Hindawi
Security and Communication Networks
Volume 2022, Article ID 8293716, 16 pages
https://doi.org/10.1155/2022/8293716

Research Article

Blockchain-Empowered Secure and Privacy-Preserving Health Data Sharing in Edge-Based IoMT

Xueli Nie,[1] **Aiqing Zhang**,[1,2] **Jindou Chen**,[1,2] **Youyang Qu**,[3] **and Shui Yu**[4]

[1]*School of Physics and Electronic Information, Anhui Normal University, Wuhu 241002, China*
[2]*Anhui Provincial Engineering Laboratory on Information Fusion, Wuhu 241002, China*
[3]*Deakin Blockchain Innovation Lab, School of Information Technology, Deakin University, Geelong, Australia*
[4]*School of Computer Science, University of Technology Sydney, Ultimo, Australia*

Correspondence should be addressed to Aiqing Zhang; aqzhang2006@163.com

Received 12 September 2021; Revised 8 December 2021; Accepted 13 January 2022; Published 21 February 2022

Academic Editor: Ding Wang

Health data sharing, as a booming demand, enables the patients with similar symptoms to connect with each other and doctors to obtain the medical history of patients. Health data are usually collected from edge-based Internet of medical things (IoMT) with devices such as smart wearable devices, smart watches, and smartphones. Since health data are highly private and have great financial value, adversaries ceaselessly launch diverse attacks to obtain private information. All these issues pose great challenges to health data sharing in edge-based IoMT scenarios. Existing research either lacks comprehensive consideration of privacy and security protection or fails to provide a proper incentive mechanism, which expels users from sharing data. In this study, we propose a novel blockchain-assisted data sharing scheme, which allows secure and privacy-preserving profile matching. A bloom filter with hash functions is designed to verify the authenticity of keyword ciphertext. Key-policy attribute-based encryption (KP-ABE) algorithm and smart contracts are employed to achieve secure profile matching. To incentivize users actively participating in profile matching, we devise an incentive mechanism and construct a two-phase Stackelberg game to address pricing problems for data owners and accessing problems of data requesters. The optimal pricing mechanism is specially designed for encouraging more users to participate in health data sharing and maximizing users' profit. Moreover, security analysis illustrates that the proposed protocol is capable of satisfying various security goals, while performance evaluation shows high scalability and feasibility of the proposed scheme in edge-based IoMT scenarios.

1. Introduction

Internet of medical things (IoMT) can improve healthcare service quality by sharing health data among users and realize remote health care. Generally, profile matching is the prerequisite of health data sharing. For example, patients with similar symptoms may want to connect with others, exchange their experiences, and broaden the understanding of the symptoms helping them to get early diagnosis or better treatments in time [1, 2].

Since IoMT devices are usually resource constraint, edge computing is introduced into the system to improve the efficiency [3, 4]. Due to privacy-sensitive nature of health data, data security and privacy preservation are crucial in edge-based IoMT system. Although the existing schemes use cryptographic algorithms to achieve security, there are still some threats because the edge is a semi-trusted center and also faces a single point of failure. In addition, these works failed to take incentive problem into consideration for users. Even though incentive mechanism is considered in [5, 6], the security issues in profile matching have hardly been solved.

Fortunately, emerging blockchain technology is tamper-proof, decentralized, transparent, anonymous, and autonomous, which is a promising solution for the aforementioned problems [7]. Although blockchain is conducive to profile matching and health data sharing, there are still three following challenges: (1) how to realize privacy preservation through profile matching in blockchain environment? (2)

How to ensure that the authenticity of the keyword is selected from a valid keyword set while not revealing users' private information? (3) How to design an incentive mechanism for motivating users to take an active part in the system?

To solve the above challenges, we put forward a secure and privacy-preserving profile matching scheme by employing key-policy attribute-based (KP-ABE) keyword search and bloom filter in edge-based IoMT. Blockchain is adopted to store the keyword ciphertext. It ensures that requesters access the desired data and protect data privacy. Moreover, the bloom filter verifies the authenticity of keyword ciphertext using various hash functions without disclosing any sensitive information. We design an optimal pricing mechanism to incentivize data owners and data requesters to participate in this system.

The major contributions of the proposed scheme are summarized as follows.

We construct a new blockchain-based profile matching framework in edge-based IoMT with security and privacy preservation. We design a bloom filter based on many hash functions to verify whether the keyword is selected from a valid keyword set before storing the keyword ciphertext in blockchain. In this way, the computational cost is significantly reduced.

We present a key-policy attribute-based keyword search and bloom filter profile matching protocol in a blockchain-enabled edge-based IoMT. Only when data users' attribute set satisfies the access structure set, users are able to access the desired data. Data owners' original data are stored in local server, while the corresponding keyword ciphertext is kept in blockchain verified by a bloom filter. It greatly improves search efficiency.

We devise an optimal pricing mechanism to motivate users to actively participate in the system. The data owners take part in pricing and setting payment for matching data according to pricing list based on an optimal pricing mechanism. In particular, it provides an economic approach to analyze the incentive mechanism. By utilizing game-based optimal pricing mechanism, data owners and data requesters can obtain their maximum benefits. Furthermore, we implement smart contracts on the Ethereum platform and conduct extensive evaluations to demonstrate the superiority of the proposed scheme.

The structure of the study is organized as follows. Related works about health data sharing with edge computing blockchain-based profile matching are elaborated in Section 2. Section 3 introduces preliminaries of the work. Section 4 presents system architecture, workflow, security threats, and design goals. Section 5 designs the detailed profile matching protocol for the proposed blockchain, proposes an optimal pricing mechanism to incentive data owner and data requester actively participating in data sharing, and presents the smart contracts. Then, Section 6 analyzes how our scheme achieves the security goals. Section 7 carries out the proposed scheme on Ethereum platform to estimate the performance of protocol and smart contracts. Finally, Section 8 concludes this work.

2. Related Work

In this section, we introduce existing relevant researches on health data sharing with edge computing, profile matching, and incentive mechanisms.

2.1. Health Data Sharing with Edge Computing. Privacy issues emerge with the fast development of edge computing and IoMT [8, 9]. The edge node was the closest to the restricted resource devices and had strong computing power. Therefore, some edge-based healthcare systems were proposed [10]. Pustokhina et al. [11] presented an effective scheme in edge computing-enabled IoMT system. The IoMT devices sensed patient's data and transferred the captured data to edge computing. Aiming at guaranteeing data security and flexible right control simultaneously, an efficient attribute-based encryption (ABE) scheme was proposed to outsource part of encryption and decryption to edge nodes [12]. It supported attribute updates and reduced the computing cost of resource-constrained devices for edge-enabled smart health care. To rely on a trusted center, Akkaoui et al. [13] proposed a blockchain-based secure health data sharing framework in edge computing, named "EdgeMediChain." The work promoted the necessary requirements for scalability, security, and privacy of medical ecosystem. Similarly, Nguyen et al. [14] proposed a new decentralized health architecture that integrates mobile edge computing and blockchain for data offloading and data sharing in distributed hospital networks. They utilized a decentralized authentication mechanism associated with a distributed IPFS storage to improve service quality. The ongoing study [15] proposed a ubiquitous healthcare framework that leveraged edge computing, big data, deep learning, high-performance computing, and the IoMT to solve the above challenges. The framework made use of four layers. Three main components improved network service quality. However, these works did not provide profile matching and incentive mechanisms for these health data.

2.2. Profile Matching. Some cloud-assisted solutions have been proposed to address the problems of data security and privacy preservation for profile matching [16–18]. Li et al. [17] put forward a privacy-preserving and scalable matching protocol without disclosing the users' personal data to the cloud. To achieve secure communication between users, He et al. [18] designed an efficient Cross-Domain HandShake (CDHS) scheme with symptom matching in hierarchical identity-based cryptosystem. Compared with the existing profile matching schemes, their schemes used cloud computing to conduct most of the computation, which effectively reduced user's computation complexity. However, the cloud is a semi-trusted center, which may collude with other malicious users to obtain the sensitive data.

Blockchain has drawn considerable interests in research and industrial communities [19–22] due to its advantages. Yang et al. [23] designed a distributed matching mechanism based on blockchain. Furthermore, their scheme transformed the social welfare maximization problem into a matching game of bilateral matching and unilateral preference. Meng et al. [24] proposed a blockchain-enabled signature matching while achieving data security and privacy preservation. In [25], a matching scheme based on hierarchical blockchain was proposed to protect users' privacy. The scheme combined ciphertext-policy attribute-based (CP-ABE) encryption and bloom filter to meet users' requirements to search friends.

The aforementioned works proposed heuristic blockchain-based profile matching schemes with data security and privacy preservation. In fact, some researches presented a structure or concept for profile matching based on blockchain without technically proposing a solution in detail for a specific application scenario. In this work, we present a novel blockchain-based profile matching framework by employing (key-policy attribute-based encryption) KP-ABE and bloom filter to achieve data security and privacy preservation. In addition, we design a detailed profile matching protocol and implement the devised smart contracts on Ethereum test platform.

2.3. Incentive Mechanisms. To encourage more users to join the system, some incentive mechanisms have been designed. Jiao et al. [26] proposed a profit maximization mechanism to maximize the profit under different users' valuation distributions. The pricing mechanism effectively solved the profit maximization problem and provided useful strategies for users. Game-theoretic methods had been extensively used in pricing mechanism [27, 28]. The uncertainty of future prices with a single-leader, multiple-follower Stackelberg game was proposed to maximize profits by setting optimal pricing strategy [27]. It provided a simple distributed algorithm for finding the unique Stackelberg equilibrium point. Chen et al. [28] proposed a pricing approach for incentive mechanisms and considered a two-stage game in a three-layer architecture. They formulated a Markov decision process (MDP)-based social welfare maximization model and studied a convex optimization pricing problem.

Some works adopted blockchain technology to design incentive mechanisms. Liu et al. [29] designed an optimal pricing mechanism and adopted a two-phase Stackelberg game to address pricing and buying problem of the users. Furthermore, they used backward induction to investigate the quantity of purchased data and pricing strategies. Li et al. [30] put forward an incentive mechanism to encourage users to participate in sharing data through price-aware top-k matching queries. Xiong et al. [31] proposed multi-leader multi-follower game-based alternating direction method of multipliers algorithm for pricing to accomplish optimum results in a distributed manner. Zhang et al. [32] designed a dynamic, hierarchical pricing mechanism based on economic modeling methods and heterogeneous agent theory. Nie et al. [33] proposed an optimal incentive mechanism of users using backward induction and applied a Stackelberg game to analyze users' participation level.

The existing work failed to apply the incentive mechanism to encourage data owners and data requesters to share health data. In this scheme, we combine blockchain technology and smart contract to design an optimal pricing mechanism with the guarantee of data security and privacy preservation for motivating users to actively join the system and maximizing their profits.

3. Preliminaries

In this section, we provide preliminaries required in this work.

3.1. Cryptographic Assumptions

Definition 1. Decisional Bilinear Diffie–Hellman Problem (DBDH). We denote an elliptic curve as E and consider a cycle group G of prime order q. Let P be a generator in G_1 and $a, b, c, z \in Z_q^*$. The DBDH problem is defined as follows: given an input tuple $(P \in G_1, aP \in G_2, bP \in G_2, cP \in G_2)$, $e(P, P)^{abc}$ is computed. Assume that an attacker $\mathscr{A}$ can successfully distinguish between $e(P, P)^{abc}$ and $e(P, P)^{z}$ with the advantage $\mathrm{Adv}_{\mathscr{A}}^{DB\ DH}(\lambda)$. If the DBDH assumption holds, the advantage $\mathrm{Adv}_{\mathscr{A}}^{DB\ DH}(\lambda) \leqslant \varepsilon$ must be ignored.

3.2. Linear Secret-Sharing Schemes (LSSS).

$$\mathrm{Adv}_{\mathscr{A}}^{DBDH}(\lambda) = \left| \begin{matrix} \Pr\left[\mathscr{A}\left(e, P, P, P, P, e(P, P)^{abc} = 1\right)\right] \\ -\Pr\left[\mathscr{A}\left(e, P, P, P, P, e(P, P)^{z} = 1\right)\right] \end{matrix} \right| \leqslant \varepsilon. \quad (1)$$

An LSSS access structure is denoted as (N, ρ), where N is an $l \times n$ matrix and ρ is an injective function from $\{1, 2, \ldots, l\}$. to attributes. Let $S \in N$ be an authorized set, defined as $I = \{i \mid \rho(i) \in S\}$. Let T_r be the set of distinct attributes in N. Here, $T_r = \{b\colon \exists i \in [1, l], \rho(i) = b\}$. There exist constants $\{\pi_i \in Z_P, i \in I\}$ such that $\sum_{i \in I} \pi_i N_i = (1, 0, \ldots 0)$, where N_i is *ith* row of N. It randomly generates a vector $(\overrightarrow{v}) = \{\mu_1, r_2, \ldots, r_n\}$, where $\mu_1 \in Z_q^*$ is the secret to be shared and $r_2, \ldots, r_n \in Z_q^*$ is randomly chosen. For $1 \leq i \leq l$, it will compute $\lambda_i = (\overrightarrow{v}) N_i$, where N_i is *ith* row of N. Given an attribute set S and N_i, $\{\pi_i \in Z_q, i \in I\}$ is found and $\sum_{i \in I} \pi_i N_i = (1, 0, \ldots 0)$ is formulated.

3.3. Bloom Filter. A bloom filter [34] is a data structure for validating whether an element comes from a set. An m-bit bloom filter can be represented by a set of hash functions $H = \{H_1', H_2', \ldots H_k'\}, i \in [1, k]$. All the positions in the array are all initially 0. To query an input $w \in W = \{w_1, \ldots w_n\}$ into the bloom filter, its corresponding position is set as 1; i.e., $BF[H_i'(w)] = 1, i \in [1, k]$ is set. The bloom filter checks whether an element w is selected from a set W. w is hashed by the k hash functions $H = \{H_1', H_2', \ldots H_k'\}$ to obtain k hashed array positions. If any of the bits in these positions are set to 0, it represents that the element w does not belong to the set. Otherwise, all the bits in the positions are set to 1, which means the element is in the set. There exists a possibility that $w \notin W$, and all bits in the corresponding positions are all 1, termed as the false-positive error. Its probability is $(1 - (1 - 1/m)^{kn})^k$. Here, $k = m/n\ln 2$. Hash functions are able to minimize false-positive probability $(0.6185)^{m/n}$. There are two algorithms, shown as follows:

$BF \longleftarrow BF\,\text{Build}\,(\{H_1', H_2', \ldots, H_k'\}\{w_1, w_2, \ldots, w_n\})\cdot BF$ Build, and algorithm puts all w into BF.

$c_w \leftarrow \text{Verify}BF\,(\{H_1', H_2', \ldots, H_k'\}, BF, w)$. If $w \notin W$, it returns 0. Otherwise, it returns 1.

3.4. Blockchain and Smart Contract. Blockchain is a distributed ledger, which records numerous transactions [35]. All nodes in the network have the same copy of the ledger. Blockchain has the features of decentralization, privacy preservation, immutability, fault tolerance, and the ability to implement smart contracts. In blockchain, consensus algorithm is used to address the consistency issue of untrusted systems [36]. Current proposed consensus mechanisms include proof of work (PoW), proof of stake (PoS), and delegated proof of stake (DPoS) [37–39]. In our system, we design a proof of existence consensus mechanism for the proposed system.

Smart contract is a computer program that can be self-executed, self-verified, and tamper-resistant without trusted parties. The credibility of smart contract drives from its unforgeability. It cannot be modified or altered once it is deployed on blockchain. Ethereum is a decentralized application platform of smart contract, which supports application of Turing complete. It allows users to create, deploy, and run smart contract on blockchain. In our scheme, we design and deploy the smart contracts on Ethereum to test its availability and evaluate its performance.

4. System Model

In this section, we first propose the system architecture built upon edge-based IoMT. Then, we analyze the security threats and set the security goals.

4.1. System Architecture. The proposed model is made up of different entities: attribute authority (AA), IoMT devices, data owners (DOs), data requesters (DRs), and edge nodes, as shown in Figure 1. The major symbols used in the system architecture are listed in Table 1.

4.1.1. Attribute Authority. AA takes charge of system initialization and registration for DO and DR. It is a trustworthy center in this system. Furthermore, AA generates attribute keys for authorized entities.

4.1.2. IoMT Devices. IoMT devices contain smart devices, such as smartphone, smart watch, and other wearable devices. They are used to collect the data, especially health data in this context. Furthermore, the collected data are uploaded to DOs who can share their data with others within the edge-based IoMT infrastructure.

4.1.3. Data Owners. Data owners refer to patients who construct a community to share their medical experiences and symptoms with others for broadening healthcare information. These DOs can form a social Internet. Additionally, they can earn fees by sharing their symptoms and experiences. They must register at the AA for obtaining attribute keys and joining the system. Moreover, DOs encrypt the keywords of their symptoms using attribute keys. They participate in pricing smart contract to derive an optimal price according to their own pricing strategies.

4.1.4. Data Requesters. Data requesters refer to patients who want to seek similar patients. Similar patients can share medical experiences, so that data requesters can better understand their health states. The DR should firstly register at the AA for attribute keys to take part in the system. The DR can generate search tokens using attribute keys and search relevant symptoms on blockchain. Once DRs receive the matching result from pricing and payment smart contract, they will select the intending data according to the matching results and pay fees to the corresponding DO by pricing and payment smart contract. Furthermore, they earn the fee by providing feedback to feedback smart contract.

4.1.5. Edge Nodes. Edge nodes are used to maintain the blockchain. Edge nodes are also responsible for packaging transactions into blocks in the blockchain.

There are three smart contracts deployed in our proposed blockchain: verification smart contract (VSC), pricing and payment smart contract (PPC), and feedback smart contract (FSC). Firstly, verification smart contract is used to verify whether the profile keywords are valid. Secondly, pricing and payment smart contract is used to set price based on the optimal pricing mechanism and transfer fee to corresponding user's account. Finally, feedback smart contract is used to reward data requester.

4.2. Workflow. Firstly, data requesters and data owners register at the AA for gaining attribute keys to join the system. Patients (data owners) construct a community to share their medical experiences and symptoms with others. They encrypt and upload their profile keywords. KP-ABE and verification smart contract verify whether the keywords belong to a keywords set. If the keywords are valid, they will be uploaded to blockchain for users to search. A data requester, who wants to seek similar patients without real identities, uses the master public key, keywords, and his/her private key to generate a token for searching. After that, if there are matching keywords, they will be sent to pricing and payment smart contract. The results of pricing will be sent to the DR. Furthermore, the DR is able to access DOs' data by paying fees to them. DRs send feedback information to

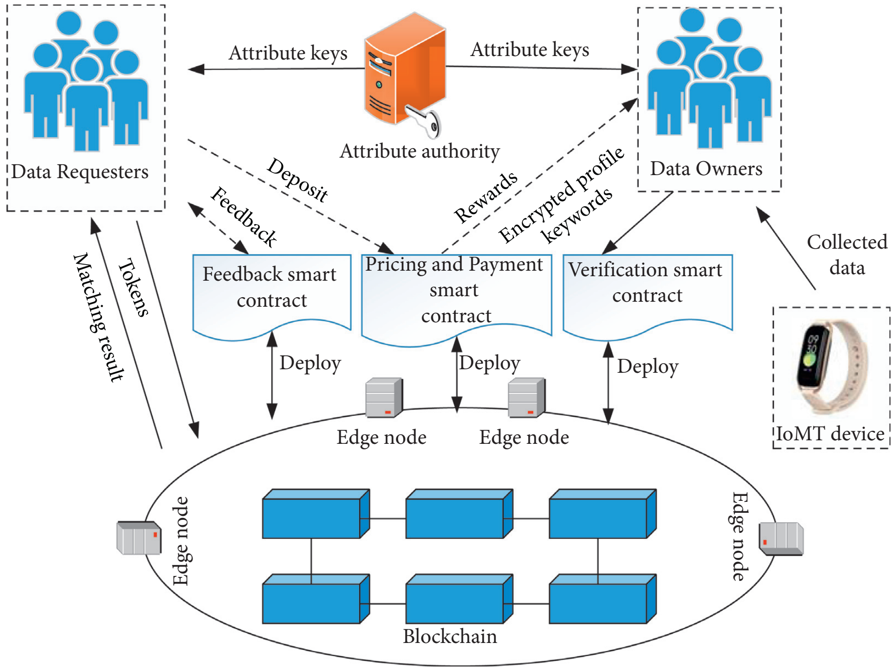


Figure 1: System architecture.

Table 1: Notation table of the architecture.

Notation	Description
KP-ABE	Key-policy attribute-based encryption
AA	Attribute authority
DOs	Data owners
DRs	Data requesters
VSC	Verification smart contract
PPC	Pricing and payment smart contract
FSC	Feedback smart contract

feedback smart contract. Moreover, a small fee is rewarded to the DR by feedback smart contract.

4.3. Security Threats and Design Goals. In our scheme, the AA is completely trusted for performing registration and generating attribute keys, but unauthorized entities may access the encrypted profile keywords and tokens to gain DOs' private information such as identity and address during its transmission in the system. Furthermore, there may be some dishonest requesters who access DOs' data without paying fees. Some DOs may provide false/fake information to requesters. Thus, we aim to achieve the following security goals.

4.3.1. Privacy Preservation. DOs' identities contain some significant privacy information, which cannot be leaked or learned by unauthorized opponents. Besides, feedback information from requesters cannot reveal the DOs' identity information.

4.3.2. Secure Match. In our scheme, the eavesdroppers attempt to guess attribute keys or search tokens for accessing data. Therefore, it is of great significance to protect keys and tokens from revealing during matching process.

4.3.3. Fairness and Incentive. DRs must pay a reasonable fee to a DO for accessing the DOs' data. On the other hand, DOs should provide true information for DRs. Smart contract could realize payment with fairness and verify the authenticity of the matched profile keywords. To incentivize more DRs and DOs to participate in sharing health data, we design an optimal pricing mechanism to maximize their profits. When DRs and DOs accomplish data sharing, they will obtain corresponding rewards.

4.3.4. Access Control. DOs have the ability to control data access and establish access policy. Access control ensures that requesters satisfying access policy can access the DOs' information. Malicious attackers cannot eavesdrop or modify patients' profile keywords, which are transmitted in the public channel of edge-based IoMT. The attribute-based encryption algorithm is adopted to achieve data confidentiality and integrity in this system.

5. The Proposed Protocol

In this section, we firstly describe the proposed protocol based on blockchain. Then, we demonstrate the optimal pricing mechanism.

5.1. Protocol Description. The proposed protocol contains three phases: system setup, index generation, and profile matching. The process of the proposed protocol is shown in Figure 2.

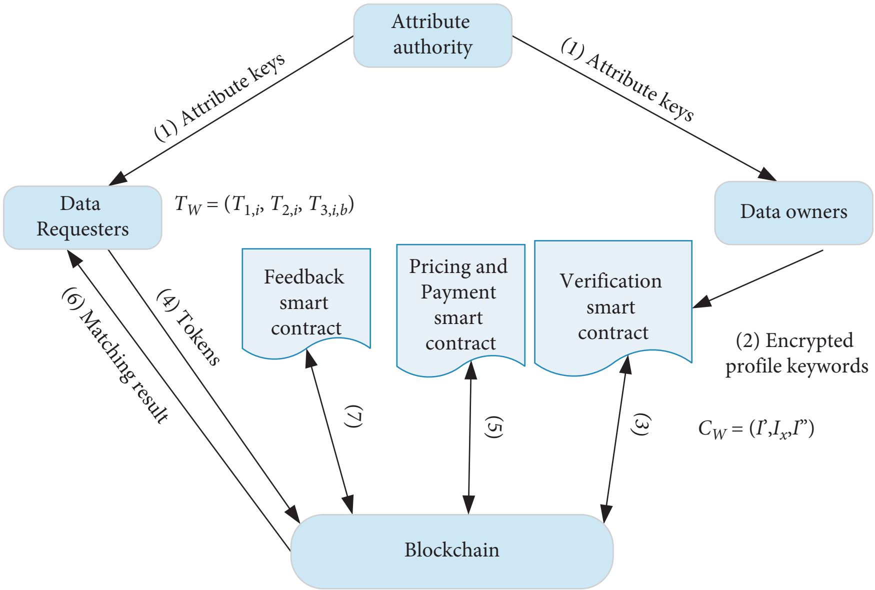


Figure 2: Proposed protocol.

5.1.1. *Phase1: System Setup.*

Given a security parameter λ, the AA selects a bilinear map [40] e: $G_1 \times G_2 \longrightarrow G_T$, where G_1 and G_2 are two additive cyclic groups of the same prime order q. G_T is a multiplicative cyclic group of prime order q. P_1 is the generator of G_1, and P_2 is the generator of G_2. AA chooses a hash function H_1, where H_1: $\{0,1\} \longrightarrow Z_q^*$. In addition, the AA randomly chooses μ_1, μ_2, and $\theta_i \in Z_q^*$, $g \underset{\wedge}{=} \widehat{e}(P_1, P_2)^{\mu_2}$. In the system, it defines the attribute space as U . Let m be the number of bits in a bloom filter (BF) and k be the amount of hash functions in a BF. The AA selects $U = |\widehat{U}|$ random group elements $h_1, h_2, \ldots, h_U \in G_1$. It generates k hash functions $H_1', H_2', \ldots, H_k'$, which are used to add an element into corresponding positions in a BF. The master public key $MPK = (H_1, \widehat{e}, g, P_1,\ P_2, \mu_1 P_1, \mu_2 P_2, \theta_i P_1, \theta_i P_2, h_1, h_2,\ \ldots, h_U, H_1', H_2', \ldots, H_k')$, and the master secret key $MSK = (\mu_1, \mu_2, \theta_i)$.

The AA generates a secret key when a requester registers at the AA. There is an LSSS access structure (N, ρ) for the requester, where N is an $l \times n$ matrix and T_r is a set of diverse attributes in N. It means $T_r = \{b\colon \exists i \in [1, l], \rho(i) = b\}$, and $\forall b \in \mathrm{T}_r/\rho(i)$. The function ρ makes a row of matrix N map to attributes. It chooses a random vector $(\overrightarrow{v}) \in Z_q^*$. This vector's values will be used to share μ_1. For $1 \le i \le l$, it computes $\lambda_i = (\overrightarrow{v})N_i$, where N_i is *ith* row of N. Furthermore, it chooses $\sigma_1, \ldots, \sigma_l \in Z_q^*$ and computes $A_i = \lambda P_2 + \theta_{\rho(i)}\sigma_i P_2$, $\mathrm{B}_i = \sigma_i P_2$, $\forall b \in \mathrm{T}_r/\rho(i)$, and $E_{i,b} = \theta_b \sigma_i P_2$. Let the private key be $ak = (\mathrm{A}_i, \mathrm{B}_i, E_{i,b})$.

5.1.2. *Phase 2: Index Generation.*

Firstly, consensus users from edge nodes participating in profile matching are selected to participate in the network consensus, shown in Figure 3. Assume that the number of all consensus users is N_c in the blockchain network. The system generates a random number as the index of the consensus users to be the selected as the miner.

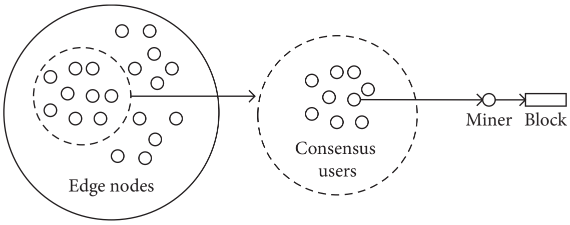


Figure 3: Consensus process.

Secondly, an edge node selects a random value $\chi \in Z_q^*$. He/she computes the keyword ciphertext $c_w = (I_1, I_z, I_2)$, where $I_1 = \chi P_1, I_z = \theta_z \chi P_1, z \in Att$, and $I_2 = g^{H_1(w)\chi}$. Then, the miner sends c_w to blockchain and generates transaction data stored in the transaction pool, packs them into a block, and sends the block to all the consensus users who can verify that the keyword of secure index in the new generating block is selected from W according to Algorithm 1. If $w \in W$, it returns 1, otherwise it returns 0. The bloom filter is used to verify the authenticity of keywords, shown in Algorithm 2.

5.1.3. *Phase 3: Profile Matching.*

Keyword search. The DR searches desired data using the DR's private key *ak* to generate a keyword trapdoor T_w in blockchain. The trapdoor $T_w = (T_{1,i}, T_{2,i}, T_{3,i,b})$ is generated as follows.

> Choose a random vector $(\overrightarrow{V}) = \{\mu, \gamma_1, \ldots, \gamma_l\}$, where random number $\gamma_1, \ldots, \gamma_l \in Z_q^*$. This vector's values will be used to share μ.
>
> For $1 \le i \le l$, it computes $\eta_i = (\overrightarrow{V}) \cdot N_i$. It chooses $\varepsilon_1, \ldots, \varepsilon_l \in Z_q^*$ and computes $T_{1,i} = \eta_i H_1(W) P_2 + \theta_i \sigma_i \varepsilon_i P_2$, $T_{2,i} = \varepsilon_i B_i$, $\forall b \in T/\rho(i)$, and $T_{3,i,b} = \varepsilon_i E_{i,b}$, where T is a set of distinct attributes in N for $1 \le i \le l$. Then, DR sends $T_w = (T_{1,i}, T_{2,i}, T_{3,i,b})$ to blockchain.

```
Input: A keywords set $W$, an array of length $m$, $n$ keywords, $k$ hash functions $H = \{H_1', H_2', \ldots, H_k'\}$.
Output: Bloom Filter $BF$.
$BF$ = a new $m$- bits array of $n$ elements
for $i = 1$ to $m$ do
    $BF[i]$ = NULL
for each element $w \in W$ do
    for $i = 1$ to $k$ do
        $j = H_i'(w)$.
        if $BF[j]$ == NULL
            $BF[j] = 1$.
return $BF$.
```

Algorithm 1: BuildBF (W, m, n, k, h).

```
Input: A keyword $w$, symmetric key $s_k$, $k$ hash functions, $Enc_{s_k}(H) = Enc_{s_k}(\{H_1'(w), H_2'(w), \ldots, H_k'(w)\})$.
Output: if $w \in W$, return 1, else return 0
for each element $w$.
    for $i = 1$ to $k$ do
        $c_j = Enc_{s_k}(H_j'(w))$.
        $j = De\ c_{s_k}(c_j)$.
    if each $BF[j] == 1$.
        return 1
    else return 0
```

Algorithm 2: VerifyBF (ADO, Hcw).

Test. Assume that attributes Att associated with c_w satisfy (N, ρ). There exists $i \in I, \{\pi_i \in Z_q^*\}$ such that $\sum_{i \in I} \pi_i N_i = (1, 0, \ldots 0)$, where N_i is *ith* row of N. Given an original ciphertext c_w, a keyword w, and a search token T_w, the following equation is verified:

$$e\left(I_1, \sum_{i \in I} \pi_i \left(T_{1,i} + \sum_{z \in \Delta/\rho(i)} T_{3,i,z}\right)\right) \overset{?}{=} e\left(\sum_{z \in \Delta} I_z, \sum_{i \in I} \pi_i T_{2,i}\right) \cdot I_2. \quad (2)$$

If (2) holds, the matching keyword $w\prime$ is sent to pricing and payment smart contract. The DO sets payment according to the optimal pricing mechanism. The incentive mechanism is expected to encourage more data owners and data requesters to participate in the process of profile matching. Otherwise, it aborts.

Correctness: when the encrypted keyword is the same as the keyword in the trapdoor, the correctness of the test algorithm is verified as follows:

$$\begin{aligned}
& e\left(I_1, \sum_{i \in I} \left(T_{1,i} \sum_{z \in \Delta/\rho(i)} T_{3,i,z}\right) \pi_i\right) = e\left(\chi P_1, \sum_{i \in I} \pi_i \left(\eta_i H_1(w) P_2 + \theta_i \sigma \varepsilon P_2 + \sum_{z \in \Delta/\rho(i)} \varepsilon_i E_{i,z}\right)\right) \\
& = e\left(\chi P_1, \sum_{i \in I} \pi_i \left((\overrightarrow{V}) \cdot M_i H_1(w) P_2\right)\right) e\left(\chi P_1, \sum_{i \in I} \pi_i \left(\theta_i \sigma_i \varepsilon_i P_2 + \sum_{z \in \Delta/\rho(i)} \varepsilon_i \theta_z \sigma_i P_2\right)\right) \\
& = e(\chi P_1, \mu_1 H_1(w) P_2) e\left(\chi P_1, \sum_{i \in I} \pi_i \left(\sum_{z \in \Delta} \varepsilon_i \theta_z \sigma_i P_2\right)\right) \qquad (3) \\
& e\left(\sum_{z \in \Delta} I_z, \sum_{i \in I} \pi_i T_{2,i}\right) \cdot I_2 = e\left(\sum_{z \in \Delta} \theta_z \chi P_1, \sum_{i \in I} \pi_i \sigma_i \varepsilon_i P_2\right) \cdot e(P_1, P_2)^{\mu_2 \chi H_1(w)} \\
& = e\left(\sum_{z \in \Delta} I_z, \sum_{i \in I} \pi_i T_{2,i}\right) \cdot I_2.
\end{aligned}$$

Based on PBFT, if not less than 2/3 consensus users verify the new block, following the basic setting of 2/3, etc. [41], the new block will be added into the blockchain as a new valid block. In this round, the generation of a valid block marks the completion of consensus. The miner will generate a random number $R\prime\in [0, N_c - 1]$ to determine the next miner in the next round. Then, the DO participates in pricing and sets the fees. When the DR wants to access the desired data, he/she needs to pay the fee to the corresponding DO by PPC. The specific pricing process can be seen in the following optimal pricing mechanism.

5.2. Optimal Pricing Mechanism. When a data requester matches the profile keyword in blockchain, he/she accesses the corresponding data by paying fees to the data owner. In this section, we design an optimal pricing mechanism based on a Stackelberg game. Optimal price maximizes the profits of the data owners and the data requesters. The data owners determine their own prices of data based on profit functions as the leaders. The data requesters determine the access quantity of data as followers. The Stackelberg game is between data owners and data requesters, shown in Figure 4.

5.2.1. Stackelberg Game Problem Formulation. The number of data owners is M. A set of data owners can be expressed by $\widetilde{M} = \{1, \ldots, M\}$. Data owners provide the desired data for the data requester. In our optimal pricing mechanism, there is a DR determining access strategy, such as access amount of data. Meanwhile, each data owner $j \in M$ determines the optimal price for corresponding data. The access amount of the data requester from data owner j is x_j. The unit price of data owner j for the data requester is p_j. We denote the access amount set and the optimal access amount set of the data requester from different data owners as $X_j = \{x_1, \ldots, x_M\}$ and $X_j^* = \{x_1^*, \ldots, x_M^*\}$, respectively. The unit price set and the optimal price set of the data requester from different data owners are denoted as $P_j = \{p_1, \ldots, p_M\}$ and $P_j^* \{\{p_1^*, \ldots, p_M^*\}$, respectively. The main symbols used in an optimal pricing mechanism are shown in Table 2.

To evaluate the access quality of data, an access quality factor is denoted by q. Data owner j provides data's quality, formulated as follows:

$$Q(x_j) = q\ln(1 + x_j). \tag{4}$$

The data requester's utility is expressed as follows:

$$\begin{aligned} U_r(x_j, p_j) &= f_j a_j Q(x_j) - x_j p_j \\ &= q f_j a_j \ln(1 + x_j) - x_j p_j. \end{aligned} \tag{5}$$

The data requester gives a feedback evaluation to the data of the data owner j and obtains small fees as rewards, denoted as f_j. a_j is the accessing willingness of the data requester. The data requester maximizes its utility based on the access quantity x_j, forming its subgame, which is given as follows.

Problem 1 (data requester's subgame for data owner j):

$$\begin{aligned} &\max_{x_j} U_r(x_j, p_j), \\ &s.t. \quad x_j \in [x_{\min}, x_{\max}], \quad \forall j \in M. \end{aligned} \tag{6}$$

$x_{\min}$ is the minimum amount of accessed data, and $x_{\max}$ is the largest amount of accessed data.

Data owner's utility can be expressed as follows:

$$U_j(x_j, p_j) = s_j(x_j p_j - x_j c_j). \tag{7}$$

Here, c_j represents the unit cost set by data owner j. The data owner's reputation score is expressed as s_j, which changes dynamically according to the data requester feedback on the data owner's data and amount of files downloaded by the data requester. $s_j \in [s_{\min}, s_{\max}]$. The terms $s_{\min}$ and $s_{\max}$ are lower and upper bound reputation scores, respectively. $s_{\min} < s_{\text{bad}} < s_{\text{init}} < s_{\text{good}} < s_{\max}$. s_{bad} is the bad reputation score threshold. s_{init} is the initial reputation score threshold. s_{good} is the good reputation score threshold. The data owner j maximizes its utility based on the price p_j, forming its subgame, which is given as follows.

Problem 2 (data owner's subgame for the data requester):

$$\begin{aligned} &\max_{\mathrm{p}_j} U_j(x_j, p_j), \\ &s.t. \quad p_j \in [c_j, p_{\max}], \quad \forall j \in M. \end{aligned} \tag{8}$$

The data requester can accept the maximum price, denoted as $p_{\max}$. Problem 1 and problem 2 constitute the Stackelberg game, which aims for finding the equilibrium points. In other words, the profit of data owner comes up to maximization when the data requester obtains its largest profit.

Definition 2. The points (x_j^*, p_j^*) are an equilibrium if it satisfies both the following two conditions:

$$U_r(x_j^*, p_j^*) > U_r(x_j^*, p_j), \quad \forall j \in M, \tag{9}$$

$$U_j(x_j^*, p_j^*) > U_j(x_j, p_j^*), \quad \forall j \in M. \tag{10}$$

To analyze the Stackelberg game, we use backward induction [28].

5.2.2. Data Requesters' Accessing Strategy in Stage II. Data owner j provides data's unit price for the data requester (i.e., p_j, for all $j \in M$). The data requester from data owner decides its optimal access strategy (i.e., x_j, for all $j \in M$).

First, we take the derivative on the data requester's utility in (5) according to x_j, which is expressed as follows:

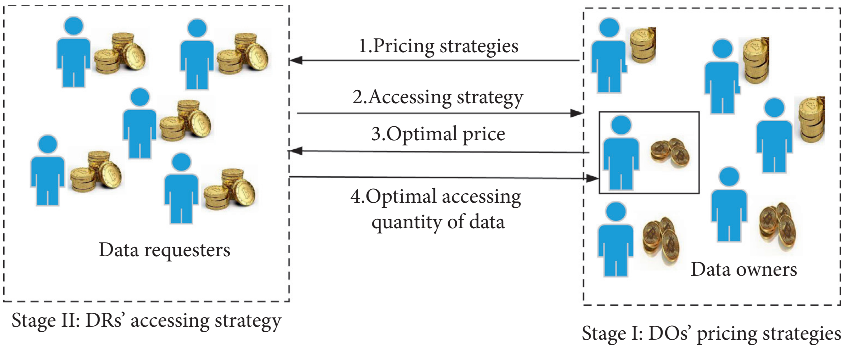

1.Pricing strategies
2.Accessing strategy
3.Optimal price
4.Optimal accessing quantity of data
Data requesters
Data owners
Stage II: DRs' accessing strategy
Stage I: DOs' pricing strategies

FIGURE 4: Stackelberg game-based optimal pricing mechanism.

TABLE 2: Notation table of the pricing mechanism.

Notation	Description
$\widehat{M}$	The set of data owners
M	The amount of data owners
x_j	The amount of accessed data
$x_{\max}$	The largest amount of accessed data
$x_{\min}$	The minimum amount of accessed data
x_j^*	The optimal amount of accessed data
X_j^*	The optimal amount set of accessed data
a_j	The accessing willingness of the data requester
s_j	The data owner's reputation score
f_j	The small fees as rewards
c_j	The cost of assessed data of data owner j.
p_j	The unit price set by data owner j.
$p_{\max}$	The maximum accepted price set by the data requester
p_j^*	The optimal price set by data owner j.
P_j^*	The set of optimal price

$$\frac{\partial U_r}{\partial x_j t} = \frac{\partial\left[f_j a_j q \ln\left(1+x_j\right) - x_j p_j\right]}{\partial x_j} = \frac{f_j a_j q}{1+x_j} - p_j, \tag{11}$$

$$\frac{\partial^2 U_r}{\partial x_j^2} = \frac{\partial\left(f_j a_j q/1 + x_j - p_j\right)}{\partial x_j} = \frac{-f_j a_j q}{1+x_j} < 0. \tag{12}$$

$U_r(x_j, p_j)$ is a strictly concave function shown by the above derivatives. To obtain the optimal response function, we set $[(\partial U_r)/(\partial x_j)] = 0$, which is given as follows:

$$\frac{\partial U_r}{\partial x_j} = \frac{f_j a_j q}{1+x_j} - p_j = 0. \tag{13}$$

Then,

$$x_j^* = \frac{f_j a_j q}{p_j} - 1. \tag{14}$$

According to (14), x_j^* is data requester's optimal quantity of accessed data.

5.2.3. Data Owners' Pricing Strategies in Stage I. Data owner can obtain his/her largest profit based on the data requester's optimal access strategy. According to formulation (7)–(14), the optimal utility of data owner for the DR can be rewritten as follows:

$$\begin{aligned} U_j\left(x_j^*, p_j\right) &= s_j\left(x_j p_j - x_j c_j\right) \\ &= \left(s_j f_j a_j q + s_j c_j\right) - \left(s_j p_j + \frac{s_j f_j a_j q c_j}{p_j}\right). \end{aligned} \tag{15}$$

First, we take the derivative on data owner's utility in (15) according to p_j, which is expressed as follows:

$$\frac{\partial U_j}{\partial p_j} = \frac{\partial\left[\left(s_j f_j a_j q + s_j c_j\right) - \left(s_j p_j + s_j f_j a_j q c_j / p_j\right)\right]}{\partial p_j} = \frac{s_j f_j a_j q c_j}{p_j^2} - s_j, \tag{16}$$

$$\frac{\partial^2 U_j}{\partial p_j^2} = \frac{\partial\left(s_j f_j a_j q c_j / p_j^2 - s_j\right)}{\partial p_j} = \frac{-2 s_j f_j a_j q c_j}{p_j^3} < 0. \tag{17}$$

$U_j(x_j^*, p_j)$ is a strictly concave function shown by the above derivatives. To obtain the optimal response function, we set $[(\partial U_j)/(\partial p_j)] = 0$, which is given as follows:

$$\frac{\partial U_j}{\partial p_j} = \frac{s_j f_j a_j q c_j}{p_j^2} - s_j = 0. \tag{18}$$

Then,

$$p_j^* = \sqrt{f_j a_j q c_j}. \tag{19}$$

According to (19), when f_j, a_j, q, and c_j are the active impact, it will get data owner's optimal price.

According to (7), we gain the data requester's optimal utility from data owner, given as follows:

$$\begin{aligned} U_j\left(x_j^*, p_j^*\right) &= s_j x_j^* p_j^* - s_j x_j^* c_j \\ &= \left(\sqrt{f_j a_j q} - \sqrt{c_j}\right)^2 s_j. \end{aligned} \tag{20}$$

According to (5)–(19), we are able to get the optimal utility of the data requester from data owner j, given as follows:

$$\begin{aligned} U_r\left(x_j^*, p_j^*\right) &= f_j q a_j \ln\left(1 + x_j^*\right) - x_j^* p_j^* \\ &= f_j q a_j \ln\sqrt{\frac{f_j a_j q}{c_j}} - f_j a_j q + \sqrt{f_j a_j q c_j}. \end{aligned} \tag{21}$$

By finding the equilibrium point of the game, both the data requester and the data owners can obtain their own optimal utility. A Nash equilibrium reaches between them. Meanwhile, the incentive mechanism promotes the data owners to take an active part in sharing their experiences. There will be some contradictions and conflicts of benefits if there is no balance between data requesters and data owners.

5.3. Smart Contract Design. To satisfy the system's requirements, we design smart contracts with various functions. The interactions among contracts (programmed in Solidity\footnotehttps: // remix ethereum org/) for Ethereum include the following steps. The function is given in Algorithm 3.

KeywordCiphertext (CW): This function is called to store keyword ciphertext in blockchain. Before storing the data in blockchain, the DO sends the keyword with k hash functions to invoke VSC for verifying that the keyword is selected from a valid keyword set. If the keyword is correct, the keyword ciphertext will be packed in a block.

Pricing (Pricinglist, pricing): This function is used to formulate pricing list for blockchain. Before generating the pricing list based on an optimal pricing mechanism provided by the DO, PPC will invoke VSC for verifying the validity of the keyword. If it is valid, the DO will take part in pricing built on an optimal pricing mechanism. Then, the pricing list will be stored in blockchain.

setFee (Paymentstruct, payment): The function is called to set charges for desired data and feedback data. The fee is divided into two parts: the DO's data fee and the DR's reward fee.

Payment (account, fee): The DR calls this function to transfer the fee to the intended DO's account. Then, he/she can access the desired data. If deposit of the DR's account is enough, it will be executed. Furthermore, PPC sends the payment result to FSC.

6. Security Proof

In this section, we analyze how the proposed scheme is able to effectively realize the goals defined in the section of system model.

6.1. Privacy Preservation. Users send data to blockchain through their blockchain account during data transmissions in edge-based IoMT. Due to the immunity characteristics of blockchain, data in the blockchain are tamper-proof. Therefore, users' sensitive information can be protected. Furthermore, the encrypted keywords are stored in blockchain. They will not divulge any information of users. Our scheme utilizes a bloom filter in smart contract to verify the validity of keywords, which reduces the unnecessary consumption. The optimal pricing mechanism is used to incentivize the DO and the DR to share their data. Malicious behaviors are prevented from getting illegal fees via the anonymous blockchain account.

6.2. Secure Match

Definition 3. Secure match. An adversary $\mathscr{A}$ cannot distinguish the keyword from keyword ciphertext or search trapdoor.

Theorem 1. *The proposed protocol can achieve secure match in the random oracle model on the DBDH assumption.*

To avoid reinvent the wheel, we refer to [43] for the keyword secrecy game. Security proof is as follows.

Proof. The random oracles of algorithms *Private key*, *Trapdoor*, and *Test* are O_{ak}, O_{trapdoor}, and O_{test}, respectively. Assume that $\mathscr{A}$ is an attacker who has advantage ε to attack the proposed chosen keyword attack game. We build a challenger $\mathscr{C}$. He/she plays game with $\mathscr{A}$ to derive the solution to the DBDH problem as follows:

List ak^{list}: record $((N, \rho), ak_{(N,\rho)})$. □

6.2.1. Setup. Given a security parameter λ and an input tuple (r_1P_1, r_2P_1, r_3P_1), challenger $\mathscr{C}$ generates the system master public key $MPK = (H_1, \hat{e}, g, P_1, P_2, \mu_1P_1, \mu_1P_2, \theta_iP_1, \theta_iP_2, H_1)$ and the master secret key $MSK = (\mu_1, \mu_2, \theta_i)$.

(a) $H_1(w)$: if the query exists on H_1^{list1} in a tuple (w, κ_1), κ_1 is returned; else, $\kappa_1 \in_R Z_q^*$ is chosen, (w, κ_1) is added to H_1^{list1}, and $H_1(w) = \kappa_1$ is returned

(b) $H_1(I_1, I_2, \{I_z\}_{z \in Att})$: if the query exists on H_1^{list2} in a tuple $(I_1, I_2, \{I_z\}_{z \in Att}, \kappa_2)$, κ_2 is returned; else,

```
Input: the keyword array Keyword, the pricing's array pricing
function KEYWORDCIPHERTEXT ($CW$) public returns()
    keyword.push ($CW$).
    return result
end function
function PRICING(Pricinglist, pricing) public
    query the corresponding $c_w$.
    if the query is ok then
        Pricinglist.push(price).
    else
        return false
    end if
    return result.
end function
function SETFEE(Paymentstruct, payemnt) only DO public
    query the corresponding pricing result
    if the query is ok then
        paymentstruct.push(payment)
    else
        return false.
    end if
    return result.
end function
function PAYMENT (DR an dD RAccount, payment) returns()
    compute the DO fee DO Fee and the DR fee DR Fee
    query the balance of DR's account
    if DR Balance > DO Fee + DR Fee then
        transfer fee to DO's account and DR's account
        return result.
    else
        return false.
    end if
end function
```

Algorithm 3: Smart Contract Algorithm.

$\kappa_2 \in_R Z_q^*$ is chosen, $(I_1, I_2, \{I_z\}_{z\in Att}, \kappa_2)$ is added to H_1^{list2}, and $H_1(I_1, I_2, \{I_z\}_{z\in Att}) = \kappa_2$ is returned

6.2.2. Phase 1.

(a) $O_{ak}(N, \rho)$: Since challenger $\mathscr{C}$ has knowledge of μ_1 and $\theta_{\rho_{(i)}}$, it can construct private key corresponding to any (N, ρ), and next, the tuple is added to ak^{list}. When (N, ρ) satisfies Att^*, $\perp$ is output.

(b) $O_{\text{trapdoor}}((N, \rho), w)$: If (N, ρ) matches Att^*, challenger $\mathscr{C}$ outputs $\perp$. Else, $\mathscr{C}$ can set $T_{2,i}, T_{3,i,b}$ as in O_{ak}. $\mathscr{C}$ chooses a random vector $(\overrightarrow{V})$ with the first element μ and next sets $\eta_i = (\overrightarrow{V}) \cdot N_i$. It chooses $\varepsilon_1, \ldots, \varepsilon_l \in Z_q^*$ and constructs the trapdoor T_w as in the real scheme.

(c) $O_{\text{test}}((N, \rho), w)$: If the attribute set Att associated with c_w is Att^*, $\mathscr{C}$ outputs $\perp$. Else, $\mathscr{C}$ can always compute a trapdoor T_w as in O_{trapdoor}. Then, it proceeds to the test easily. If the test holds, $\mathscr{C}$ outputs 1 and 0 otherwise.

6.2.3. Challenge. $\mathscr{A}$ outputs w_1^* and w_0^*. Challenger $\mathscr{C}$ sets the original challenge ciphertext as follows:

(a) $I_1^* = r_1 P_1$, $I_z^* = r_1 r_2 P_1$, and $I_2^* = (g^{H_{1(w)}})^{r_3}$ are set.

(b) A random coin $b \in (0, 1)$ is flipped, and an H_1 query on (w_b^*) is issued to achieve κ_1^*.

(c) An H_1 query on $H_1(I_1^*, I_2^*, \{I_z^*\}_{z\in Att^*})$ is issued to achieve κ_2^*.

(d) The challenge original ciphertext is output as $c_w^* = (Att^*, \{I_1^*, I_2^*, \{I_z^*\}_{z\in Att^*}\})$.

(e) If $Q = r_1 r_2 r_3 P_1 \in G_1$, then c_w^* generated a valid ciphertext in which $r_1' = r_1$ and $r_2' = r_2 r_3$. After that, $\mathscr{C}$ sends (c_w^*, T_w^*) to $\mathscr{A}$.

6.2.4. Phase 2. The phase is the same as the *Phase 1.*

6.2.5. Guess. $\mathscr{A}$ outputs a guess bit $b\prime$, if $b = b\prime$, and $\mathscr{C}$ outputs 1; else, it outputs 0.

In the guess phase, if $Q \neq r_1 r_2 r_3 P_1$, then we have the following:

$$Pr\left[\mathscr{A}(P_1, r_1 P_1, r_2 P_1, r_3 P_1, Q) = 1\right] = \frac{1}{2}. \qquad (22)$$

In addition,

$$\begin{aligned}
&\Pr\left[\mathscr{A}\left(\left(P_1, r_1P_1, r_2P_1, r_3P_1, Q\right) = 1\right)\right] \\
&= \Pr\left[\mathscr{A}\left(P_1, r_1P_1, r_2P_1, r_3P_1, Q\right) = 1 | Awins\right]\Pr\left[Awins\right] \\
&\quad + \Pr\left[\mathscr{A}\left(P_1, r_1P_1, r_2P_1, r_3P_1, Q\right) = 1 | \overline{Awins}\right]\Pr\left[\overline{Awins}\right] \\
&= 1 \cdot \varepsilon''(\lambda) + \frac{1}{2}\left(1 - \varepsilon''(\lambda)\right) = \frac{\varepsilon''(\lambda)}{2} + \frac{1}{2}.
\end{aligned} \tag{23}$$

Consequently,

$$\mathrm{Adv}_{\mathscr{A}}^{DBDH}(\lambda) = \frac{\varepsilon''(\lambda)}{2} = \varepsilon'. \tag{24}$$

Thus, the probability of $\mathscr{A}$ wining the keyword secrecy game is ε' at least.

6.3. Fairness and Incentive. The DO's attribute key is used to encrypt the profile keywords. The bloom filter and smart contract can verify the authenticity of keywords. By this way, only valid keywords can be uploaded to blockchain. To search the desired keyword, DR must generate a search trapdoor according to his/her selected access structure. Thus, other entities cannot obtain any information about keywords and matching results during the process of keyword searching. Attackers cannot learn any information from encrypted keywords and trapdoors. Therefore, our scheme can achieve secure match.

6.4. Access Control. Our scheme utilizes key-policy attribute-based encryption (KP-ABE) in the proposed protocol. The keyword w is encrypted by the DO's attribute set Att. The DR uses his/her key ak to generate a search trapdoor T_w, which is related to the access structure. It is used for searching the matching keyword. Only when the attribute Att of keyword ciphertext c_w satisfies the access structure of search trapdoor T_w, the DR can access the desired data after calling pricing and payment smart contract (PPC) to transfer fees to a specific DO's account. If the DR fails to pay fees to the DO's account, he/she cannot query the intended data. Thus, DO is able to control the access of his/her data.

7. Implementation and Performance Evaluation

In this section, we implement the proposed algorithms in a simulated edge-based IoMT environment with Java programming and JPBC library. We deploy the designed smart contracts on Ethereum test platform. Firstly, we introduce the parameter settings. We compare the security properties of our solution with other relevant solutions. Then, the computational overhead and communication overhead are analyzed for the proposed protocol. We design the smart contracts and evaluate the performance of the designed smart contracts on Ethereum test platform. Finally, we evaluate the performance of the proposed optimal pricing mechanism.

7.1. Parameter Settings and Platform. The system security parameter $\lambda = 128$. For some prime $p = 3\mathrm{mod}4$, we utilize type A pairing on the elliptic curve $y^2 = x^3 + x$ over the field F_{p}. The cryptographic primitives are implemented using JPBC library and Java on a laptop computer with Intel (R) Core (TM) i5-7400 CPU @3.00 GHz, 8 GB RAM, and Microsoft Windows 10 operating system.

In addition, we employ Ganache (client version) to build a local test chain on Linux system. We use solidity language to write data into smart contracts and then upload it to blockchain. Smart contract framework and solidity compiler are truffle @0.5.0 and solc @0.5.0, respectively. To gain time consumption of publishing smart contracts, we utilize Web3js library of Nodejs to interact with smart contracts on the blockchain and test the time cost of marking transactions. Due to the limited space, the specific deployment process is skipped.

7.2. Comparisons of Security Properties. We compare the security properties of our scheme with other matching schemes in Table 3. The comparison results indicate that the proposed scheme is capable of providing a promising solution to improve profile matching service in edge-based IoMT scenarios [25, 42, 43]. Their scheme does not achieve security properties of *blockchain-based* and *fairness and incentive*. However, our scheme can provide all of the security properties.

7.3. Communication Overhead and Computational Cost. In edge-based IoMT scenarios, the communication and computation resources are constrained. In this subsection, we show the improved performances of the proposed scheme. We denote $|G_1|$, $|G_2|$, $|G_T|$, and $|Q|$ as the size of elements in G_1, G_2, G_T, and Z_p. The communication overhead is caused by index generation phase and keyword search phase, shown in Table 4. In index generation phase, DO sends c_w to blockchain for searching data, and the total length is $(n+1)|G_1| + |G_T|$ bytes. The DR sends search trapdoor T_w using the secret key to blockchain for searching the desired data, and the total length is $3l|G_2|$ bytes during the process of keyword search. We compare the communication overhead with [42, 43], shown in Table 4. As can be seen from Table 4, the index generation overhead of the proposed scheme is lower. In addition, the overhead of keyword search phase in [42, 43] is higher than our proposed scheme.

We compare the computational overhead in Table 5. The algorithm *SystemInit* simulates system setup phase. The generated keyword ciphertext is simulated by the algorithm *Encrypt*. Furthermore, the algorithm *Trapdoor* generates secret key and search trapdoor for the DR. The algorithm *Test* is used to test whether the keyword ciphertext and trapdoor match. From Table 5, we observe that our computational overhead is higher than that in [43] during the process of the algorithm *Test*. Nonetheless, our scheme's computational overhead is lower than other two schemes.

Table 3: Comparison of security properties.

Properties	[25]	[42]	[43]	The proposed
Blockchain integration	√	×	×	√
Privacy preservation	√	√	√	√
Secure match	√	×	×	√
Fairness and incentive	×	×	×	√
Access control	√	√	√	√

√: the scheme supports this property; ×: the scheme does not support this property.

Table 4: Communication overhead of proposed protocol.

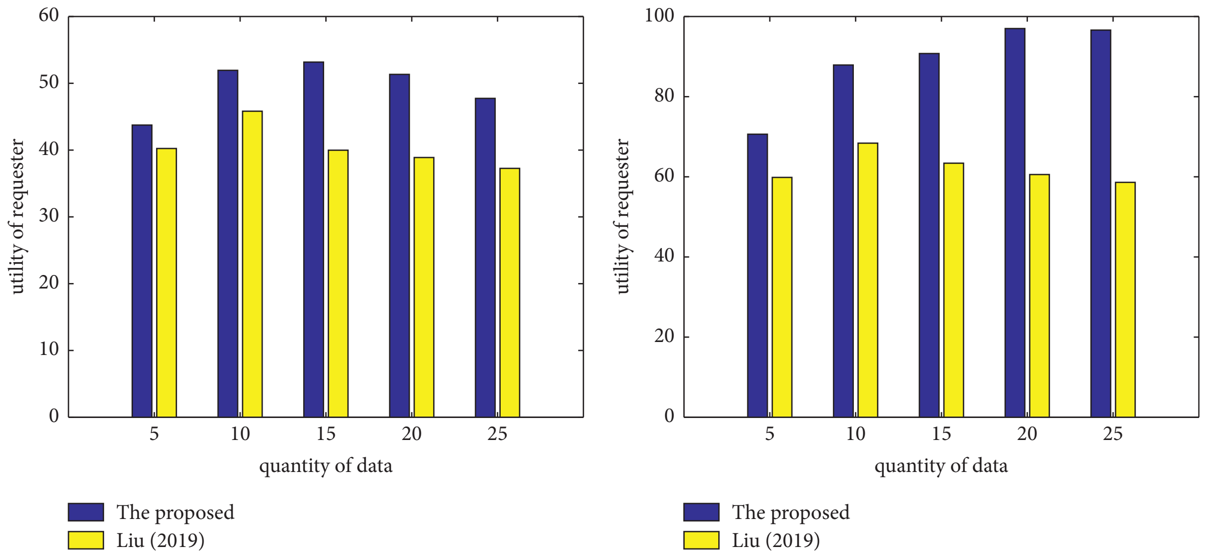
Phases	The proposed	Cui et al. [42]	Miao et al. [43]
Index generation	$(n+1)\lvert G_1\rvert + \lvert G_T\rvert$	$\lvert G_1\rvert + (5n+1)\lvert G\rvert$	$(2n+1)\lvert G\rvert + \lvert G_T\rvert$
Keyword search	$3l\lvert G_2\rvert$	$(6l+2)\lvert G\rvert + \lvert M\rvert$	$(4n+3)\lvert G\rvert + \lvert Q\rvert$

Table 5: Computational overhead of cryptographic algorithms (in *ms*).

	Algorithms	SystemInit	Encrypt	Trapdoor	Test
The proposed	Average time	152	547	523	368
	Max time	665	651	592	376
	Min time	23	527	482	361
Cui et al. [42]	Average time	232	550	696	431
	Max time	746	618	731	457
	Min time	97	513	672	420
Liu et al. [43]	Average time	369	610	626	321
	Max time	766	650	700	367
	Min time	263	582	581	290

Table 6: Time consumption of transactions.

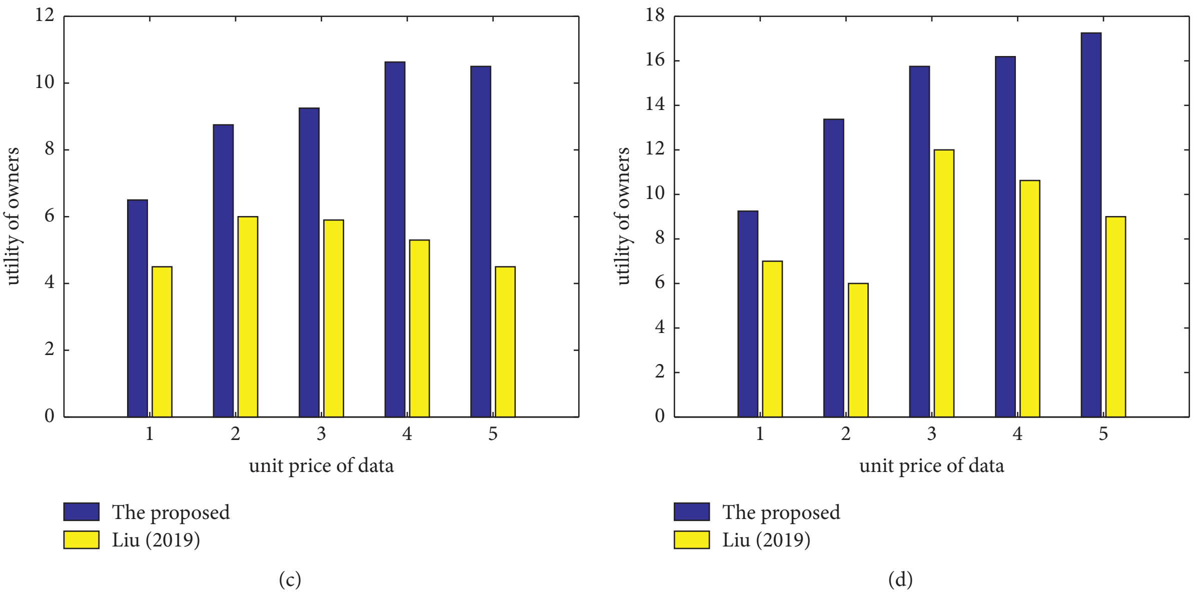
Transactions	Cipheretext	Pricing	Feedback
Average time (ms)	98.8	354.84	79.49
Max time (ms)	255	469	255
Min time (ms)	82	251	46
Gas cost (ether)	0.014663	0.0354413	0.0103504



(a)

(b)

Figure 5: Continued.



(c)

(d)

Figure 5: Computational cost taken by (a) $a_j = 20$, (b) $a_j = 30$, (c) $a_j = 20$, and (d) $a_j = 30$.

7.4. Time Consumption of Smart Contracts and Resource Consumption of Nodes. Because the length of data affects the time consumption of transmitting a transaction to blockchain, we firstly discuss its length. According to Section 7.3, the length of index generation and keyword search is $(n+1)|G_1| + |G_T|$ and $3l|G_2|$. As can be seen from Table 6, we see that the average time consumption of sending a transaction *KeywordCipheretext* to blockchain is 98.8 ms, a transaction *Pricing* is 354.84 ms, and a transaction *Feedback* is 79.49 ms. The gas consumption of transaction *KeywordCipheretext* is 0.0146637 ether, transaction *Pricing* is 0.03544138 ether, and transaction *Feedback* is 0.01035046 ether.

7.5. Performance Analysis of Pricing Mechanism. We evaluate the performance for data owners and data requesters under the proposed optimal pricing mechanism, as shown in Figure 5. We set some parameter values, denoted as follows: $p_{\max} = 5$, $x_{\min} = 10$, $x_{\max} = 100$, $s_j \in [0, 1]$, and $f_j \in [1, 5]$. The parameter a_j represents the data requester's willingness to access the data. When a_j values are different, we simulate the utility of data owners and data requesters, where a_j is equal to B in [44]. In Figures 5(a) and 5(b), we compare optimal pricing mechanism with independent pricing scheme in [44]. We find that the amount of accessed data increases and data requester's utility increases. It shows that in our scheme, the data requester can have more utility than Liu's scheme in [44]. Figure 5(c) and 5(d) shows that the data owners can get more utility with the increase in unit price. Moreover, the results show that the data owners get more profits in our scheme than Liu's scheme in [44] when the accessing willingness of requester increases.

8. Conclusion and Future Work

In this work, we introduce a new blockchain-based profile matching scheme by utilizing KP-ABE algorithm and bloom filter, which guarantees privacy preservation and security of health data in edge-based IoMT. Firstly, we present a system framework based on blockchain for profile matching among different users. Secondly, we design a consensus mechanism for proposed blockchain to achieve the consensus of the system. Thirdly, smart contract with an optimal pricing mechanism is designed to formulate pricing list and encourage more users to participate in the system. We evaluate the performance of communication overhead. We employ JPBC library to evaluate computational cost of the proposed protocol, compared with other schemes. Finally, we deploy the smart contracts on Ethereum platform and test the time consumption of smart contracts.

In our future work, we plan to deploy smart contracts on Hyperledger Fabric and store original data using encryption algorithm in IPFS for profile matching, which has a potential to improve the performances in edge-based IoMT scenarios.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62072005 and the Natural Science Foundation of Anhui Province under Grant 2108085Y22.

References

[1] D. He, N. Kumar, H. Wang, L. Wang, K.-K. R. Choo, and A. Vinel, "A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 633–645, 2018.

[2] W. Tang, J. Ren, and Y. Zhang, "Enabling trusted and privacy-preserving healthcare services in social media health networks," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 579–590, 2019.

[3] Y. Liu, M. Peng, G. Shou, Y. Chen, and S. Chen, "Toward edge intelligence: multiaccess edge computing for 5G and Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6722–6747, 2020.

[4] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor Authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, p. 1, 2020.

[5] A. Asheralieva and D. Niyato, "Distributed dynamic resource management and pricing in the IoT systems with blockchain-as-a-service and UAV-enabled mobile edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1974–1993, 2020.

[6] K. Liu, W. Chen, Z. Zheng, Z. Li, and W. Liang, "A novel debt-credit mechanism for blockchain-based data-trading in Internet of vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9098–9111, 2019.

[7] Y. Qu, L. Gao, T. H. Luan et al., "Decentralized privacy using blockchain-enabled federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5171–5183, 2020.

[8] L. Cui, Y. Qu, G. Xie et al., "Security and privacy-enhanced federated learning for anomaly detection in IoT infrastructures," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3492–3500, 2022.

[9] Q. Wang, D. Wang, C. Cheng, and D. He, "Quantum2FA: efficient quantum-resistant two-factor Authentication scheme for mobile devices," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2021.

[10] S. U. Amin and M. S. Hossain, "Edge intelligence and Internet of Things in healthcare: a survey," *IEEE Access*, vol. 9, pp. 45–59, 2021.

[11] I. V. Pustokhina, D. A. Pustokhin, D. Gupta, A. Khanna, K. Shankar, and G. N. Nguyen, "An effective training scheme for deep neural network in edge computing enabled Internet of medical Things (IoMT) systems," *IEEE Access*, vol. 8, Article ID 107112, 2020.

[12] H. Zhong, Y. Zhou, Q. Zhang, Y. Xu, and J. Cui, "An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare," *Future Generation Computer Systems*, vol. 115, pp. 486–496, 2021.

[13] R. Akkaoui, X. Hei, and W. Cheng, "EdgeMediChain: a hybrid edge blockchain-based framework for health data exchange," *IEEE Access*, vol. 8, Article ID 113467, 2020.

[14] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "BEdgeHealth: a decentralized architecture for edge-based IoMT networks using blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 14, Article ID 11743, 2021.

[15] T. Muhammed, R. Mehmood, A. Albeshri, and I. Katib, "UbeHealth: a personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities," *IEEE Access*, vol. 6, Article ID 32258, 2018.

[16] R. Guo, G. Yang, H. Shi, Y. Zhang, and D. Zheng, "O3-R-CP-ABE: an efficient and revocable attribute-based encryption scheme in the cloud-assisted IoMT system," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8949–8963, 2021.

[17] M. Li, N. Ruan, Q. Qian, H. Zhu, X. Liang, and L. Yu, "SPFM: scalable and privacy-preserving friend matching in mobile cloud," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 583–591, 2017.

[18] D. He, N. Kumar, H. Wang, L. Wang, K.-K. R. Choo, and A. Vinel, "A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 633–645, 2018.

[19] V. Malamas, P. Kotzanikolaou, T. K. Dasaklis, and M. Burmester, "A hierarchical multi blockchain for fine grained access to medical data," *IEEE Access*, vol. 8, Article ID 134393, 2020.

[20] 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 Transactions on Services Computing*, 2021.

[21] P. Zhang and M. Zhou, "Security and trust in blockchains: architecture, key technologies, and open Issues," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 3, pp. 790–801, 2020.

[22] K. Gai, J. Guo, L. Zhu, and S. Yu, "Blockchain meets cloud computing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2009–2030, 2020.

[23] L. Yang, M. Li, H. Zhang, H. Ji, M. Xiao, and X. Li, "Distributed resource management for blockchain in fog-enabled IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2330–2341, 2021.

[24] W. Meng, W. Li, S. Tug, and J. Tan, "Towards blockchain-enabled single character frequency-based exclusive signature matching in IoT-assisted smart cities," *Journal of Parallel and Distributed Computing*, vol. 144, pp. 268–277, 2020.

[25] F. Yang, Y. Wang, C. Fu, C. Hu, and A. Alrawais, "An efficient blockchain-based bidirectional friends matching scheme in social networks," *IEEE Access*, vol. 8, Article ID 150902, 2020.

[26] Y. Jiao, P. Wang, S. Feng, and D. Niyato, "Profit maximization mechanism and data management for data analytics services," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2001–2014, 2018.

[27] G. El Rahi, S. R. Etesami, W. Saad, N. B. Mandayam, and H. V. Poor, "Managing price uncertainty in prosumer-centric energy trading: a prospect-theoretic stackelberg game approach," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 702–713, 2019.

[28] X. Chen, C. Tang, Z. Li, L. Qi, Y. Chen, and S. Chen, "A pricing approach toward incentive mechanisms for participant mobile crowdsensing in edge computing," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1220–1232, 2020.

[29] K. Liu, X. Qiu, W. Chen, X. Chen, and Z. Zheng, "Optimal pricing mechanism for data market in blockchain-enhanced Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9748–9761, 2019.

[30] Y. Li, J. Wan, R. Chen et al., "Top-k vehicle matching in social ridesharing: a price-aware approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 3, pp. 1251–1263, 2021.

[31] Z. Xiong, J. Kang, D. Niyato, P. Wang, and H. Poor, "Cloud/edge computing service management in blockchain networks: multi-leader multi-follower game-based ADMM for pricing,"

IEEE Transactions on Services Computing, vol. 13, no. 2, pp. 356–367, 2020.

[32] W. Zhang, Z. Hong, and W. Chen, "Hierarchical pricing mechanism with financial stability for decentralized crowdsourcing: a smart contract approach," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 750–765, 2021.

[33] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. Wang, "A stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 724–738, 2019.

[34] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[35] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, no. 1, pp. 45–58, 2019.

[36] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[37] W. Li, M. Cao, Y. Wang, C. Tang, and F. Lin, "Mining pool game model and nash equilibrium analysis for PoW-based blockchain networks," *IEEE Access*, vol. 8, Article ID 101049, 2020.

[38] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-Stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities," *IEEE Access*, vol. 7, Article ID 85727, 2019.

[39] F. Yang, W. Zhou, Q. Wu, R. Long, N. N. Xiong, and M. Zhou, "Delegated proof of Stake with downgrade: a secure and efficient blockchain consensus algorithm with downgrade mechanism," *IEEE Access*, vol. 7, Article ID 118541, 2019.

[40] L. Xu, W. Li, F. Zhang, R. Cheng, and S. Tang, "Authorized keyword searches on public key encrypted data with time controlled keyword privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2096–2109, 2020.

[41] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain," *Journal of Medical Systems*, vol. 42, no. 147, pp. 140–214, 2018.

[42] H. Cui, Z. Wan, R. H. Deng, G. Wang, and Y. Li, "Efficient and expressive keyword search over encrypted data in cloud," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 3, pp. 409–422, 2018.

[43] Y. Miao, R. Deng, X. Liu, K. Choo, H. Wu, and H. Li, "Multi-authority attribute-based keyword search over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1667–1680, 2021.

[44] K. Liu, X. Qiu, W. Chen, X. Chen, and Z. Zheng, "Optimal pricing mechanism for data market in blockchain-enhanced Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9748–9761, 2019.