)$. If the Dg believes that the DU healthcare service is adequate, it pays the consultation fee corresponding to H_r for that DU via Sc. It will change the status of $paid$ to true.

F. Repeatable and Terminable Sharing

1) When the Dg finds that an outsourced EHR ciphertext CT_S needs to update the authorization list, i.e. add authorized

users or withdraw authorized users or both, the Dg recalculates partial ciphertext and uploads it to the CSP for replacement. The Dg also updates the CT_l authorization users deposited on the chain via Sc.

- $\mathsf{ReAuth}_{\mathsf{Dg}}(PP, S') \rightarrow c_{new}$. Assume S' is the updated set of authorized DUs. The Dg recalculates c_{new} based on the t existing at the previous Authorize and uploads c_{new} to the CSP along with H_l and S', where

$$S' = \{ID_i\}_{i=1}^{\tau \leq m}, c_{new} = h^{t \prod_{i=1}^{\tau} (\alpha + H_0(ID_i))}.$$

- $\mathsf{ReTran}_{\mathsf{CSP}}(CT_l, c_{new}) \rightarrow CT_S$. Upon receiving c_{new} from the Dg, the CSP retrieves the corresponding CT_l based on H_l. It sets $c_2 = c_{new}$ and obtains the updated CT_S.
- $\mathtt{UpAuth}_{\mathtt{CT_S}}(\mathtt{Sc}, H_l, S')$. The Dg uses Sc to specify H_l and publish the new set of authorized users S' to CB for deposit. $\mathtt{UpAuth}_{\mathtt{CT_S}}$ is only callable by Dgs, and if H_l has not been uploaded or does not belong to the Dg, the processing is interrupted. Otherwise, the algorithm collects the predefined coins and updates the structure of H_l with S' accordingly. Additionally, the $\mathtt{UpAuth}_{\mathtt{CT_S}}$ algorithm allows for the addition of parameter Tm to update the number of DUs that can request, with a corresponding coin charge.

2) When the Dg decides that an EHR ciphertext no longer needs to be accessed by DUs, the Dg updates the open status of the corresponding ciphertext of H_l on CB.

- $\mathtt{Close}_{\mathtt{CT_S}}(\mathtt{Sc}, H_l)$: This algorithm first checks whether H_l has been uploaded and whether it belongs to the Dg. If it passes the check, the access status Cls of the stored H_l structure is closed, indicating that the corresponding CT_S in the system will no longer accept access, and the CSP will no further allow any DU to access this CT_S.

V. Soundness and Security Analysis

In this section, we analyze that the proposed SeCoSe scheme is sound, and realizes semantic security and the design goals.

A. Soundness of SeCoSe

Theorem 1: The proposed SeCoSe scheme captures soundness.

Proof. For the SeCoSe scheme $\mathbf{\Lambda = (\Pi, \Gamma, \overleftarrow{\Pi}, \overleftarrow{\Gamma})}$, we have the following soundness proof:

In $\mathbf{\Pi}$ and $\mathbf{\overleftarrow{\Pi}}$, for any valid IBE ciphertext CT_{IBE} or CT_l or CT_r, a user with the correct private key can always successfully decrypt it to obtain msg or EHR or DTG and verify its validity. For any valid IBBE ciphertext CT_S, DUs with the correct private key and $ID \in S$ can always successfully decrypt it to obtain the EHR and verify its validity. The correctness of decryption can be inferred based on the correctness of the IBET scheme [35] and the correctness of the symmetric encryption algorithm used. Further details are skipped due to space limitations. The verifiability of decryption is based on 0^{ξ}, $H_l = H_2(C_R, M)$ and $H_r = H_2(C_G, M)$ added in $\mathbf{\Pi}$ and $\mathbf{\overleftarrow{\Pi}}$. If the symmetric encryption algorithm used is secure, 0^{ξ} will be appended to the decrypted msg or EHR or DTG. If H_2 used is strongly collision-resistant, then the validity of the symmetric key obtained during decryption can be verified, thus verifying the plaintext information obtained.

In $\mathbf{\Gamma}$ and $\mathbf{\overleftarrow{\Gamma}}$, for retrieval conditions that satisfy the Search criteria, Search can always return the matching set of retrieval IDs. This is based on the correctness of Algorithm 1 designed. For algorithms that publish and modify variables or states stored on-chain, if H_2 and H_3 are strongly collision-resistant, it will always successfully match the corresponding variables or data structures. This is based on the unique user addresses on the chain and the unique identifiers in the data structures.

B. Semantic Security of SRTIBE

Theorem 2: The proposed SRTIBE in SeCoSe has ciphertexts indistinguishability against the selective identity and chosen plaintext attack (IND-sID-CPA) under the GDDHE assumption [33].

Proof. To prove the theorem, we need to show that SRT-IBE is IND-sID-CPA secure for encrypted IBE ciphertext, transformed IBBE ciphertext, and repeatedly transformed IBBE ciphertext. The proof of the first two cases can be inferred from [35]. As for the detailed definition and proof of SRTIBE's IND-sID-CPA security in the case of repeatedly transformed IBBE ciphertext, it can be found in **Supplementary Materials C and D**.

C. Security Analysis of SeCoSe

We analyze that the proposed SeCoSe scheme enjoys the following security properties.

1) Data privacy protection and user privacy protection.

- The integrity and validity of shared data can be ensured through the soundness property of SeCoSe, as stated in Theorem 1. Unauthorized access to shared data can be guaranteed through rigorous admission scrutiny of CB and Theorem 2. The latter also ensures *Fine-grained access control*.
- The topic of the AMOP messages sent by DUs is $H_3(secTp)$ contained in EHR plaintext. The privacy of this message is ensured by the security of the SRTIBE scheme, the collision resistance of H_3, and the effectiveness of AMOP technology.
- DOs have anonymous addresses issued by the RA and do not participate in on-chain transactions. With the security of ciphertext transmission, DOs' private information such as name, contact information, and medical condition will not be disclosed or inferred. That is, the CSP that without access permission cannot learn about DOs' medical conditions from their anonymous identities, and authorized DUs who know DOs' medical conditions cannot associate them with the real or anonymous identities of DOs. Although addresses on the blockchain are actually pseudo-identities, only the authoritative RA can know the mapping relationship between users' real identities and anonymous addresses. Due to the complete

trustworthiness of the RA in the system, it is ensured that the real identities corresponding to the anonymous identities of users, especially Dgs and DUs, will not be disclosed, and their privacy information will not be associated with anonymous identities. Similarly, many works rely on blockchain to maintain the anonymity of entity identities [1], [38], [42], [43], [44].

2) Traceability.

- Dgs can trace the history of EHR ciphertext requests and accesses using the $\mathtt{Sta}_{\mathtt{CT_S}}$ algorithm on Sc, while DUs can trace the history of DTG ciphertext requests and accesses using the $\mathtt{Sta}_{\mathtt{CT_r}}$. The CSP is obligated to honestly execute $\mathtt{Sta}_{\mathtt{CT_S}}$ and $\mathtt{Sta}_{\mathtt{CT_r}}$ based on the ciphertext access, as any dishonesty will be discovered by DUs and Dgs (when the CSP is not colluding with DUs or Dgs).
- The traceability of transactions on CB is guaranteed by the tamper-proofing of blocks as explained in the next property. Malicious and erroneous behaviors can be traced due to the anti-tampering evidence of off-chain transactions on the chain.

3) Attack-resistance.

- By setting algorithm usage permissions, gas consumption limits, and coin collateral rules on the smart contract Sc, abuse and malicious resource consumption can be resisted.
- Block tampering attacks will be discarded due to blockchain rules. Collision attacks will be safeguarded due to the security of the selected hash function.
- In a CB where only a part of rigorously audited users have permission to participate in maintaining the blockchain and where PoW is not used as the consensus mechanism, 51% attacks can be avoided.

Moreover, any user who appears in the above attacks can be revealed by the RA as the real off-chain identity corresponding to the anonymous on-chain identity. For details regarding the proof of traceability and tamper-proofing, please refer to **Supplementary Material E**.

VI. Performance Evaluation

In this section, we analyze the proposed SeCoSe in terms of theoretical analysis and practical performance. Further discussions on SeCoSe can be found in **Supplementary Material F**.

A. Theoretical Analysis

In this subsection, we mainly analyze SRTIBE theoretically. Since the efficiency of smart contracts and AMOP depends on specific experimental deployments, we evaluate this part in Section VI-B. The notations used for comparison in the theoretical analysis are described in TABLE III. In the theoretical analysis, we only consider time-consuming cryptographic operations: one exponentiation in groups $\mathbb{G}$, $\mathbb{G}_T$, and a bilinear pairing operation, with time costs t_{e1}, t_{e2}, and t_p, respectively. Compared to other operations, these operations are much more expensive and need to be prioritized. Additionally, for the sake of comparison, we do not consider the overhead of symmetric encryption, aligning with previous works [10], [45].

TABLE III
Notations for Comparison Analysis

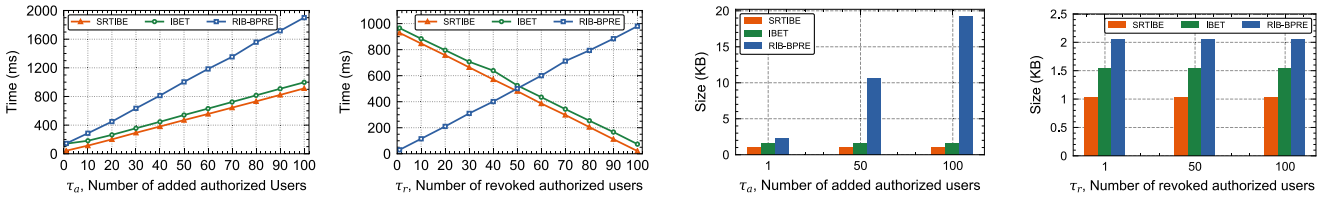
Notation	Description
$\lvert\mathbb{G}\rvert$, $\lvert\mathbb{G}_T\rvert$	size of an element in group $\mathbb{G}$, $\mathbb{G}_T$
t_{e1}, t_{e2}	time taken by one exponentiation in group $\mathbb{G}$, $\mathbb{G}_T$
t_p	time taken by a bilinear pairing operation
$\lvert RT\rvert$	size of token for repeated authorization and transformation
$\lvert CT_{S'}\rvert$	size of repeatedly transformed ciphertext
m	maximum number of users access to ciphertext
n_a	number of initial added authorized users
n	number of authorized users at decryption
k_r	maximum revocation number (in RIB-BPRE)
v	number of users already authorized
τ_a	number of added authorized users
τ_r	number of revoked authorized users
τ	number of renewed authorized users ($\tau = \tau_a + \tau_r$)

TABLE IV
Primary Storage Overhead and Computation Overhead in SRTIBE

Type	Storage cost	Algorithm	Computation cost
$\lvert PP\rvert$	$(m+4)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	Setup	$(m+2)t_{e1} + t_p$
$\lvert SK\rvert$	$\lvert\mathbb{G}\rvert$	KeyGen	t_{e1}
$\lvert CT_l\rvert$	$2\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	Enc	$2t_{e1} + t_{e2}$ or $4t_{e1} + t_{e2}$
$\lvert CT_r\rvert$	$\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	Authorize	$(n_a+4)t_{e1} + t_{e2}$
$\lvert d_{Dg\to S}\rvert$	$4\lvert\mathbb{G}\rvert$	Transform	t_p
$\lvert CT_S\rvert$	$4\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	DecVer	t_p or $(n-1)t_{e1} + 3t_p$

TABLE IV presents the primary storage and computation overhead in the SRTIBE scheme, which is mainly for the phases of system initialization, encryption, decryption, and initial authorization and transformation. Let $|PP|$, $|SK|$, $|CT_l|$, $|CT_r|$, $|d_{Dg\to S}|$, and $|CT_S|$ denote the sizes of public parameters, user private keys, CT_l, CT_r, tokens for initial authorization, and ciphertext after transformation, respectively. In TABLE IV, most of the overhead of SRTIBE remains the same as that of IBET [35], except that $|PP|$ has 3 more $|\mathbb{G}|$. It can be observed that the Setup overhead executed by RA and the generated $|PP|$ overhead are related to the maximum authorized users m. The time consumption of Authorize executed by a Dg is related to the number of initially added authorized users n_a. The DecVer cost performed by DUs is related to the remaining authorized user count n. The Enc cost of message transmission between DO and Dg is $2t_{e1} + t_{e2}$, and the DecVer cost is tp. The encryption and decryption time consumed by DUs towards Dg follows the same pattern. Encrypting the ciphertext with scalable permissions by DO requires a consumption of $4t_{e1} + t_{e2}$, but subsequent authorization operation (Authorize) is carried out by the delegated Dg, while the transformation task (Transfrom) is performed by the CSP.

TABLE V compares the storage and computation overhead of the SRTIBE scheme with RBE [46], RIB-BPRE [37], and IBET [35] during repeated authorization and transformation, which is an important part of achieving dynamic and flexible EHR sharing. RBE only supports adding or revoking one user's permission at a time, so the storage and computation costs of generating tokens are related to the number of users added (τ_a) or revoked (τ_r). However, since RBE encrypts



(a) Time for adding users (b) Time for revoking users (c) Size for adding users (d) Size for revoking users

Fig. 6. Experimental performance of repeated authorization and transformation.

the data only after confirming the updated authorization and completing the transformation process, the storage overhead of $|CT_{S'}|$ in RBE is relatively small. The cost of adding authorization in RIB-BPRE is influenced by the maximum number of possible revocations, and the time cost of revocation is only related to the number of users being revoked. However, RIB-BPRE incurs a relatively higher time cost for the transformation process. In the original scheme, IBET does not mention the operation of repeated ciphertext transformations. For comparability, in this paper, the token generation and ciphertext transformation steps from IBE to IBBE form of IBET are repeated to support repeated authorization and transformation. The token of SRTIBE for repeated authorization and transformation is the partial ciphertext c_{new} related to the new authorization list, occupying only one $|\mathbb{G}|$, and requiring a time cost of $(v+\tau_a+1)t_{e1}$ or $(v-\tau_r+1)t_{e1}$. In contrast, IBET needs to recalculate a four $|\mathbb{G}|$ token for each transformation, with a time cost of $(v+\tau_a+4)t_{e1}+t_{e2}$ or $(v-\tau_r+4)t_{e1}+t_{e2}$. Moreover, there are cryptographic operations in SRTIBE only in the repeated authorization phase, i.e., in the authorization token generation part of TABLE V. In the repeated transformation phase of SRTIBE, only the ciphertext stored in the CSP is partially replaced, and therefore the transformation cost of SRTIBE given in TABLE V is 0. Consequently, SRTIBE has lower storage and computation overheads than IBET in repeated authorization and transformation.

B. Experimental Analysis

1) Experiment Setting: We conducted comprehensive experiments to evaluate the performance of the proposed SeCoSe, all of which were performed on a PC running Ubuntu 18.04 (Intel(R) Xeon(R) E3-1230 v5 CPU @3.40GHz; 16G RAM). To implement the SRTIBE of SeCoSe, we utilized the Java Pairing Based Cryptography (JPBC) library. Our experiments employed the A-type elliptic curve with 160-bit group order in JPBC, which is also known as $E/F_p : y^2 = x^3 + x$. For the implementation of SeCoSe functions and transactions on the blockchain, we utilized platforms that support Java SDK interaction. Specifically, we used the Solidity[1] language to write smart contracts and leveraged the open-source FISCO BCOS[2] for deployment testing. BCOS provides support for the Solidity language, the Java SDK, and the AMOP protocol.

2) Experiment Results:

a) Data sharing efficiency: Since the primary overhead of SRTIBE is basically the same as that of IBET [35] as shown in TABLE IV, only the practical efficiency of the SRTIBE proposed in SeCoSe is compared with RIB-BPRE [37] and IBET [35] in terms of repeated authorization and transformation. Each scheme utilizes key encapsulation technology to encrypt the data, employing AES for symmetric encryption in the SEnc and SDec algorithms. In the experiment, we tested two scenarios of repeated authorization and transformation: adding 1-100 authorized users when the number of authorized users is 1 and revoking 1-100 authorized users when the number of authorized users is 101. The time consumption of the tests, as shown in Fig. 6(a) and 6(b), includes the time for recalculating authorization tokens and transforming ciphertexts. The storage size in Fig. 6(c) and 6(d) includes the storage space required for authorization tokens and transformed ciphertexts. Due to the quadratic growth in time overhead and linear growth in storage overhead with the increase of authorized users in RBE [46], its costs are significantly higher compared to other schemes. Therefore, we do not display the costs of RBE in Fig. 6.

As shown in Fig. 6(a), when adding authorized users, our SRTIBE outperforms other schemes in terms of efficiency. In Fig. 6(b), the time consumption of SRTIBE and IBET decreases linearly with τ_r, and SRTIBE's time cost is smaller. This is because the time for repeated authorization tokens in SRTIBE and IBET depends on the remaining number of authorized users $(v - \tau_r)$, which is in contrast to RIB-BPRE's linear growth with revocation count τ_r. From Fig. 6(c) and 6(d), it can be seen that the storage space required for the update process of SRTIBE and IBET is fixed, and SRTIBE has a smaller storage overhead. In our experiments, the maximum number of revocations (k_r) defined in RIB-BPRE's process of adding authorized users is equal to the maximum number of authorizations (m) defined in SRTIBE's initialization, where m equals the number of added authorized users (τ_a). Therefore, the storage space for adding authorized users in RIB-BPRE increases linearly with τ_a.

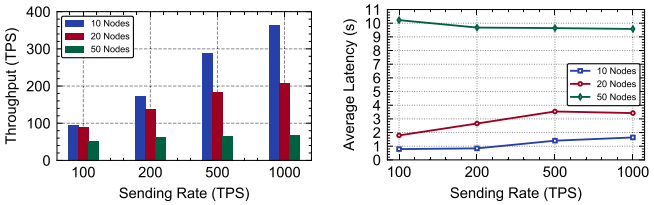
Furthermore, in SRTIBE, since the DU-to-Dg direction is determined to be a one-to-one IBE encryption, compared with the Dg-to-DUs direction that supports flexible repeated authorization and transformation, DU-to-Dg direction has faster encryption and decryption speed, which helps the Dg get professional service advice from the DU in a timely manner.

b) Blockchain transaction efficiency and gas cost: The testing of blockchain efficiency depends on the proposed system model, the task requirements of the application scenario, and the actual deployment platform, resulting in a general lack of comparability between blockchain efficiencies in different

[1]https://soliditylang.org/
[2]http://www.fiscobcos.org/

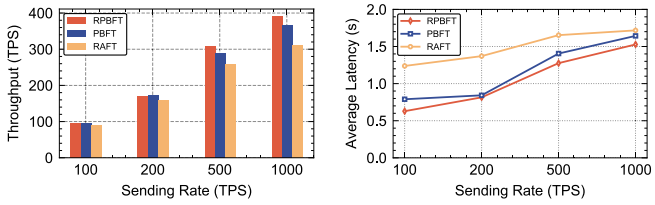
TABLE V
STORAGE OVERHEAD AND COMPUTATION OVERHEAD OF REPEATED AUTHORIZATION AND TRANSFORMATION

Scheme	Storage			Computation		
	$\|RT\|$		$\|CT_{S'}\|$	Authorization token generation		Transformation
	add	revoke		add	revoke	
RBE [46]	$4\tau\|\mathbb{G}\|$	$4\tau\|\mathbb{G}\|$	$3\|\mathbb{G}\|$	$(\tau_a v+3+\frac{\tau_a(\tau_a+3)}{2})t_{e1}+t_{e2}$	$(\tau_r v+3+\frac{\tau_r(\tau_r-1)}{2})t_{e1}+t_{e2}$	$3t_{e1}+t_{e2}$
RIB-BPRE [37]	$(5+k_r)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$5\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$4\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$(v+\tau_a+k_r+5)t_{e1}+t_{e2}$	$(2\tau_r+2)t_{e1}+t_p$	$t_{e2}+2t_p$
IBET [35]	$4\|\mathbb{G}\|$	$4\|\mathbb{G}\|$	$4\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(v+\tau_a+4)t_{e1}+t_{e2}$	$(v-\tau_r+4)t_{e1}+t_{e2}$	t_p
SRTIBE	$\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$4\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(v+\tau_a+1)t_{e1}$	$(v-\tau_r+1)t_{e1}$	0



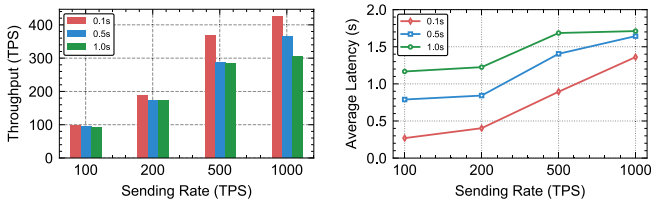
(a) Throughput (b) Average latency

Fig. 7. Blockchain performance with varying the number of nodes.



(a) Throughput (b) Average latency

Fig. 8. Blockchain Performance with varying consensus mechanisms.



(a) Throughput (b) Average latency

Fig. 9. Blockchain Performance with varying block generation time.

schemes. Therefore, most related works omit the testing and comparison of blockchain transaction efficiency [2], [4], [11], [17], [47]. We analyze the performance of SeCoSe under different CB parameter configurations such as the number of nodes, consensus mechanism, and block generation time. Specifically, we take a coin pledge transaction in SeCoSe as an example, make all nodes participate in consensus, set the number of transactions to be processed to 1000, and each block contains at most 1000 transactions. Then we vary the sending rate under different CB configurations, i.e., 100, 200, 500, and 1000 transactions per second. Each case is tested for 5 rounds and averaged, and the performance is finally measured in terms of throughput and average latency.

By changing the number of CB nodes, consensus mechanism, and block generation time, different experimental comparison results are obtained in Fig. 7-9, and other variables are strictly controlled in each experiment test. From Fig. 7-9, we observe that the throughput and latency rise as the sending rate increases, which is consistent with the results observed by [25]. In particular, in the test case of 50 nodes, there is no significant trend in throughput and latency with different sending rates, because the massive network communication with a large number of CB nodes has become the main bottleneck limiting the throughput and latency. Also, it can be observed from Fig. 7 that the performance of SeCoSe decreases as the node size increases. Fig. 8 shows that SeCoSe has better performance under the setting of consensus mechanism as RPBFT, which is due to the RPBFT proposed by BCOS decoupling the consensus algorithm complexity from the consensus node size and improving the scalability of the blockchain system. Fig. 9 shows that as the generation time of blocks decreases, the performance improves. Especially when it is shortened to 0.1 s, the throughput and latency are optimized more substantially.

Furthermore, we measure the gas consumption of the smart contract designed for SeCoSe, as shown in TABLE VI and Fig. 10 (short algorithm name). It can be seen that deploying the smart contract `Sc` on CB requires a huge amount of gas, but this process only needs to be executed once. There is not much difference in the gas used between registering one

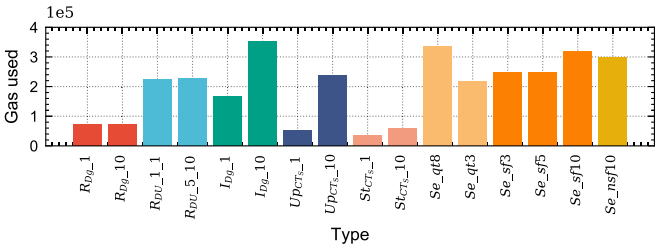
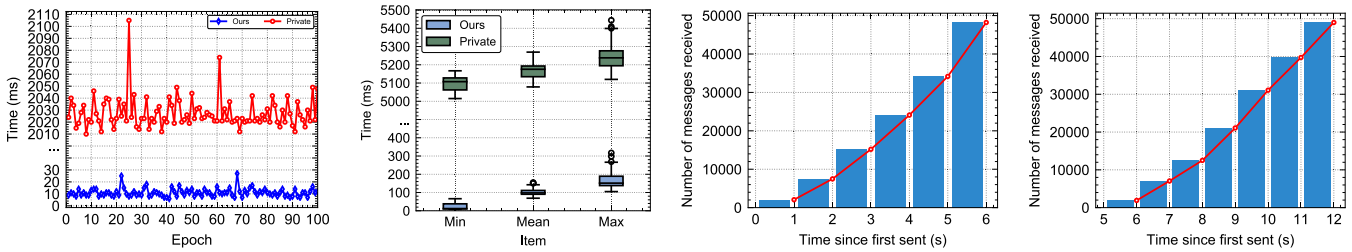


Fig. 10. Comparison of gas consumption.

TABLE VI
SMART CONTRACT GAS COST

Algorithm	Gas used	Algorithm	Gas used	Algorithm	Gas used
`deploy`	2,189,469	$\texttt{UpAuth}_{CT_S}$_1	50,621	`Search_qt8`	336,705
$\texttt{Reg}_{Dg}$_1	71,538	$\texttt{UpAuth}_{CT_S}$_10	236,966	`Search_qt3`	216,645
$\texttt{Reg}_{Dg}$_10	73,609	$\texttt{Sta}_{CT_S}$_1	34,985	`Search_sf3`	248,035
$\texttt{Reg}_{DU}$_1_1	223,061	$\texttt{Sta}_{CT_S}$_10	60,564	`Search_sf5`	248,658
$\texttt{Reg}_{DU}$_5_10	226,651	$\texttt{Issue}_{DU}$	105,370	`Search_sf10`	318,007
$\texttt{Issue}_{Dg}$_1	167,379	$\texttt{Sta}_{CT_r}$	28,632	`Search_nsf10`	297,507
$\texttt{Issue}_{Dg}$_10	353,724	$\texttt{FeBack}_{Dg}$	43,647	$\texttt{Close}_{CT_S}$	29,185



(a) 1-to-1 communication (b) 1-to-49 communication (c) 1000-to-49 communication of ours (d) 1000-to-49 communication of private topic

Fig. 11. The efficiency of message transmission.

attribute ($\texttt{Reg}_{\texttt{Dg}}\texttt{\_1}$) and ten attributes ($\texttt{Reg}_{\texttt{Dg}}\texttt{\_10}$) for Dg. The registration of DUs with different numbers of attributes follows a similar pattern, although registering a DU involves operations such as creating structures and storing mappings, resulting in higher gas consumption. The gas consumption of the $\texttt{Issue}_{\texttt{Dg}}\texttt{\_1}$ algorithm for S of length 1 and 10 is 167,379 and 353,724, respectively. $\texttt{UpAuthor}_{\texttt{CTs}}$ and $\texttt{Sta}_{\texttt{CTs}}$ also follow this pattern, but as the number of updates increases, the gas consumption of $\texttt{UpAuthor}_{\texttt{CTs}}$ grows more significantly due to the algorithm's involvement in changing the values of the states already stored on the blockchain. Other algorithms have relatively small gas consumption and lower execution costs.

Note that there are some view-type algorithms on SeCoSe's smart contract, particularly the `Search` algorithm. These algorithms do not need to be synchronized or sent to other nodes, and they can quickly return results locally without consuming gas when not called by other smart contracts. In TABLE VI, we still provide the gas consumption of `Search` to indicate the computational overhead and resource consumption. Specifically, let dis in `Search` take 3. When the `Search` fails to find DUs that meet the conditions in the eighth matching condition of cds (`Search_qt8`), the gas used is greater than when it exits in the third matching condition (`Search_qt3`). Therefore, we recommend putting important attributes such as DUs' reputation at the beginning of the attribute array to filter out non-compliant DUs earlier. When `Search` is successful, the gas consumption increases with the number of DUs found, and the gas consumption of finding 10 DUs with specified conditions (`Search_sf10`) is greater than the gas consumption of finding 10 DUs without constraints (`Search_nsf10`). However, `Search_nsf10` still incurs significant gas consumption, indicating that SeCoSe encourages finding a small number of suitable DUs by imposing limiting constraints.

c) Message transmission efficiency: We test the efficiency of AMOP for message transmission here because it does not rely on blockchain consensus, and thus it has a different performance compared to blockchain transaction efficiency. In the SeCoSe system, we only use the public topic mode of AMOP for message notification. When transmitting EHR ciphertexts, the EHR includes an $H_3(secTp)$, which is subsequently used as the designated topic name for DUs to send back AMOP messages, privately notifying the specific Dg only. This eliminates the need for the more costly private topic mode of AMOP to implement private specified notifications for returned messages. Moreover, the use of AMOP does not rely on blockchain transactions and consensus, reducing latency. Therefore, the proposed SeCoSe achieves a balance between AMOP transmission efficiency and security, which is an improvement of SeCoSe in applying AMOP for message transmission. Here, we compare the efficiency of transmitting messages using public topics and private topics. As shown in Fig. 11(a), we test the time delay of one hundred rounds of 1-to-1 transmission of AMOP messages, which is often used in the case of notifying a specified Dg after the DU completes a diagnosis. It can be seen that the public topic approach we used can quickly complete the notification of messages in an average of 15 ms, while the private topic approach takes more than 2 s. Fig. 11(b) shows a comparison of the minimum, average, and maximum time for 49 nodes to receive a message from one node, which is the result of averaging 100 tests. We observe that, under the private topic mode, approximately 5 s of waiting time for signature verification and transmission is required to ensure that the majority of subscribing nodes receive the message. On the other hand, the public topic mode is capable of delivering the message to all subscribing nodes within an average of 200 ms. We also simulated a scenario where 1000 nodes send messages to 49 nodes to perform a pressure test on the AMOP function. The result is shown in Fig. 11(c)-11(d) after averaging 10 tests. Both public and private topics can finish sending 1000 messages within 0.8 s. In particular, the public topic approach we used can complete the transmission of 49,000 messages in 6 s, while the private topic takes 12 s.

TABLE VII
STORAGE OVERHEAD OF DIFFERENT COMMUNICATION MODES

Item	Private AMOP	Ours
toml_send	284	145
toml_rece	311	151
public_key	174	0
private_key	321	0
hash value	0	64
Sum(nDgs → mDUs)	284n+311m	145n+151m
Sum(mHC → nPnF)	274m+311n+174m+321n	145m+151n+64n

In addition, we also compare the storage overhead (TABLE VII) of public and private topics. Private topics require additional configuration file storage overhead

(toml_send and toml_rece) for both the sender and the receiver because of the use of public and private keys (public_key and private_key). To achieve the security level of the private topic mode, we use a hash as the topic name that is only known to specific Dg and DU in the public topic mode. Therefore, additional hash value storage overhead is required. Overall, our utilization of AMOP demonstrates higher efficiency compared to the message transmission approach involving private topics, while still achieving the same private message transmission effect.

VII. Conclusion

In this paper, we propose SeCoSe to bridge the gap for searchable and communicable healthcare service seeking in flexible and secure EHR sharing. SeCoSe aims to provide technical support for the practical application of blockchain-based EHR sharing systems. To achieve this goal, we first propose a method called SRTIBE to enable fine-grained access control and dynamic authorization updates. Then, attribute-identity mapping contracts and evidence-based contracts on the blockchain are designed to enable healthcare providers to be searchable and transactions to be traceable. Besides, AMOP technology is employed to achieve secure bidirectional message transmission. Our security analysis demonstrates that SeCoSe meets the security requirements. Detailed performance evaluation and analysis show the effectiveness of SeCoSe in service provider retrieval, dynamic authorization, transaction processing, and online communication. As a future research effort, we plan to evaluate SeCoSe's performance in real-world applications through realistic healthcare scenarios and test SeCoSe's generality in similar IoT scenarios.

References

[1] R. Zhang, R. Xue, and L. Liu, "Security and privacy for healthcare blockchains," *IEEE Trans. Services Comput.*, vol. 15, no. 6, pp. 3668–3686, Dec. 2022.

[2] S. Xu et al., "A secure EMR sharing system with tamper resistance and expressive access control," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 53–67, Jan./Feb. 2023.

[3] S. Liu, L. Chen, G. Wu, H. Wang, and H. Yu, "Blockchain-backed searchable proxy signcryption for cloud personal health records," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 3210–3223, Sep./Oct. 2023.

[4] E. Zaghloul, T. Li, M. W. Mutka, and J. Ren, "MABE: Distributed multilevel attribute-based EMR management and applications," *IEEE Trans. Services Comput.*, vol. 15, no. 3, pp. 1592–1605, May 2022.

[5] A. Dagliati, A. Malovini, V. Tibollo, and R. Bellazzi, "Health informatics and EHR to support clinical research in the COVID-19 pandemic: An overview," *Briefings Bioinf.*, vol. 22, no. 2, pp. 812–822, Mar. 2021.

[6] S. Vilendrer et al., "Rapid deployment of inpatient telemedicine in response to COVID-19 across three health systems," *J. Amer. Med. Inform. Assoc.*, vol. 27, no. 7, pp. 1102–1109, Jul. 2020.

[7] M. Wang, Y. Guo, C. Zhang, C. Wang, H. Huang, and X. Jia, "MedShare: A privacy-preserving medical data sharing system by using blockchain," *IEEE Trans. Services Comput.*, vol. 16, no. 1, pp. 438–451, Jan./Feb. 2023.

[8] H. S. G. Pussewalage and V. Oleshchuk, "A delegatable attribute based encryption scheme for a collaborative e-health cloud," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 787–801, Mar. 2023.

[9] J. Wang, X. Yin, J. Ning, S. Xu, G. Xu, and X. Huang, "Secure updatable storage access control system for EHRs in the cloud," *IEEE Trans. Services Comput.*, vol. 16, no. 4, pp. 2939–2953, Jul./Aug. 2023.

[10] Y. Yang, X. Liu, R. H. Deng, and Y. Li, "Lightweight sharable and traceable secure mobile health system," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 1, pp. 78–91, Jan./Feb. 2020.

[11] F. Li, K. Liu, L. Zhang, S. Huang, and Q. Wu, "EHRChain: A blockchain-based EHR system using attribute-based and homomorphic cryptosystem," *IEEE Trans. Services Comput.*, vol. 15, no. 5, pp. 2755–2765, Sep. 2022.

[12] Y. Miao et al., "Time-controllable keyword search scheme with efficient revocation in mobile e-health cloud," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 3650–3665, May 2024.

[13] J. Zhang, Y. Yang, X. Liu, and J. Ma, "An efficient blockchain-based hierarchical data sharing for healthcare Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7139–7150, Oct. 2022.

[14] S. Li et al., "HealthFort: A cloud-based ehealth system with conditional forward transparency and secure provenance via blockchain," *IEEE Trans. Mobile Comput.*, vol. 22, no. 11, pp. 6508–6525, Nov. 2023.

[15] G. S. Aujla and A. Jindal, "A decoupled blockchain approach for edge-envisioned IoT-based healthcare monitoring," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 491–499, Feb. 2021.

[16] M. Shen, J. Duan, L. Zhu, J. Zhang, X. Du, and M. Guizani, "Blockchain-based incentives for secure and collaborative data sharing in multiple clouds," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1229–1241, Jun. 2020.

[17] B. Chen, T. Xiang, D. He, H. Li, and K. R. Choo, "BPVSE: Publicly verifiable searchable encryption for cloud-assisted electronic health records," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 3171–3184, 2023.

[18] G. Lin, H. Wang, J. Wan, L. Zhang, and J. Huang, "A blockchain-based fine-grained data sharing scheme for e-healthcare system," *J. Syst. Archit.*, vol. 132, Nov. 2022, Art. no. 102731.

[19] S. Singh, S. Rathore, O. Alfarraj, A. Tolba, and B. Yoon, "A framework for privacy-preservation of IoT healthcare data using federated learning and blockchain technology," *Future Gener. Comput. Syst.*, vol. 129, pp. 380–388, Apr. 2022.

[20] H. Kordestani, K. Barkaoui, and W. Zahran, "HapiChain: A blockchain-based framework for patient-centric telemedicine," in *Proc. IEEE 8th Int. Conf. Serious Games Appl. Health (SeGAH)*, BC, BC, Canada, Aug. 2020, pp. 1–6.

[21] L. Liu, X. Li, M. H. Au, Z. Fan, and X. Meng, "Metadata privacy preservation for blockchain-based healthcare systems," in *Database Systems for Advanced Applications*. Cham, Switzerland: Springer, 2022, pp. 404–412.

[22] WE Contributors. *What Is a General Practitioner?* Accessed: Oct. 29, 2023. [Online]. Available: https://www.webmd.com/a-to-z-guides/what-is-a-general-practitioner

[23] Z. Deng, Z. Hong, W. Zhang, R. Evans, and Y. Chen, "The effect of online effort and reputation of physicians on patients' choice: 3-wave data analysis of China's good doctor website," *J. Med. Internet Res.*, vol. 21, no. 3, Mar. 2019, Art. no. e10170.

[24] N. C. Zwijnenberg et al., "Patients' need for tailored comparative health care information: A qualitative study on choosing a hospital," *J. Med. Internet Res.*, vol. 18, no. 11, p. e297, Nov. 2016.

[25] A. P. Singh et al., "A novel patient-centric architectural framework for blockchain-enabled healthcare applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5779–5789, Aug. 2021.

[26] L. Chen, W.-K. Lee, C.-C. Chang, K.-K. R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future Gener. Comput. Syst.*, vol. 95, pp. 420–429, Jun. 2019.

[27] X. Liu, X. Yang, Y. Luo, and Q. Zhang, "Verifiable multikeyword search encryption scheme with anonymous key generation for medical Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22315–22326, Nov. 2022.

[28] X. Tang, C. Guo, K.-K.-R. Choo, Y. Liu, and L. Li, "A secure and trustworthy medical record sharing scheme based on searchable encryption and blockchain," *Comput. Netw.*, vol. 200, Dec. 2021, Art. no. 108540.

[29] Q. Wang, C. Lai, R. Lu, and D. Zheng, "Searchable encryption with autonomous path delegation function and its application in healthcare cloud," *IEEE Trans. Cloud Comput.*, vol. 11, no. 1, pp. 879–896, Jan./Mar. 2023.

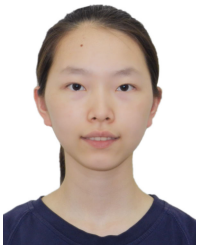
[30] M. Du, Q. Chen, J. Chen, and X. Ma, "An optimized consortium blockchain for medical information sharing," *IEEE Trans. Eng. Manag.*, vol. 68, no. 6, pp. 1677–1689, Dec. 2021.

[31] M. Ahmed et al., "A blockchain-based emergency message transmission protocol for cooperative VANET," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19624–19633, Oct. 2022.

[32] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO*, vol. 2139. Berlin, Germany: Springer, 2001, pp. 213–229.

[33] C. Delerablée, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *Advances in Cryptology—ASIACRYPT*, vol. 4833. Berlin, Germany: Springer, pp. 200–215.
[34] *Advance Messages Onchain Protocol—FISCO BCOS FISCO BCOS Documentation*. Accessed: Oct. 29, 2023. [Online]. Available: https://fisco-bcos-documentation-en.readthedocs.io/en/latest/docs/AMOP/README.html#
[35] H. Deng et al., "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3168–3180, 2020.
[36] J. Sun, G. Xu, T. Zhang, X. Yang, M. Alazab, and R. H. Deng, "Verifiable, fair and privacy-preserving broadcast authorization for flexible data sharing in clouds," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 683–698, 2023.
[37] C. Ge, Z. Liu, J. Xia, and L. Fang, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Trans. Depend. Secure Comput.*, vol. 18, no. 3, pp. 1214–1226, May 2021.
[38] B. D. Deebak and S. O. Hwang, "Healthcare applications using blockchain with a cloud-assisted decentralized privacy-preserving framework," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5897–5916, May 2024.
[39] C. Zhang, M. Zhao, L. Zhu, W. Zhang, T. Wu, and J. Ni, "FRUIT: A blockchain-based efficient and privacy-preserving quality-aware incentive scheme," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3343–3357, Dec. 2022.
[40] Y. Yi, Y. Yang, K. Cheng, Y. Wu, and X. Wang, "Information dissemination with service-oriented incentive mechanism in industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 16897–16907, Sep. 2022.
[41] Y. Liu, Z. Liu, Q. Zhang, J. Su, Z. Cai, and X. Li, "Blockchain and trusted reputation assessment-based incentive mechanism for healthcare services," *Future Gener. Comput. Syst.*, vol. 154, pp. 59–71, May 2024.
[42] Y. Liu et al., "A blockchain-based decentralized, fair and authenticated information sharing scheme in zero trust Internet-of-Things," *IEEE Trans. Comput.*, vol. 72, no. 2, pp. 501–512, Feb. 2023.
[43] S. Liu, Y. Chai, L. Hui, and W. Wu, "Blockchain-based anonymous authentication in edge computing environment," *Electronics*, vol. 12, no. 1, p. 219, Jan. 2023.
[44] H. Huang, P. Zhu, F. Xiao, X. Sun, and Q. Huang, "A blockchain-based scheme for privacy-preserving and secure sharing of medical data," *Comput. Secur.*, vol. 99, Dec. 2020, Art. no. 102010.
[45] J. Shen, P. Zeng, K. R. Choo, and C. Li, "A certificateless provable data possession scheme for cloud-based EHRs," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1156–1168, 2023.
[46] L. Zhou, V. Varadharajan, and M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 12, pp. 1947–1960, Dec. 2013.
[47] C. Lin, X. Huang, and D. He, "Efficient blockchain-based electronic medical record sharing with anti-malicious propagation," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 3294–3304, Sep./Oct. 2023.

Zhihuang Liu received the B.E. and M.S. degrees from the College of Computer and Data Science, Fuzhou University, in 2020 and 2023, respectively. He is currently pursuing the Ph.D. degree with the College of Computer, National University of Defense Technology. His research interests include blockchain, the IoT security, and applied cryptography.



Ling Hu received the bachelor's degree in computer science from the National University of Defense Technology, China, where she is currently pursuing the master's degree. Her research interests include visual analysts, data privacy, and artificial intelligence.

Zhiping Cai (Member, IEEE) received the B.Eng., M.A.Sc., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), China, in 1996, 2002, and 2005, respectively. He is currently a Full Professor with the College of Computer, NUDT. His current research interests include artificial intelligence, network security, and big data. He is a Senior Member of the CCF.

Ximeng Liu (Senior Member, IEEE) received the B.Sc. degree in electronic engineering and the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2010 and 2015, respectively. He is currently a Full Professor with the College of Computer and Data Science, Fuzhou University. He was a Research Fellow with the School of Information System, Singapore Management University, Singapore. He has published more than 250 research articles, including IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and IEEE INTERNET OF THINGS JOURNAL. His research interests include cloud security, applied cryptography, and big data security. He was awarded the "Minjiang Scholars" Distinguished Professor, "Qishan Scholars" in Fuzhou University, and ACM SIGSAC China Rising Star Award in 2018. He served as a program committee for several conferences, such as the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and the 2017 IEEE Global Communications Conference. He served as a Lead Guest Editor for WCMC, IJDSN, and ETT.

Yanhua Liu received the B.S. and M.S. degrees from the College of Computer and Data Science, Fuzhou University, China, in 1996 and 2003, respectively, and the Ph.D. degree from the College of Physics and Information Engineering, Fuzhou University, in 2016. He is currently an Associate Professor and a Researcher with Fujian Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University. His research interests are intelligent computing, computer security, and big data. His research work has won several government awards.