Blockchain-Based Anonymous Data Sharing With Accountability for Internet of Things

Tong Wu, *Member, IEEE*, Weijie Wang, Chuan Zhang, *Member, IEEE*, Weiting Zhang, *Member, IEEE*, Liehuang Zhu, *Senior Member, IEEE*, Keke Gai, *Senior Member, IEEE*, and Haotian Wang

***Abstract*—Blockchain has been a promising infrastructure for enabling secure data sharing for the Internet of Things (IoT). With the widespread of IoT applications, security issues, such as data privacy, anonymity, and accountability become critical concerns for the users, which are essential principles for secure communication in those applications. However, the existing blockchain-based data-sharing schemes mainly consider data privacy. Only a few works can support anonymity with strong, trusted assumptions. Thus, there is a research gap on the anonymity of blockchain-based data sharing for IoT, which does not rely on any trusted party. In this article, we propose a blockchain-based anonymous data-sharing scheme (BA-DS) by adopting a novel public key encryption derived from a ring signature. In BA-DS, we remove the trusted party and ensure anonymity by using an unconditional linkable ring signature and Signature of Knowledge (SoK). During the revocation, we apply blockchain infrastructure to record the valid revocation list and generate a tag for data stored on the cloud, providing solid accountability. The formal security analysis shows that BA-DS is selective indistinguishable secure in the random oracle model. Additionally, we also prove that BA-DS holds anonymity, data privacy, accountability, and authenticity. The extensive experiments indicate that our proposed BA-DS achieves reasonable efficiency in terms of computational complexity, communication overhead, and consumption on the blockchain.**

Manuscript received 15 September 2022; revised 14 October 2022; accepted 10 November 2022. Date of publication 16 November 2022; date of current version 7 March 2023. This work was supported in part by the China Postdoctoral Science Foundation under Grant 2021TQ0041, Grant 2021TQ0042, and Grant 2021M700435; in part by the National Natural Science Foundation of China under Grant 62102027, Grant 62202051, and Grant 62201029; in part by the National Key Research and Development Program of China under Grant 2021YFB2700503; and in part by the Cryptographic Application Industry Chain Supply and Demand Docking Platform of New Energy and Intelligent Connected Vehicle Industry under Grant 2021-0181-1-1. *(Corresponding authors: Keke Gai; Chuan Zhang; Weiting Zhang.)*

Tong Wu is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China, and also with the Defense Innovation Institute, Chinese Academy of Military Science, Beijing 100089, China (e-mail: tongw@bit.edu.cn).

Weijie Wang is with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: weijiew@bit.edu.cn).

Chuan Zhang and Liehuang Zhu are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: chuanz@bit.edu.cn; liehuangz@bit.edu.cn).

Weiting Zhang is with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: wtzhang@bjtu.edu.cn).

Keke Gai is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China, and also with the Innovation Center on Network Security and Information Confrontation, Yangtze Delta Region Academy of Beijing Institute of Technology, Jiaxing 314003, Zhejiang, China (e-mail: gaikeke@bit.edu.cn).

Haotian Wang is with the College of Arts and Science, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: albertht@sas.upenn.edu).

Digital Object Identifier 10.1109/JIOT.2022.3222453

***Index Terms*—Accountability, anonymity, blockchain, data sharing, Internet of Things (IoT).**

I. Introduction

INTERNET of Things is a network convergence of multiple intelligent devices, such as smartphones, sensors, and actuators, enabling communication between devices themselves, devices and people, and exchanging data over the Internet. Internet of Things (IoT) is extensively applied in various fields, such as economy, industry, infrastructure, vehicle service and smart health [1], [2], [3], [4], [5]. Moreover, the continuous increase of data volume from IoT applications leads to data outsourcing for the computation and storage limitation of IoT devices [6], [7], [8]. Till now, the cloud has been widely accepted as a new computing and storage paradigm due to its low maintenance and scalability characteristics. Specifically, cloud storage service provides enterprises and individuals with data storage services with a multitude of storage devices through the low-cost and extensible platform service. There are a vast array of applications on cloud storage [9], [10], [11] that have emerged for IoT. For utilizing the data of IoT devices stored in the cloud, data sharing is the most fundamental functionality for such applications, by which users' data can be shared through cloud storage service. Data sharing can bring many benefits, such as unlocking valuable data assets and allowing for more efficient use of resources. A report by Protenus put the value of data in the U.K. public sector at £6.8 billion, and McKinsey forecasts that the global data-sharing market will drive the growth of $3–5 trillion a year [12]. Open data can make it easier for citizens and governments to collaborate and increase the use of resources. However, there are still some challenges to data sharing that impede the development of IoT applications. First of all, security is the long-term concern of users, ensuring the secrecy of shared data from others. Second, data privacy is another essential requirement of data sharing. For sensitive data, the users are not willing to disclose it to others. Especially, the data is widely treated as a digital asset. Third, anonymity is crucial during the data-sharing procedure. More specifically, taking smart health as an example, the patient's personal information, i.e., identity information, should not be disclosed. For instance, a Protenus report [13] states that nearly 2.6 million patients were affected by a data breach in 2021. No fewer than 60 000 data breaches have been reported in each of the past three years. Last but not least, the accountability [14]

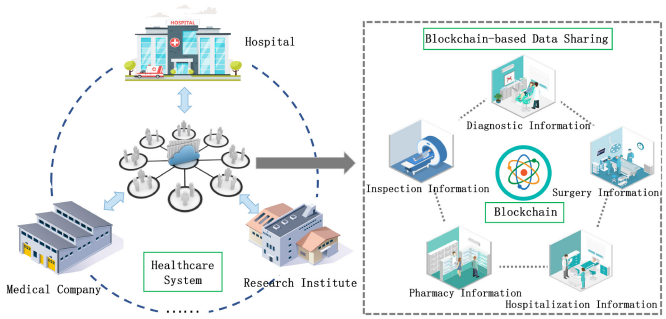

Fig. 1. Smart health platform.

becomes an important feature of data sharing, which ensures the shared data can be traced once needed. More specifically, accountability is defined as an entity being held accountable for its actions in performing a specific task or for a specific scheme [15], with a focus on maintaining transparency and traceability of data usage [16]. Consequently, it is imperative for the establishment of a secure data-sharing scheme with data privacy, anonymity, and accountability for impelling the widespread of IoT applications in the cloud.

Blockchain as a decentralized ledger, which is jointly upheld by a mutually untrusted network of nodes [17]. Through hash chain technology and consensus protocols, data stored in the blockchain is immutable and protected from tampering. Therefore, blockchain is widely considered a promising way for realizing the accountability of data sharing, that all shared data can be tracked by going through the ledger. Additionally, blockchain will take considerable profit to practical applications. For instance, a BIS report [18] figures out that the implementation of blockchain in the healthcare industry could result in savings of up to $100 billion annually by 2025. For the aforementioned advantages of blockchain, i.e., decentralization, openness, autonomy, and tamper resistance, several data-sharing works are proposed based on the blockchain technique. Liu et al. [19] proposed a blockchain-based data-sharing scheme combined with deep reinforcement learning to provide reliability and security. Gao et al. [20] used proxy re-encryption to address data leakage of blockchain-based data sharing due to a single point of attack. These schemes can achieve a high level of privacy protection, but their works only focus on data privacy [21], [22], [23], [24]. Recently, there are a few blockchain-based works [25] that achieved anonymity and accountability by adopting group signatures to track and revoke malicious users. Huang et al. [26] proposed a blockchain-based data-sharing scheme to enable anonymous data sharing between different groups. We note that the existing anonymous data-sharing schemes rely on the assumption of the trusted managers. In practice, it is impractical to have a fully trusted manager in some public IoT platforms, as shown in Fig. 1. The healthcare system has a large number of patient records and medical data collected from IoT devices. Data sharing should ensure that the symptom characteristics of patients in hospitals can be shared with other organizations securely without revealing the individual's privacy. Besides, the other unauthorized organizations are unable to get any knowledge of the shared data. Additionally, the anonymity of the patient should be guaranteed during the entire procedure. In case malicious participants release misleading information, all data in the system should be traceable. Suppose that such an IoT system is built under the existing anonymous data-sharing schemes, once the managers are flawed, the anonymity of participants will be compromised.

To the best of our knowledge, there is a research gap on anonymous data-sharing schemes without trusted managers. In this article, we propose a blockchain-based anonymous data-sharing scheme (BA-DS) with accountability to solve the aforementioned challenging problems. We construct a secure data-sharing paradigm based on a variant of public key encryption to provide security. Then, to conquer the anonymity and privacy requirements, we deploy the unconditional ring signature and Signature of Knowledge (SoK) to construct the tag for each sharing. Additionally, the underlying ring signature is traceable, which enables the accountability of data sharing. We deploy the blockchain infrastructure to record the sharing tag for further tracing the malicious data and revocation lists to eliminate unauthorized manipulation from unauthorized participants. Finally, we analyze our proposed data-sharing scheme from theoretical and experimental aspects. The results show that our data-sharing scheme achieves the expected security features and efficiency in terms of computational complexity, communication overhead, and blockchain infrastructure consumption. Specifically, the contributions of this article are summarized as follows.

1) We propose an anonymous data-sharing scheme without any additional trusted parties, which allows devices to stay unknown to any parties during the data-sharing procedure. The data only can be known by the designed devices, even if some IoT devices have been compromised.
2) We deploy the blockchain infrastructure to record the sharing tag generated by the linkable ring signature and SoK in order to support accountability and authority, respectively. Authority refers to the fact that only IoT devices with SoK can prove that the current signature is signed under the corresponding public key. For tracing the malicious IoT devices, we will allow their public key to be exposed.
3) Our proposed data-sharing scheme supports the ciphertext update if the devices in the system have been revoked. Updating the partial ciphertext instead of regenerating all ciphertexts can reduce the computation overhead once the revocation list is changed or the devices in the ring are revoked. To further reduce the computation burden on the resource-restricted IoT devices, we deploy the edge devices to conduct complex computation during updating ciphertext, and IoT devices only need to generate the new tag during the procedure.
4) We conduct extensive experiments for our proposed data-sharing scheme. Compared with other schemes, our data-sharing scheme possesses practical security features and acceptable efficiency in terms of computational complexity, communication overhead, and consumption on the blockchain.

TABLE I
COMPARISON OF RELATED WORKS

Scheme	Authenticity	Data privacy	Anonymity	Accountability	Security
[27]	✓	×	×	×	×
[19]	✓	✓	×	×	$High$
[22]	✓	✓	×	×	$High$
[23]	✓	✓	×	×	$High$
[21]	✓	✓	×	×	×
[26]	✓	✓	partial	✓	$High$
[20]	✓	✓	×	×	×
[25]	✓	✓	partial	✓	$High$
[28]	✓	✓	✓	×	$High$
Ours	✓	✓	✓	✓	$High$

Organization: The remainder of this article is structured as follows. In Section II, we present the state-of-the-art data-sharing schemes and blockchain-based works. In Section III, we introduce the mathematical and cryptographic primitives involved in our work. The formal definitions of system architecture, threat models, and security models are given in Section IV. Then, in Section V, we provide a concrete construction of our proposed data-sharing scheme and analyze the correctness and security characteristics in Section VI. The performance of the simulation implementation over the blockchain platform with the smart contract is provided in Section VII. Finally, in Section VIII, we conclude our work and introduce some interesting research points for future works.

II. RELATED WORK

In this section, we present some state-of-the-art data-sharing schemes and the existing blockchain-based data-sharing schemes. In Table I, we show the comparison of our proposed scheme with the existing schemes.

A. Data Sharing

The conventional information systems can only exchange data internally [29]. Secure data sharing enables the interconnection between different organizations to be true. The idea of data sharing was first inspired by Goh et al. [30], who introduced a secure peer-to-peer (P2P) file system over an insecure network by a centralized authentication for each user. The conventional data-sharing schemes are mostly based on the centralized architectures [31], [32], [33]. The rapid growth of cloud storage, big data, and IoT promotes the revolution of data-sharing schemes [34], [35], [36], in terms of the security, scalability, integrity, and other characteristics. Wang et al. [37] introduced an attributed-based data-sharing scheme derived from the classic ciphertext-policy attribute-based encryption (CP-ABE) [38] for multiauthority setting in cloud storage. Their mechanism could achieve both forward security and backward security. However, in [39], their scheme was proved to be with security vulnerability. Later, John et al. [27] suggested another attribute-based data-sharing scheme in cloud to improve the key escrow issue in CP-ABE, and reduce the complexity of access control policy. Unfortunately, their work was attacked by two different methods [40], from an authenticated user and unauthenticated user, respectively. To prevent revoked users from decrypting previous ciphertexts in a public cloud storage, a revokable ABE scheme based on the proxy re-encryption technique have been proposed in [33], [41], and [42] to update the ciphertext by cloud. However, the aforementioned works only focus on the security and revocation in data sharing. None of them takes the anonymity of users into consideration. Additionally, the aforementioned data-sharing schemes lack the accountability, which is another practical requirement in real-world applications. Therefore, it is desirable to develop an anonymous data-sharing scheme to cater to the increasing demand on anonymity and accountability in real-world applications driven by data sharing.

B. Blockchain-Based Data Sharing

Blockchain was originally proposed by Nakamoto [43] in 2008 as a digital cryptocurrency based on the distributed ledger technology. The blockchain technology has been widely applied to voting, supply chain, healthcare, IoT, and smart infrastructure [44] for its attractive features. Blockchain enables decentralized data sharing and records the data transactions between data providers and data requesters immutably. At the same time, data exchanges can be monitored, and data transaction history can be kept across nodes in a distributed leaderless manner. There are a few works on developing blockchain-based data-sharing schemes. The blockchain-based data-sharing scheme with traceability and revocability was suggested by Yu et al. [25] for smart factories. In their work, they set the domain administrator for each domain in the smart factories and organize the users by different domains, allowing the users in a domain to access data in other domains by re-encryption. The system is flexible since the assumption of trusted conditions on domain administrators. Once the domain administrators get compromised, the security of the system will be broken. The server-aided revocable bilateral attribute-based encryption (SRB-ABE) [28] was proposed to provide a secure and lightweight bilateral access control system with dynamic user groups in a fog computing system. Their work supports the fine-grained access control over data user and data owner simultaneously, server-aided user revocation with public update, and lightweight data decryption. Their work is derived from matchmaking encryption, which is a time-cost cryptographic primitive. However, the aforementioned blockchain-based data-sharing schemes heavily rely on the trusted party or the complex cryptographic primitives to realize anonymity. Therefore, there is a research gap in realizing an efficient anonymous blockchain-based data-sharing

scheme without a trusted party. Thus, in this work, we exploit the possibility of building an anonymous blockchain-based data-sharing scheme with privacy preservation and accountability, which ensures security characteristics with relatively high efficiency.

III. Preliminary

In this section, we introduce some mathematical preliminaries, cryptographic primitives, and the blockchain infrastructure used in our proposed scheme.

A. Bilinear Group

Definition 1 (Bilinear Group): $\mathbb{G}$ and $\mathbb{G}_T$ constitute a bilinear group if there exists a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $|\mathbb{G}| = |\mathbb{G}_T| = p$.

The formal description of the bilinear pairing is given as follows.

Definition 2 (Bilinear Pairing): Suppose that $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic groups with the same prime order p. Suppose that g and h are generators of $\mathbb{G}$. A bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ holds properties as follows.

1) *Bilinearity:* For any $g, h \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$, $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$.
2) *Nondegeneration:* $e(g, h) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ is the generator of $\mathbb{G}_T$.
3) *Computability:* There exists an efficient algorithm to compute $e(g, h)$, for any $g, h \in \mathbb{G}$.

B. Linkable Ring Signature

Definition 3 (Linkable Ring Signature): A linkable ring signature scheme usually contains five algorithms: 1) **Setup**; 2) **KeyGen**; 3) **Sign**; 4) **Verify**; and 5) **Link**.

1) ***Setup***$(1^\lambda) \to$*param:* On input a security parameter 1^λ, it outputs the system parameters param. The system parameters generated by this algorithm are assumed to be implicit inputs to all algorithms below.
2) ***KeyGen***$(param) \to (sk_i, pk_i)$*:* On input a security parameter 1^λ, it outputs key pair (sk_i, pk_i). This key generation algorithm generates a private signature key sk_i and a public verification key pk_i.
3) ***Sign***$(e, n, Y, sk, M) \to \sigma$*:* On input event-id e, size of ring n, a set of public key denoted as Y, signer's private key sk, and message M, it outputs a ring signature σ.
4) ***Verify***$(e, n, Y, M, \sigma) \to$ ***accept*** *or* ***reject****:* On input event e, size of ring n, a set of public key denoted as Y, message M and signature σ, it outputs **accept** or **reject**. If **accept**, the signature pair is valid.
5) ***Link***$(e, n_1, n_2, Y_1, Y_2, M_1, M_2, \sigma_1, \sigma_2) \to$ *linked/unlinked:* On input event e, size of ring $n1$, $n2$, two sets of public key $Y1$, $Y2$, two valid signature and message pairs $(M_1, \sigma_1, M_2, \sigma_2)$, it outputs **linked** or **unlinked**. If **linked**, the signatures are generated by the same signer.

If a member generates two signatures σ_0 and σ_1 for two messages m_0 and m_1, respectively, then other members can determine whether they were signed by one person based on these two signatures, but cannot determine who signed it actually.

C. Blockchain

Blockchain, as the underlying technology of Bitcoin, is essentially a decentralized database. Currently, blockchain networks can be broadly divided into three categories: 1) public blockchain; 2) private blockchain; and 3) consortium blockchain.

1) *Public Blockchain:* A public blockchain is a consensus blockchain that is open to everyone, where anyone can read data, send transactions, and where transactions can be validly confirmed.
2) *Private Blockchain:* A private blockchain is a blockchain in which the write access is completely held by an organization, the degree of publicness is determined by that organization, and all nodes involved in this blockchain are strictly controlled.
3) *Consortium Blockchain:* A consortium blockchain is a blockchain jointly managed by multiple organizations, each organization or organization manages one or more nodes, and its data is only allowed to be read, written, and sent by different organizations in the system. A consortium chain can be seen as a kind of private blockchain, only with a different degree of privacy and more complex permission design requirements, and higher trustworthiness.

Currently, most of the existing blockchain-based applications of data sharing are deployed on the private blockchain. On the private blockchain, information opacity can occur due to the unequal status of parties, which leads to the limitation of blockchain applications. Compared with the private blockchain, the public blockchain has greater trustworthiness of information, but the public blockchain needs to include more participants. At the same time, it is difficult to guarantee the privacy and security of all participants. In addition, the "completely decentralized" nature of blockchain also hinders the design of the system. In contrast, the consortium blockchain has the feature of "partial decentralization," which is more conducive to the application of data sharing. Smart contracts, one of the success stories in blockchain 2.0, have been widely utilized in a broad range of applications, including those involving IoT [45].

D. Bilinear Diffie–Hellman Problem

The Bilinear Diffie–Hellman (BDH) problem is derived from the discrete logarithm (DL) problem and the Diffie–Hellman (DH) problem. Given $(P, aP, bP, cP)((a, b, c) \in Z_q^*)$, compute $w = \hat{e}(P, P)^{abc} \in G_2$, where $\hat{e}$ is a bilinear map and P is the generating element of G_1, G_1, and G_2 are two groups of order q of prime numbers. Let algorithm A be used to solve the BDH problem with advantage defined as τ, if

$$\Pr \mid A(P, aP, bP, cP) = \hat{e}(P, P)^{abc} \mid \geq \tau.$$

There is no effective algorithm to solve the BDH problem, so it can be assumed that the BDH problem is a difficult problem, which is the BDH assumption.

TABLE II
NOTATIONS

Notation	*Description*
$\mathbb{G}, \mathbb{G}_T$	bilinear groups
H, H_1, H_2	crytographic hash functions
l	the length of message
$List$	revocation list
PK	public key for encryption
SK	private key for decryption
PK_{LRS}	public key of linkable ring signature
SK_{LRS}	private key of linkable ring signature
$\hat{Y}$	public key set in the ring
σ	signature
SoK	signature of knowledge
C	ciphertext
aux	the auxiliaty information
Tag	the tag of message

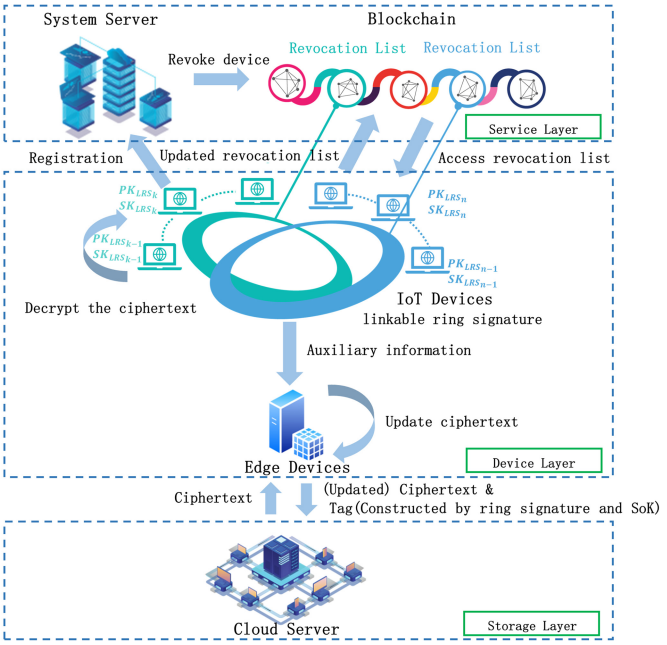

Fig. 2. System architecture.

IV. System Architecture and Security Models

Before we introduce our proposed data-sharing scheme, we first give some notations used in our proposed scheme Table II.

A. System Architecture

The system architecture of our proposed anonymous blockchain-based data-sharing scheme with privacy preservation and accountability is shown in Fig. 2. There are four entities in the system: 1) system server; 2) cloud server; 3) IoT devices; and 4) blockchain infrastructure.

1) *System Server:* The system server, as a trusted entity, provides the registration service for new IoT devices joining the system.
2) *Cloud Server:* The cloud server provides the storage service to store the data and tag for IoT devices.
3) *IoT Devices:* The IoT device is the basic unit in the system which communicates with others. The devices can perform the encryption, decryption, and computation of updating the tag. The devices also can be chosen to form a ring in order to ensure the anonymity of devices.
4) *Edge Device:* The edge device records the auxiliary information and performs the most of heavy computation during the Update later.
5) *Blockchain Infrastructure:* The blockchain records the revocation list. If any device has been removed from the system, the revocation list on the blockchain will be changed by recording a new ledge.

Then, we give a formal definition of our proposed data-sharing scheme as follows, including six algorithms Ω = (Setup, KeyGen, Enc, Dec, Update, Link).

1) Setup(1^λ): Given a security parameter 1^λ, it outputs the public parameters pp.
2) KeyGen(pp, *List*): On input the public parameters pp, and the revocation list List, it first checks whether the device in List. Otherwise, it outputs the public/private keys $(\mathrm{PK}, \mathrm{PK}_{LRS}, \mathrm{SK}, \mathrm{SK}_{LRS})$ for each device.
3) Enc$(List, n, \{\mathrm{PK}_{LRS}\}, \{\mathrm{PK}\}, \mathrm{SK}_{LRS}, m)$: On input the revocation list List, the size of ring n, the set of public keys in the ring $\{\mathrm{PK}_{LRS}\}$ and $\{\mathrm{PK}\}$ for signing the message and encryption, and a message $m \in \{0,1\}^l$, it outputs ciphertext C, the auxiliary information aux, and tag Tag.
4) Dec$(\{\mathrm{PK}\}, List, \mathrm{SK}, C)$: On input the set of public keys in the ring $\{\mathrm{PK}\}$, the revocation list List, the private key SK, and a ciphertext C, it outputs the message m.
5) Update$(\{\mathrm{PK}'\}, List', \mathrm{aux}, C)$: On input the updated public keys in the ring $\{\mathrm{PK}'\}$, the revocation list List$'$, the auxiliary information aux, and ciphertext C, it outputs the new ciphertext C' and new auxiliary information aux$'$.
6) Link$(\mathrm{Tag}_1, \mathrm{Tag}_2)$: On input the tag for C_1 as aux_1, and the tag for C_2 as aux_2. If the link occurs, it outputs **link**; otherwise, **unlink**.

The above descriptions are the system architecture and formal definition of our proposed data-sharing system. Then, in the following part of this section, we will discuss the threat models and formally define the security models of our proposed scheme.

B. Threat Model

In our proposed scheme, the system server is a fully trusted entity, and all interactions with the system server are considered secure. Adversaries can be classified into three types according to their abilities, i.e., malicious participants, cloud adversaries, and external adversaries. The detailed description of the threat model is as follows.

1) *Malicious Participants:* The participants act as sender or receiver. The participants are untrustworthy and can launch any possible attacks, such as collusion attacks and selectively chosen ciphertext attacks (i.e., there are $N-2$ malicious devices that may collude and use their own privacy in self-organized ring signatures to crack the ciphertext and extract the input keys from other parties.). Additionally, malicious participants may attempt to decrypt any unauthorized ciphertexts or pretend to use

any unauthorized device for generating messages. Thus, our work prevents malicious participants from determining who is the actual receiver and obtaining a message that is not for them.

2) *Honest-but-Curious Cloud:* The cloud is responsible for storing encrypted data throughout the data sharing. The cloud is semi-trusted and follows our scheme but always attempt to launch passive attacks, such as ciphertext-only attack to obtain messages.
3) *External Adversary:* An external attacker can gain access to the ciphertext by eavesdropping on the communication in the system and launching a ciphertext-only attack to obtain the message.

C. Security Model

Informally, the device in the ring can use its own private keys to break the ciphertext. $\mathcal{A}$ will launch the selective indistinguishable chosen ciphertext attack (sIND-CCA) with some random oracles.

Definition 4 (sIND-CCA): The security definition of sIND-CCA for our proposed scheme is based on the game between a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ and challenger $\mathcal{C}$.

1) *Initial:* $\mathcal{C}$ conducts Setup and KeyGen to establish the system and sends device's public key in the system to $\mathcal{A}$. $\mathcal{C}$ determine the target device U_T.
2) *Phase 1:* $\mathcal{A}$ can get access the hash oracle $(\mathcal{O}_H, \mathcal{O}_{H_1}, \mathcal{O}_{H_2})$ and decryption oracle $\mathcal{O}_D$ for polynomial time, in order to querying the hash values and the message of the chosen ciphertext.
3) *Challenge:* $\mathcal{A}$ chooses two equal-length message m_0 and m_1 and gives them to $\mathcal{C}$. Then, $\mathcal{C}$ tosses a coin $b \in \{0, 1\}$, and generates the challenged ciphertext C^* with m_b and $\mathtt{PK}^*$). $\mathcal{C}$ sends the challenged ciphertext C^* to $\mathcal{A}$.
4) *Phase 2:* Phase 2 is similar to Phase 1, that $\mathcal{A}$ can get access to the hash oracles and decryption oracle for polynomial time, except that they are not permitted to query the challenge ciphertext.
5) *Guess:* $\mathcal{A}$ outputs a guess b'.

The output of this game is defined as 1 if $b' = b$, and 0, otherwise. If the output of the game is 1, we say that $\mathcal{A}$ succeeded. We denote the advantage of $\mathcal{A}$ winning the game by

$$\mathrm{Adv}_{\mathcal{A}}^{\text{sIND-CCA}}(\lambda) = \left|\Pr[b' = b] - \frac{1}{2}\right|.$$

Informally, the anonymity is to demonstrate that a PPT adversary $\mathcal{A}$ should not tell the signer's public key with a probability greater than $1/n$, where n is the base number of the ring.

Definition 5: The anonymity definition for our proposed scheme is based on the game between a PPT adversary $\mathcal{A}$ and challenger $\mathcal{C}$.

1) *Initial:* $\mathcal{C}$ conducts Setup to generate the system parameters and public keys. $\mathcal{C}$ sends the system parameters to $\mathcal{A}$.
2) *Query:* When a new device joins the system and gets the corresponding public key $\mathtt{PK}$, $\mathcal{A}$ can adaptively query the private key of devices.
3) *Challenge:* $\mathcal{A}$ gives $\mathcal{C}$ a revocation list L, a set of public key in the domain $\hat{Y} = \{Z_1, \ldots, Z_n\}$, the size of domain n, and a message m. $\mathcal{C}$ randomly picks $\pi_R \in \{1, \ldots, n\}$ and conducts Enc with the target public key $\mathtt{PK}_\pi$, to generate the challenged ciphertext C^*, auxiliary information $\mathtt{aux}$ and tag $\mathtt{Tag}$. $\mathcal{C}$ releases C^*, $\mathtt{aux}$ and $\mathtt{Tag}$ to $\mathcal{A}$.
4) *Guess:* $\mathcal{A}$ outputs a guess $\pi' \in \{1, \ldots, n\}$.

The output of this game is defined as 1 if $\pi' = \pi$, and 0, otherwise. If the output of the game is 1, we say that $\mathcal{A}$ succeeded. We denote the advantage by

$$\mathrm{Adv}_{\mathcal{A}}^{\text{Anony}}(\lambda) = \left|\Pr[\pi' = \pi] - \frac{1}{n}\right|.$$

D. Design Goal

BA-DS requires to achieve the following characteristics, including security, anonymity, data privacy, accountability, and authenticity.

1) Security: The system is semantic secure against any PPT attacker under chosen ciphertext attack. The attacker is to query the decryption algorithm by choosing the ciphertext in their way adaptively. Also, at most $N - 2$ devices can collude together and launch an attack on a target device in order to figure out who is the actual sender/receiver.

2) Anonymity: The anonymity ensures that the participants should not tell the actual sender/receiver, except that the sender will be revoked. The bilateral anonymity will provide strong privacy protection without assistance from any trusted party.

3) Data Privacy: The data is sensitive in the data sharing. The data-sharing scheme should provide data privacy which prevents the attacker from eavesdropping or modifying.

4) Accountability: The malicious devices should be tracked in a data-sharing scheme. The system can track all data sent by the malicious device, achieving transparency, and traceability in accountability, which is an important feature in practice.

5) Authentication: Only the valid IoT devices can launch a valid communication, which means none of the devices (even the revoked devices) can generate valid encryption without authorization from the system server.

V. Blockchain-Based Anonymous Data-Sharing Scheme With Accountability for IoT

A. Workflow

In this section, we display the workflow of our proposed BA-DS for the IoT system, as shown in Fig. 3. First of all, when a new IoT device joins the system, the system server is to generate the public/private key pairs for it. All IoT devices in the system can access the revocation list. If the revocation list is changed, all participants have to update their revocation list. Then, if an IoT device is to launch a data sharing with other devices, it encrypts the data and uploads it to the edge device. Later, the edge device will store the auxiliary information for updating their ciphertext, once needed, and upload the ciphertext with a tag to the cloud server. The IoT devices can access the shared data by requesting the cloud and decrypting it locally. Once the revocation list changes,

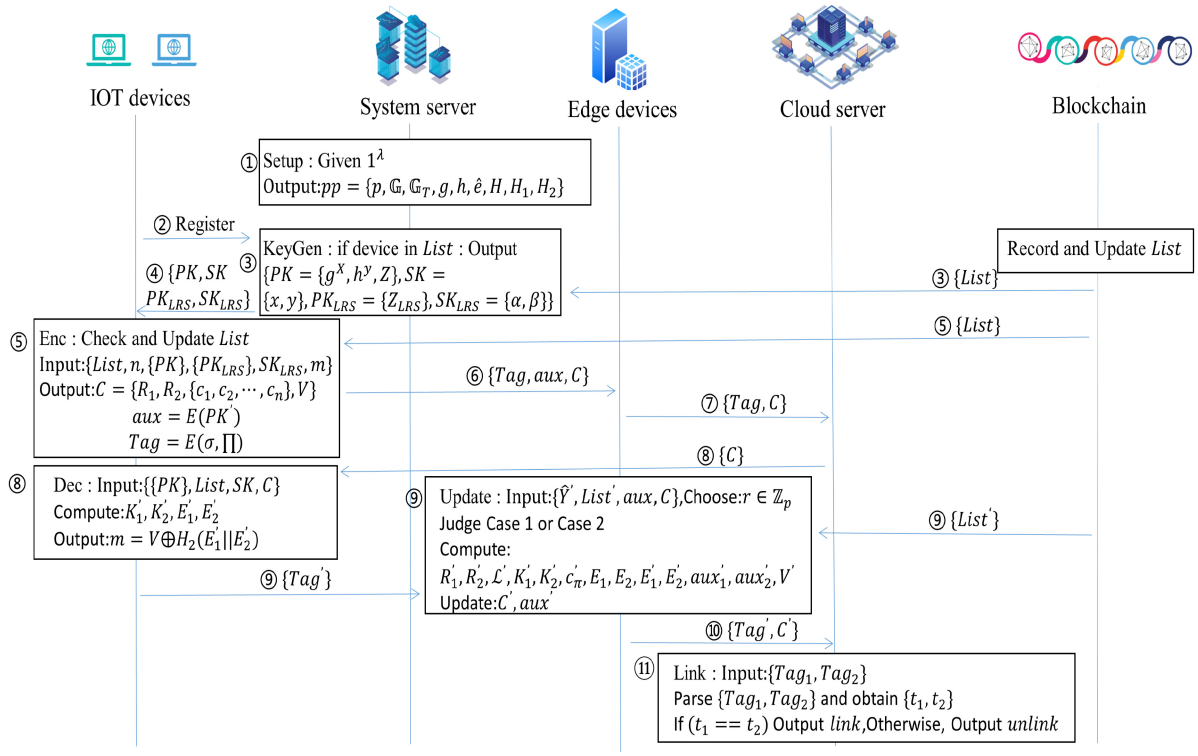

Fig. 3. Workflow of BA-DS with accountability for IoT. ①System server performs system setup. ②–④ IoT device registers with system server and generates public-private key pair and signing key after judging that device is not in *List*. ⑤–⑦ Perform encryption operation, send C, aux, Tag to edge device, send Tag, C to cloud server. ⑧Perform decryption operation. ⑨ and ⑩ Update the ciphertext when the revocation list is changed. New ciphertext C' with new tag Tag$'$ is sent to cloud server. ⑪ Linking to the malicious device.

the edge devices will conduct the update operations on behalf of each IoT device. IoT devices only need to regenerate the new tag for each message and give them to the edge devices. Finally, if any malicious device is found, the cloud server can efficiently track all their data by conducting the link.

B. Concrete Construction

The construction of our proposed BA-DS contains six algorithms $\Omega = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Update}, \mathsf{Link})$, described as follows.

Registration: In our proposed BA-DS, the system server is served as a trusted key management center. When an IoT device has registered on the system server, it can access the system to exchange the data with others. The blockchain infrastructure is to store the revocation list.

1) *Setup*(1^{λ}): Given a security parameter 1^{λ}, set $(p, \mathbb{G}, \mathbb{G}_T, g, \hat{e})$ to be the bilinear group applied in our construction, where $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, and $\mathbb{G}$ is a prime p order group. We choose three hash function as $H : \{0,1\}^* \rightarrow \mathbb{G}$, $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p$, and $H_2 : \mathbb{G}_T \times \mathbb{G}_T \rightarrow \{0,1\}^l$ (l is the length of message). Choose another generator $h \leftarrow \mathbb{G}$, and define the linkable ring signature as LRS. It outputs the public parameter $\mathtt{pp} = (p, \mathbb{G}, \mathbb{G}_T, g, h, \hat{e}, H, H_1, H_2)$.
2) *KeyGen*(pp, *List*): On input the public parameters pp, and the revocation list List, it first checks whether the device in List. If not, the device chooses two random numbers $x, y \leftarrow \mathbb{Z}_p$. The system server computes the public key $\mathtt{PK} = (g^x, h^y, Z = g^x h^y)$. The device keeps the private key $\mathtt{SK} = (x, y)$ secretly. Then, the device requests a pair of signing key $\{\mathtt{PK}_{LRS}, \mathtt{SK}_{LRS}\}$ by the key generation of linkable ring signature LRS.KeyGen(pp), as shown in Algorithm 1. Here, we adopt the work proposed by Liu et al. [46], that $\mathtt{PK}_{LRS} = (Z_{LRS} = g^{\alpha} h^{\beta})$, $\mathtt{SK}_{LRS} = (\alpha, \beta)$. Finally, the system server gives PK, SK, $\mathtt{PK}_{LRS}$, and $\mathtt{SK}_{LRS}$ to device, where SK and $\mathtt{SK}_{LRS}$ are sent through the secure channel.

Algorithm 1: KeyGen(pp, *List*)

Input: Public parameters pp, the revocation list *List*.
Output: The public key PK, $\mathtt{PK}_{LRS}$, the private key SK, $\mathtt{SK}_{LRS}$.
if *the device not in List* **then**
  randomly choose $x, y, \alpha, \beta \leftarrow \mathbb{Z}_p$
  return $\mathtt{PK} \leftarrow (g^x, h^y, Z = g^x h^y)$, $\mathtt{PK}_{LRS} \leftarrow (Z_{LRS} = g^{\alpha} h^{\beta})$, $\mathtt{SK} \leftarrow (x, y)$, and $\mathtt{SK}_{LRS} \leftarrow (\alpha, \beta)$
else
  return Null

Sharing: The IoT device chooses a set of devices to launch a ring. The sender generates the ciphertext binding with the present revocation list and tag for each share, as shown in Algorithm 2. The receiver only can decrypt the data which is sent to it. Neither the other devices in the same ring can decrypt the ciphertext, as shown in

Algorithm 2: $\mathsf{Enc}(List, n, \{\mathtt{PK}\}, \{\mathtt{PK}_{LRS}\}, \mathtt{SK}_{LRS}, m)$

Input: The revocation list *List*, the size of devices to form a ring n, the set of public keys $\{\mathtt{PK}\}$, $\{\mathtt{PK}_{LRS}\}$, the sender's signing key $\mathtt{SK}_{LRS}$, and message m.
Output: The ciphertext C, auxiliary information $\mathtt{aux}$, and tag $\mathtt{Tag}$.

Let $Y = \{Z^1_{LRS}, \ldots, Z^n_{LRS}\}$, $\hat{Y} = \{Z_1, \ldots, Z_n\}$;
Generate a signature $\sigma \leftarrow \mathsf{LRS.Sign}(List, n, Y, \mathtt{SK}_{LRS}, m)$ and the signature of knowledge $\Pi = SoK(m', \mathsf{LRS}, m, \mathsf{E}(\mathtt{PK}_{LRS}, m', m), \mathtt{PK}_{LRS})$
Choose r_1, r_2;
Compute $R_1 = g^{r_1}$, $R_2 = h^{r_2}$;
Choose a set of randomness for each device in the ring $\{c_1, \ldots, c_{\pi-1}, c_{\pi+1}, \ldots, c_n\} \in \mathbb{Z}_p$;
Compute $K_1 = \hat{e}(g^{r_1}h^{r_2} \cdot \prod_{i=1, i\neq\pi}^{n} Z_i^{c_i}, \mathcal{L})$ and $K_2 = \hat{e}(\mathcal{L}^{r_1}, g) \cdot \hat{e}(\mathcal{L}^{\sum_{i=1,i\neq\pi}^{n} c_i}, \mathtt{PK}_1)$;
Select c_π that satisfies $c_1 + c_2 + \cdots + c_n = H_1(\hat{Y}||List||\mathcal{L}||K_1||K_2)$;
Compute $\mathtt{PK}' = (\mathtt{PK}_1^{r_1}, \mathtt{PK}_2^{r_2})$, $E_1 = \hat{e}(\mathcal{L}, \mathtt{PK}'^{c_\pi}_1)$, $E_2 = \hat{e}(\mathcal{L}, \mathtt{PK}'^{c_\pi}_2)$, and $V = m \oplus H_2(E_1||E_2)$;
return ciphertext $C = (R_1, R_2, \{c_1, \ldots, c_n\}, V)$, the auxiliary information $\mathtt{aux} = \mathsf{E}(\mathtt{PK}')$, and the tag $\mathtt{Tag} = \mathsf{E}(\sigma, \Pi)$.

Algorithm 3: $\mathsf{Dec}(\{\mathtt{PK}\}, List, \mathtt{SK}, C)$

Input: The set of public key in the ring $\{\mathtt{PK}\} = \hat{Y}$, the revocation list *List*, the private key $\mathtt{SK}$, and a ciphertext C.
Output: The message m.

Parse the ciphertext C to the following form: $(R_1, R_2, \{c_1, \ldots, c_n\}, V)$;
Compute

$$K_1' = \hat{e}\left(R_1R_2 \cdot \prod_{i=1}^{n} Z_i^{c_i}, \mathcal{L}\right) \cdot \hat{e}\left(g, \mathcal{L}^{-xc_\pi}\right) \cdot \hat{e}\left(h, \mathcal{L}^{-yc_\pi}\right),$$
$$K_2' = \hat{e}(\mathcal{L}, R_1) \cdot \hat{e}\left(\mathcal{L}^x, g^{\sum_{i=1}^{n} c_i - c_\pi}\right);$$

Check $c_1 + \cdots + c_n \stackrel{?}{=} H_1(\hat{Y}||List||\mathcal{L}||K_1'||K_2')$;
Compute

$$E_1' = \hat{e}\left(\mathcal{L}, R_1^{c_\pi x}\right),$$
$$E_2' = \hat{e}\left(\mathcal{L}, R_2^{c_\pi y}\right);$$

return message $m \leftarrow V \oplus H_2(E_1'||E_2')$.

Algorithm 3, nor they can figure out who is the actual receiver.

1) $\mathsf{Enc}(List, n, \{\mathtt{PK}\}, \{\mathtt{PK}_{LRS}\}, \mathtt{SK}_{LRS}, m)$: On input the revocation list List, the size of devices to form a ring n, the set of public keys $\{\mathtt{PK}\}$, $\{\mathtt{PK}_{LRS}\}$, the signing key of sender $\mathtt{SK}_{LRS}$, and a message $m \in \{0, 1\}^l$, it proceeds as follows.
   a) Let $Y = \{Z^1_{LRS}, \ldots, Z^n_{LRS}\}$, $\hat{Y} = \{Z_1, \ldots, Z_n\}$.
   b) Compute $\mathcal{L} = H(\text{List})$.
   c) The signature is computed as follows: Compute $t = \mathcal{L}^\alpha$. Randomly generate

$$r_\alpha, r_\beta, d_1, \ldots, d_{\pi-1}, d_{\pi+1}, \ldots, d_n \in \mathbb{Z}_p.$$

Compute

$$K = g^{r_\alpha}h^{r_\beta} \prod_{i=1, i\neq\pi}^{n} Z^{i\ d_i}_{LRS}$$
$$K' = \mathcal{L}^{r_\alpha} \cdot t^{\sum_{i=1,i\neq\pi}^{n} d_i}.$$

Find d_π such that

$$d_1 + \cdots + d_n = H_1\big(Y||\text{List}||t||m||K||K'\big).$$

Compute

$$\tilde{\alpha} = r_\alpha - d_\pi\alpha, \tilde{\beta} = r_\beta - d_\pi\beta.$$

Generate the signature $\sigma = (t, \tilde{\alpha}, \tilde{\beta}, d_1, \ldots, d_n)$.
   d) Compute the SoK on σ as follows:

$$\Pi = \text{SoK}\big(m', \mathsf{LRS}, m, \mathsf{E}\big(\mathtt{PK}_{LRS}, m', m\big), \mathtt{PK}_{LRS}\big)$$

to prove the signature is signed under the public key $\mathtt{PK}_{LRS}$.
   e) Choose random numbers r_1 and r_2, and compute $R_1 = g^{r_1}, R_2 = h^{r_2}$.
   f) Choose a set of random numbers for each device's public key in the ring $\{c_1, \ldots, c_{\pi-1}, c_{\pi+1}, \ldots, c_n\} \in \mathbb{Z}_p$.
   g) Compute

$$K_1 = \hat{e}\left(g^{r_1}h^{r_2} \cdot \prod_{i=1, i\neq\pi}^{n} Z_i^{c_i}, \mathcal{L}\right)$$
$$K_2 = \hat{e}\big(\mathcal{L}^{r_1}, g\big) \cdot \hat{e}\Big(\mathcal{L}^{\sum_{i=1,i\neq\pi}^{n} c_i}, \mathtt{PK}_1\Big).$$

   h) Choose c_π, that

$$c_1 + c_2 + \cdots + c_n = H_1\Big(\hat{Y}||\text{List}||\mathcal{L}||K_1||K_2\Big).$$

   i) Compute $\mathtt{PK}' = (\mathtt{PK}_1^{r_1}, \mathtt{PK}_2^{r_2})$

$$E_1 = \hat{e}\big(\mathcal{L}, \mathtt{PK}'^{c_\pi}_1\big)$$
$$E_2 = \hat{e}\big(\mathcal{L}, \mathtt{PK}'^{c_\pi}_2\big)$$
$$V = m \oplus H_2(E_1||E_2).$$

   j) Output ciphertext $C = (R_1, R_2, \{c_1, \ldots, c_n\}, V)$, the auxiliary information $\mathtt{aux} = \mathsf{E}(\mathtt{PK}')$, and the tag $\mathtt{Tag} = \mathsf{E}(\sigma, \Pi)$.

$\{C, \mathtt{aux}, \mathtt{Tag}\}$ will be sent by the IoT device to the edge device. The ciphertext and the tag will be uploaded to the cloud. The auxiliary information will be stored on edge devices for conducting the Update later. Note that $\mathsf{E}(\cdot)$ denotes the encryption algorithm. For the security concern, $\mathtt{PK}'$, σ, and Π should be encrypted.

2) $\mathsf{Dec}(\{\mathtt{PK}\}, List, \mathtt{SK}, C)$: On input the set of public key in the ring $\{\mathtt{PK}\}$ denoted as $\hat{Y}$, the revocation list List, the private key $\mathtt{SK}$, and ciphertext C, it proceeds as follows.
   a) Parse C as $(R_1, R_2, \{c_1, \ldots, c_n\}, V)$ and compute

$$K_1' = \hat{e}\left(R_1R_2 \cdot \prod_{i=1}^{n} Z_i^{c_i}, \mathcal{L}\right) \cdot \hat{e}\left(g, \mathcal{L}^{-xc_\pi}\right) \cdot \hat{e}\left(h, \mathcal{L}^{-yc_\pi}\right)$$
$$K_2' = \hat{e}(\mathcal{L}, R_1) \cdot \hat{e}\left(\mathcal{L}^x, g^{\sum_{i=1}^{n} c_i - c_\pi}\right).$$

   b) Check $c_1 + \cdots + c_n \stackrel{?}{=} H_1(\hat{Y}||\text{List}||\mathcal{L}||K_1'||K_2')$.

Algorithm 4: Update($\{\mathtt{PK}'\}$, *List*′, aux, C)

Input: The set of public keys in the ring $\{\mathtt{PK}'\}$ as $\hat{Y}'$, the revocation list *List*′, the auxiliary information aux, and ciphertext C.

Output: The updated ciphertext C' and auxiliary information aux.

Parse the ciphertext C to the following form: $(R_1, R_2, \{c_1, \ldots, c_n\}, V)$;

Decrypt the auxiliary information and parse it into $(\mathtt{aux}_1, \mathtt{aux}_2)$;

Choose a random number r and compute $R'_1 = R_1 g^r$ and $R'_2 = R_2 h^r$;

if *all devices in the ring are still valid as* $\hat{Y}' = \hat{Y}$ **then**

  Compute $\mathcal{L}' = H(List')$,

$$K'_1 = \hat{e}(R'_1 R'_2 \cdot \prod_{i=1}^{n} Z_i^{c_i}, \mathcal{L}')$$

  and

$$K'_2 = \hat{e}(\mathcal{L}', R'_1) \cdot \hat{e}(\mathcal{L}'^{\sum_{i=1, i\neq\pi}^{n} c_i}, \mathtt{PK}_1);$$

  Compute $c'_\pi = H_1(\hat{Y}||List'||\mathcal{L}'||K'_1||K'_2) - \sum_{i=1, i\neq\pi} c_i$;

  Compute

$$E_1 = \hat{e}\big(\mathcal{L}, \mathtt{aux}_1^{c_\pi}\big), E_2 = \hat{e}\big(\mathcal{L}, \mathtt{aux}_2^{c_\pi}\big);$$
$$E'_1 = \hat{e}\big(\mathcal{L}', \mathtt{aux}_1 \cdot \mathtt{PK}_1^r\big)^{c'_\pi},$$
$$E'_2 = \hat{e}\big(\mathcal{L}', \mathtt{aux}_2 \cdot \mathtt{PK}_2^r\big)^{c'_\pi};$$
$$\mathtt{aux}'_1 = \mathtt{aux}_1 \cdot \mathtt{PK}_1^r, \mathtt{aux}'_2 = \mathtt{aux}_2 \cdot \mathtt{PK}_2^r.$$

else

  if *a device* u_j *in the ring is revoked as* $\hat{Y}' \neq \hat{Y}$ **then**

    Compute $R'_1 = g^r$ and $R'_2 = R_2 h^r$;

    Compute $\mathcal{L}' = H(List')$,

$$K'_1 = \hat{e}\left(R'_1 R'_2 \cdot \prod_{i=1, i\neq\pi, j}^{n} Z_i^{c_i}, \mathcal{L}'\right),$$
$$K'_2 = \hat{e}\big(\mathcal{L}'^r, g\big) \cdot \hat{e}\Big(\mathcal{L}'^{\sum_{i=1, i\neq\pi}^{n} c_i}, \mathtt{PK}_1\Big);$$

    Compute
    $c'_\pi = H_1(\hat{Y}'||List'||\mathcal{L}'||K'_1||K'_2) - \sum_{i=1, i\neq\pi, j} c_i$;

    Compute

$$E_1 = \hat{e}(\mathcal{L}, \mathtt{aux}_1^{c_\pi}), E_2 = \hat{e}\big(\mathcal{L}, \mathtt{aux}_2^{c_\pi}\big);$$
$$E'_1 = \hat{e}\big(\mathcal{L}', \mathtt{PK}_1^r\big)^{c'_\pi}, E'_2 = \hat{e}\big(\mathcal{L}', \mathtt{aux}_2 \mathtt{PK}_2^r\big)^{c'_\pi};$$
$$\mathtt{aux}'_1 = \mathtt{PK}_1^r, \mathtt{aux}'_2 = \mathtt{aux}_2 \cdot \mathtt{PK}_2^r.$$

  end

end

Compute $V' = V \oplus H_2(E_1||E_2) \oplus H_2(E'_1||E'_2)$;

retrun the ciphertext and auxiliary information as:

$$C' = \big(R'_1, R'_2, \{c_1, \ldots, c'_\pi, \cdots, c_n\}, V'\big)$$
$$\mathtt{aux} = \mathsf{E}\big(\mathtt{aux}'_1, \mathtt{aux}'_2\big).$$

c) If so, compute $E'_1 = \hat{e}(\mathcal{L}, R_1^{c_\pi x}), E'_2 = \hat{e}(\mathcal{L}, R_2^{c_\pi y})$.

d) Output message m by computing

$$m = V \oplus H_2\big(E'_1||E'_2\big).$$

Once the revocation list is changed, the ciphertext should remain valid. In our proposed BA-DS, we consider two cases during updating ciphertext, as shown in Algorithm 4. The first case is that none of the devices in the ring is revoked. In this case, we only need to update the structure computed with the revocation list. The second case is that there is a device in the ring which is revoked (not the sender). To reduce the computation overhead on IoT devices, we delegate the heavy computation on edge devices. The IoT devices only need to generate a new tag when the update occurs.

1) Update*($\{\mathtt{PK}'\}$, List*′, aux, *C):* On input the set of public key in the ring $\{\mathtt{PK}'\}$ as $\hat{Y}'$, the revocation list List′, the stored auxiliary information aux, and ciphertext C, it proceeds as follows.

   a) Parse the ciphertext C to the following form: $(R_1, R_2, \{c_1, \ldots, c_n\}, V)$.

   b) Decrypt the auxiliary information and parse it into $(\mathtt{aux}_1, \mathtt{aux}_2)$.

   c) Choose a random number $r \in \mathbb{Z}_p$.

   d) There are two cases during the update. One is that all devices in the ring is not in the revocation list. Another case is that there is a device in the ring has been revoked. In the following steps, we discuss the computation of update according to the aforementioned two cases.

   Case 1: If all devices in the ring are still valid that $\hat{Y} = \hat{Y}'$, device just updates the ciphertext for the revocation list changed that List′ ≠ List. Compute $R'_1 = R_1 g^r, R'_2 = R_2 h^r$.Then, compute $\mathcal{L}' = H(\text{List}')$

$$K'_1 = \hat{e}\left(R'_1 R'_2 \prod_{i=1}^{n} Z_i^{c_i}, \mathcal{L}'\right)$$
$$K'_2 = \hat{e}\big(\mathcal{L}', R'_1\big) \cdot \hat{e}\Big(\mathcal{L}'^{\sum_{i=1, i\neq\pi}^{n} c_i}, \mathtt{PK}_1\Big).$$

   Compute

$$c'_\pi = H_1\Big(\hat{Y}||\,\text{List}'\,||\mathcal{L}'||K'_1||K'_2\Big) - \sum_{i=1, i\neq\pi} c_i.$$

   Compute

$$E_1 = \hat{e}\big(\mathcal{L}, \mathtt{aux}_1^{c_\pi}\big), E_2 = \hat{e}\big(\mathcal{L}, \mathtt{aux}_2^{c_\pi}\big)$$
$$E'_1 = \hat{e}\big(\mathcal{L}', \mathtt{aux}_1 \cdot \mathtt{PK}_1^r\big)^{c'_\pi}$$
$$E'_2 = \hat{e}\big(\mathcal{L}', \mathtt{aux}_2 \cdot \mathtt{PK}_2^r\big)^{c'_\pi}$$
$$\mathtt{aux}'_1 = \mathtt{aux}_1 \cdot \mathtt{PK}_1^r$$
$$\mathtt{aux}'_2 = \mathtt{aux}_2 \cdot \mathtt{PK}_2^r.$$

   Case 2: If one device u_j in the ring is revoked that $\hat{Y} \neq \hat{Y}'$, device will update the ciphertext as follows. Compute $R'_1 = g^r, R'_2 = R_2 h^r$.Then, compute $\mathcal{L}' = H(\text{List}')$

$$K'_1 = \hat{e}\left(R'_1 R'_2 \cdot \prod_{i=1, i\neq\pi, j}^{n} Z_i^{c_i}, \mathcal{L}'\right)$$
$$K'_2 = \hat{e}\big(\mathcal{L}'^r, g\big) \cdot \hat{e}\Big(\mathcal{L}'^{\sum_{i=1, i\neq\pi}^{n} c_i}, \mathtt{PK}_1\Big).$$

   Compute

$$c'_\pi = H_1\Big(\hat{Y}'||\text{List}'||\mathcal{L}'||K'_1||K'_2\Big) - \sum_{i=1, i\neq\pi, j} c_i.$$

Algorithm 5: Link($\mathtt{Tag}_1, \mathtt{Tag}_2$)

Input: The tag of ciphertext C_1 as $\mathtt{Tag}_1$, The tag of ciphertext C_2 as $\mathtt{Tag}_2$.
Output: `link` or `unlink`.

Parse $\mathtt{Tag}_1, \mathtt{Tag}_2$ into $(\mathtt{Tag}_{1,1}, \mathtt{Tag}_{1,2})$ and $(\mathtt{Tag}_{2,1}, \mathtt{Tag}_{2,2})$.
if $\mathtt{Tag}_{1,2}$ *and* $\mathtt{Tag}_{2,2}$ *are valid* **then**
  Extract t_1 from $\mathtt{Tag}_{1,1}$ and t_2 from $\mathtt{Tag}_{2,1}$.
  if $t_1 = t_2$ **then**
    return `link`.
  else
    return `unlink`.
else
  return `unlink`.

Compute

$$E_1 = \hat{e}\big(\mathcal{L}, \mathtt{aux}_1^{c_\pi}\big), E_2 = \hat{e}\big(\mathcal{L}, \mathtt{aux}_2^{c_\pi}\big)$$
$$E_1' = \hat{e}\big(\mathcal{L}', \mathtt{PK}_1^r\big)^{c_\pi'}, E_2' = \hat{e}\big(\mathcal{L}', \mathtt{aux}_2\mathtt{PK}_2^r\big)^{c_\pi'}$$
$$\mathtt{aux}_1' = \mathtt{PK}_1^r, \mathtt{aux}_2' = \mathtt{aux}_2 \cdot \mathtt{PK}_2^r.$$

e) Finally, compute

$$V' = V \oplus H_2(E_1||E_2) \oplus H_2\big(E_1'||E_2'\big).$$

Then, update the ciphertext as follows:

$$C' = \big(R_1', R_2', \{c_1, \ldots, c_\pi', \ldots, c_n\}, V'\big)$$

and auxiliary information as $\mathtt{aux}' = \mathsf{E}(\mathtt{aux}_1', \mathtt{aux}_2')$.

After that, the new tag $\mathtt{Tag}'$ will be sent to the edge devices by the IoT devices. The edge devices store the new auxiliary information and sends the new ciphertext with the new tag to the cloud server.

Tracking: The cloud server is assumed to be semi-trusted but not collude with the system server. The cloud stores all data of devices in our system. Additionally, our proposed BA-DS provides the accountability to find out all data generated by malicious devices. Thus, our proposed BA-DS provides the link functionality by the tag stored in the cloud, as shown in Algorithm 5.

1) Link($\mathtt{Tag}_1, \mathtt{Tag}_2$): On input the two tags $\mathtt{Tag}_1$ and $\mathtt{Tag}_2$ on two ciphertext C_1 and C_2, respectively, it outputs `link`, if two ciphertext are generated by the same device; otherwise, `unlink`.
   a) Parse $\mathtt{Tag}_1$ and $\mathtt{Tag}_2$ into $(\mathtt{Tag}_{1,1}, \mathtt{Tag}_{1,2})$ and $(\mathtt{Tag}_{2,1}, \mathtt{Tag}_{2,2})$.
   b) Check the validity of $\mathtt{Tag}_{1,2}$ and $\mathtt{Tag}_{2,2}$.
   c) Extract t_1 from $\mathtt{Tag}_{1,1}$ and t_2 from $\mathtt{Tag}_{2,1}$.

   If $t_1 = t_2$, it output `link`; otherwise, `unlink`.

VI. Security Analysis

Theorem 1: Our proposed BA-DS holds the correctness, if the following equations hold.

Proof: First, we verify c_π is correct by checking

$$c_1 + \cdots + c_n \stackrel{?}{=} H_1\Big(\hat{Y}||\text{List}||\mathcal{L}^x||K_1'||K_2'\Big).$$

If the verification is successful, we can believe that $K_1 = K_1'$ and $K_2 = K_2'$, where

$$K_1 = \hat{e}\left(g^{r_1}h^{r_2} \cdot \prod_{i=1, i\neq\pi}^{n} Z_i^{c_i}, \mathcal{L}\right)$$
$$K_2 = \hat{e}\big(\mathcal{L}^{r_1}, g\big) \cdot \hat{e}\Big(\mathcal{L}^{\sum_{i=1,i\neq\pi}^{n} c_i}, \mathtt{PK}_1\Big)$$
$$K_1' = \hat{e}\left(R_1R_2 \cdot \prod_{i=1}^{n} Z_i^{c_i}, \mathcal{L}\right) \cdot \hat{e}\big(g, \mathcal{L}^{-xc_\pi}\big) \cdot \hat{e}\big(h, \mathcal{L}^{-yc_\pi}\big)$$
$$K_2' = \hat{e}(\mathcal{L}, R_1) \cdot \hat{e}\Big(\mathcal{L}^x, g^{\sum_{i=1}^{n} c_i - c_\pi}\Big).$$

For each E_1, E_2, E_1', and E_2', if $E_1 = E_1'$ and $E_2 = E_2'$, the ciphertext can be decrypted correctly. Here, we list the computation process as follows:

$$V = m \oplus H_2(E_1||E_2), \qquad m = V \oplus H_2\big(E_1'||E_2'\big)$$
$$E_1 = \hat{e}\big(\mathcal{L}, \mathtt{PK}_1'\big)^{c_\pi} = \hat{e}\big(\mathcal{L}, g^{xr_1}\big)^{c_\pi} = \hat{e}\big(\mathcal{L}, R_1^x\big)^{c_\pi} = E_1'$$
$$E_2 = \hat{e}\big(\mathcal{L}, \mathtt{PK}_2'\big)^{c_\pi} = \hat{e}\big(\mathcal{L}, h^{yr_2}\big)^{c_\pi} = \hat{e}\big(\mathcal{L}, R_2^y\big)^{c_\pi} = E_2'. \quad \blacksquare$$

Theorem 2: BA-DS is sIND-CCA secure, if the BDH assumption holds.

Proof: We prove that our proposed BA-DS is secure under sIND-CCA model by conducting a game between a PPT adversary $\mathcal{A}$ and simulator $\mathcal{S}$. Here, the simulator holds a BDH tuple as $(g^a, g^b, g^c, e(g,g)^d)$, to determine that $e(g,g)^d \stackrel{?}{=} e(g,g)^{abc}$. If $e(g,g)^d = e(g,g)^{abc}$, the simulator gets the fact that the given BDH tuple is correct. Otherwise, the given tuple is not a BDH instance.

Initial: $\mathcal{S}$ selects the target device U_T to be broken. $\mathcal{S}$ will replace the target device's public key as $(g^a, h^{y_T}, g^a h^{y_T})$, where a is unknown to $\mathcal{S}$. g^a is an element in the given BDH tuple. h is a generator of $\mathbb{G}$, and y_T is randomly chosen from $\mathbb{Z}_p$. Z_T of the target device's public key is computed by g^a and h^{y_T} to be $Z_T = g^a h^{y_T}$. Here, $\mathcal{A}$ cannot tell the difference between the real-world public key of the target device with the simulated one, since g^a and g^x seem to be random from the view of $\mathcal{A}$.

Phase 1 (Hash Query): $\mathcal{S}$ will replace the hash function by random oracles, as $\mathcal{O}_H$ to be $H(\cdot)$, $\mathcal{O}_{H_1}$ to be $H_1(\cdot)$, and $\mathcal{O}_{H_2}$ to be $H_2(\cdot)$.

$\mathcal{A}$ can make q queries to $\mathcal{O}_H$ for getting the hash value of List_i. For each List_i, when $\mathcal{S}$ gets a query from $\mathcal{A}$, $\mathcal{S}$ chooses a random b_i and returns g^{b_i} to $\mathcal{A}$. For the target list *List**, $\mathcal{S}$ sets the hash value as g^b, which is an elements of the BDH tuple. $\mathcal{S}$ records the hash value in the hash table *Tab*.

$\mathcal{A}$ can make q_1 and q_2 queries to $\mathcal{O}_{H_1}$ and $\mathcal{O}_{H_2}$ for getting the hash value. For each query, $\mathcal{S}$ chooses a random value from $\mathbb{Z}_p$ and a space of fixed-length string, respectively. $\mathcal{S}$ records the query in the hash table Tab_1 and Tab_2, respectively.

Decrypt Query: $\mathcal{A}$ can adaptively query the decryption of ciphertext except that from the target device.

Challenge: $\mathcal{A}$ gives two equal-length messages m_0 and m_1 to $\mathcal{S}$. $\mathcal{S}$ choose $b \in \{0, 1\}$ randomly, and encrypts m_b as the challenged ciphertext C^*. The simulation is conducted as follows.

1) $\mathcal{S}$ generates $\texttt{Tag} = H(List^*)^{\alpha} = g^{b\alpha}$ and corresponding SoK.
2) $\mathcal{S}$ sets $R_1^* = g^c$ and $R_2^* = g^{r_2^*}$.
3) $\mathcal{S}$ chooses a set $\{c_i^*\}_{i=1,i\neq\pi}^n$, and computes

$$K_1^* = \hat{e}\Big(g^c, g^b\Big)\hat{e}\Big(g^{r_2}, g^b\Big)\hat{e}\left(\prod_{i=1,i\neq\pi}^{n} Z_i^{c_i^*}, g^b\right)$$

and

$$K_2^* = \hat{e}\Big(g^c, g^b\Big)\hat{e}\Big(g^b, g^a\Big)^{\sum_{i=1,i\neq\pi}^{n} c_i^*}.$$

4) $\mathcal{S}$ compute $c_{\phi}^* = H_1(\hat{Y}^*, List^*, g^b, K_1^*, K_2^*)$.
5) $\mathcal{S}$ sets $E_1^* = \hat{e}(g,g)^{dc_{\pi}^*}$ and $E_2^* = \hat{e}(g^b, h^{y_T r_2 c_{\pi}^*})$, and computes $V^* = m_b \oplus H_2(E_1^* || E_2^*)$.

Finally, $\mathcal{S}$ sends the challenged ciphertext $C^*$to $\mathcal{A}$

$$C^* = \big(R_1^*, R_2^*, \{c_1, \ldots, c_n\}, V^*\big).$$

Phase 2: This phase is similar to phase 1, that $\mathcal{A}$ can access the hash oracle and the decryption oracle.

Guess: $\mathcal{A}$ finally give a guess b' on b. If $b' = b$, we can say $\mathcal{A}$ wins the game, and $\mathcal{S}$ can solve the BDH tuple with nonnegligible advantage.

Suppose that $\hat{e}(g,g)^d = \hat{e}(g,g)^{abc}$, $\mathcal{S}$ can determine that $(g^a, g^b, g^c, \hat{e}(g,g)^d)$ is a BDH tuple, with the guess $b' = b$. If $\hat{e}(g,g)^d \neq \hat{e}(g,g)^{abc}$, $\hat{e}(g,g)^d$ is a random element of $\mathbb{G}_T$. Then, there is no advantage for $\mathcal{S}$ from the guess of $\mathcal{A}$, since b' is a random guess of $\mathcal{A}$. Thus, we can get the advantage of $\mathcal{A}$ break the security of our proposed scheme as follows:

$$\text{Adv}_{\mathcal{A}}^{\text{sIND-CCA}} \leq \frac{q-1}{2q}\epsilon_{\text{BDH}}$$

where ϵ_{BDH} is negligible, as the advantage of solving the BDH problem. ■

Theorem 3: BA-DS holds anonymity, if the BDH assumption holds and the underlying ring signature holds anonymity.

Proof: We prove that our proposed data-sharing scheme possesses anonymity under our defined model before. If the adversary can break the anonymity of our proposed BA-DS, the simulator is able to utilize such response to break the anonymity of the underlying ring signature [46].

Initial: $\mathcal{S}$ runs Setup to initialize the system. Then, $\mathcal{S}$ runs KeyGen to generate the key pairs for devices in the system.

Query: When a new device is added to the system, $\mathcal{S}$ generates the public key $\texttt{PK} = (g^x, h^y, Z = g^x h^y)$, $\texttt{PK}_{LRS} = (Z_{LRS} = g^{\alpha} h^{\beta})$ to the device, which is computed with randomly chosen (x, y, α, β) as private key.

Challenge: $\mathcal{A}$ provides the target revocation list List, a set of devices to form a ring with public key PK, $\texttt{PK}_{LRS}$, and a challenged message m. $\mathcal{S}$ conducts Enc to generate a response, in which σ is the challenge from the game on the underlying ring signature.

Guess: For the given response, $\mathcal{A}$ will give a guess on π, as π'.

First, we prove that given the signature $\sigma = (t, \tilde{\alpha}, \tilde{\beta}, d_1, \ldots, d_n)$ on message m and the revocation list List will not reveal the actual singer. For each possible public key Z_i, there exists a corresponding private key (α_i, β_i) for any $i \in \{1, \ldots, n\}$. Then, we will have $t_i = H(\text{list})^{\alpha_i}$ for each device. From the view of other devices, t is a random element since α_i is unknown to other devices, except the actual signer. Second, given such a private key (α_i, β_i), there exists a tuple of values $(r_{\alpha_i}, r_{\beta_i})$ such that σ is created using randomness $(r_{\alpha_i}, r_{\beta_i})$. Thus, σ seems to be randomly chosen from the signature space. Third, we show that for $i \in \{1, \ldots, n\}$, the distribution of $(\alpha_i, \beta_i, r_{\alpha_i}, r_{\beta_i})$ is the same. In summary, in the adversary's view, the signature σ is independent of π, i.e., the actual signer. Therefore, we can conclude that there is not an adversary can find π with better probability than a random guess to break the anonymity of BA-DS.

For SoK on σ, according to the inherent property of SoK, it will reveal nothing on an individual's private information to determine the relationship between the signature and public key. Thus, it will not reveal the actual signer.

For the ciphertext C and auxiliary information aux, they are generated only with the receiver public keys. For each device in the ring, $\mathcal{A}$ with their public key cannot identify the actual receiver.

1) $(R_1, R_2, c_1, \ldots, c_n)$ will not reveal the actual receiver for the similar reason as we described for a signature part.
2) The randomness r_1 and r_2 leads to that the distribution of $H_2(E_1||E_2)$ is the same as the every possible results in the space of $H_2(E_1||E_2)$. For the hardness of the BDH problem, even if $\mathcal{A}$ tries to compute V with each c_i for $i \in \{1, n\}$, it is impossible to find the actual $H_2(E_1||E_2)$.

Therefore, if $\mathcal{A}$ can tell who is the actual signer, $\mathcal{S}$ can use such a guess to break the unconditional anonymity of the underlying linkable ring signature. If $\mathcal{A}$ can tell who is the actual receiver, $\mathcal{S}$ can use such a guess to break the BDH problem. Here, we can get the advantage of $\mathcal{A}$ break the anonymity of our proposed scheme as follows:

$$\text{Adv}_{\mathcal{A}}^{\text{Anony}} \leq \frac{1}{n} + \epsilon_{\text{BDH}}$$

where n is the size of ring. ■

Theorem 4: BA-DS holds data privacy, if BA-DS is sIND-CCA, the underlying ring signature is unforgeable, and the SoK is soundness.

Proof: Data privacy requires that no one can learn any information from the communication by eavesdropping or conducting any unauthorized modification. We prove the data privacy of BA-DS from the ciphertext, auxiliary information, and tag.

From the ciphertext C aspect, the security of BA-DS guarantees that the message in a ciphertext is indistinguishable from the view of any PPT adversary. sIND-CCA ensures that the ciphertext seems to be a random string by eavesdropping. Thus, the data privacy gets protected in the ciphertext. Additionally, any modification to the ciphertext will cause the decryption fails for the security of BA-DS, and only the ciphertext generated correctly can be successfully decrypted. From the auxiliary information aux aspect, the auxiliary information is as $\texttt{PK}' = \texttt{PK}^r$, in which r is a randomness. Therefore, the auxiliary information seems to be a random element in the group $\mathbb{G}$. Similarly, any malicious modification will cause the decryption of the updated ciphertext fails for the security

TABLE III
COMPARISON WITH THE STATE-OF-THE-ART OF DATA SHARING

Scheme	CP-WABE-RE	BMGDS-AT	DEDS-TR	Ours
Authenticity	✓	✓	✓	✓
Data privacy	×	✓	✓	✓
Anonymity	×	partial	partial	✓
Accountability	×	✓	✓	✓
Security	×	High	High	High

of BA-DS. Finally, tag Tag is generated by the ring signature with unforgeability and SoK with solid soundness, which ensures no one can conduct the modification on the tag. If it occurs, we can use such an attacker to break the underlying cryptographic primitives.

Therefore, we finish the proof that our proposed BA-DS holds data privacy. If BA-DS is sIND-CCA, the underlying ring signature is unforgeable, and the SoK is soundness. ■

Theorem 5: BA-DS holds accountability if the underlying ring signature is linkable.

Proof: The accountability is based on the linkability of the underlying ring signature and the tamper resistance of the blockchain. For the details of the linkability of the underlying ring signature, please refer to [46]. ■

Theorem 6: BA-DS holds authenticity if the registration records are tamper-resisted.

Proof: In BA-DS, the encryption algorithm is conducted with the device's key pairs, which are recorded in the system server during the registration phase. Since the security of key pairs is based on the intractability of computing the DL problem as g^x, it is intractable for an unauthenticated attacker to forge a valid device's key in order to generate a valid ciphertext on behave of other devices in the system. The device sends a ciphertext that will also generate a tag for it, which is based on a ring signature with unforgeability and a solid SoK. If there is an attacker that can break the authenticity of our proposed BA-DS, we can use such an attacker to break the underlying cryptographic primitives. Therefore, the authenticity of BA-DS gets proved. ■

VII. PERFORMANCE

In this section, we analyze and compare the efficiency of the proposed scheme with that of schemes [25], [26], [27] in terms of theoretical and experimental aspects.

A. Theoretical Analysis

In our experiments, we compare our proposed BA-DS with several other related works (CP-WABE-RE [27], BMGDS-AT [26], and DEDS-TR [25]), as shown in Table III. The common denominator of all four schemes is that they all implement authenticity. In terms of user revocation, CP-WABE-RE and BMGDS-AT do not implement revocation, while DEDS-TR, and our solution provide direct revocation. In terms of user tracking, BMGDS-AT, DEDS-TR, and our solution all implement user tracking, while only CP-WABE-RE does not provide tracking capabilities. It is worth noting that both BMGDS-AT and DEDS-TR rely on the assumption of a group manager to implement user tracking, while our solution does not have such an assumption.

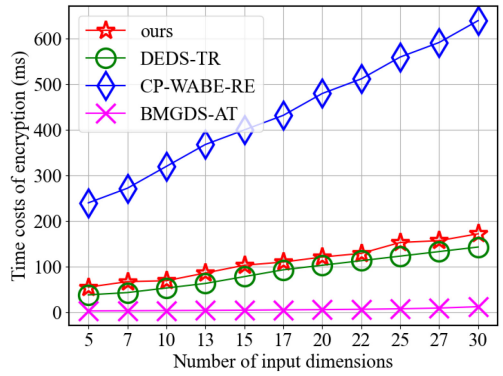

Fig. 4. Performance of Enc algorithm.

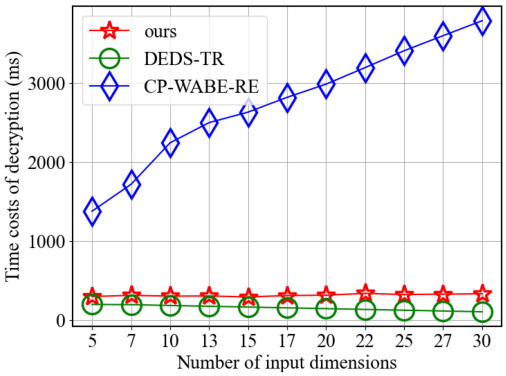

Fig. 5. Performance of Dec algorithm.

B. Experimental Analysis

In this section, we demonstrate the experimental performance of our proposed BA-DS. The experiments give a straight view of computation complexity, communication overhead, and consumption on the blockchain. The platform we used in the experiments is Ethereum, and smart contracts are deployed with Solidity on the Substrate chain. We use Java to implement our simulations, using the java pairing-based cryptography library (JPBC) [47] to execute the BA-DS scheme. Also, we simulated the schemes in CP-WABE-RE, BMGDS-AT, and DEDS-TR under the same conditions. The operating environment is with Windows 7, 64 bit, Intel Core i5-6200 CPU, @2.3 GHz, and 8G RAM. We store the revocation list in the blockchain. The substrate chain is of the 2.0.0 version. Then, we deploy and debug the smart contract on remix (the online smart contract IDE provided by Ethereum), which access the MetaMask wallet through Injected Web3 and interact with the substrate chain through MetaMask.

1) Computation Overhead: The evaluation of the computation overhead is on two aspects: 1) encryption time and 2) decryption time. Here, we only consider the most time-cost algorithms in our proposed BA-DS. The results are shown in Figs. 4 and 5.

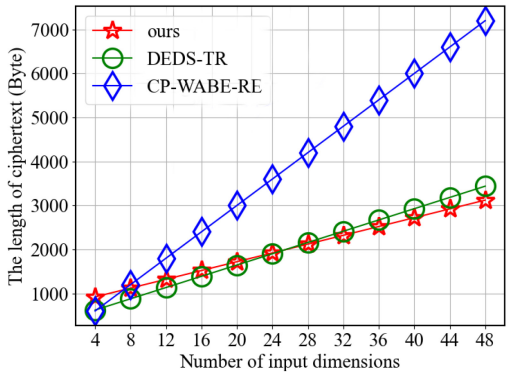

Fig. 6. Length of ciphertext.

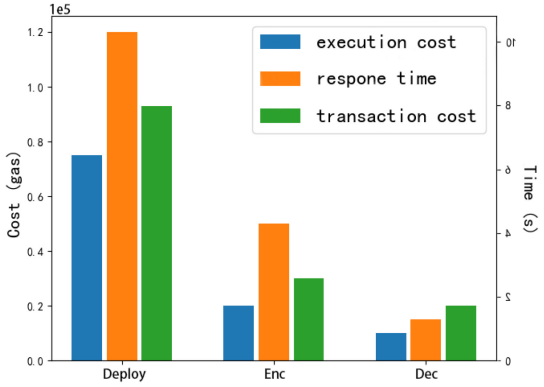

Fig. 7. Consumption on smart contract.

First, we evaluate the computation consumption of encryption when the input dimension is increasing, as shown in Fig. 4. The encryption time of CP-WABE-RE, DEDS-TR, and our scheme shows a linear growth trend with the increase of the number of input dimension, among which CP-WABE-RE shows the most apparent trend compared with the other three schemes. CP-WABE-RE adopts attribute-based encryption, which is less efficient than other schemes and already exceeds 600 ms at an input dimension of 30. The line on BMGDS-AT shows no significant change and stays within 10ms since that BMGDS-AT only focuses on data integrity during data sharing. In other words, BMGDS-AT does not involve the encryption process, which is the most time-cost operation.

Then, we compare the computation consumption of decryption with the increase of input dimension, as shown in Fig. 5. We do not compare it with other schemes because BMGDS-AT does not have a decryption operation. The curve of CP-WABE-RE shows a linear growth trend as the input dimension increases. Our proposed BA-DS remains almost the same as DEDS-TR. The only difference is that the decryption time of our scheme is slightly higher than that of DEDS-TR, where the decryption time of our scheme stays between 200 and 300 ms, while DEDS-TR stays around 200 ms. DEDS-TR uses edge servers or cloud server providers to help users perform many complex calculations. In our scheme, the decryption is conducted by the devices themselves for security concerns.

2) Communication Overhead: The communication overhead is evaluated by comparing the length of the public key and ciphertext [48] of our scheme with that of the other two schemes.

First, the length of the public key in CP-WABE-RE and BMGDS-AT is of a linear increase $\mathcal{O}(n)$ under the input dimension n. The length of the public key of our scheme is constant $\mathcal{O}(1)$, which is half of that in DEDS-TR. Second, the ciphertext length of our proposed data-sharing scheme is linearly increased with the number of input dimension, as shown in Fig. 6. Compared with the other two schemes, the ciphertext of our scheme is on average. However, with the increase of the input dimension, the length of ciphertext in DEDS-TR will finally exceed that in our scheme. As shown in Fig. 6, after the input dimension exceeds 24, the length of the ciphertext in our scheme will be lower than the length of the ciphertext in DEDS-TR. The ciphertext length in CP-WABE-RE grows rapidly with the input dimension and already exceeds 7000 bytes when the input dimension is 48. Consequently, by the performance evaluation on communication overhead, the length of the public key in our scheme is significantly lower than the other three schemes, and the size of the ciphertext is on the average level. Therefore, the aforementioned results indicate that our proposed privacy-preserving data-sharing scheme can guarantee both data privacy and anonymity without losing efficiency.

3) Consumption on Blockchain: The platform we use is Ethereum, and Solidity smart contracts are deployed on the Substrate chain. The operating environment is Windows 7, 64 bit, Intel Corei5-6200 CPU, @2.3 GHz, and 8G RAM. In this experiment, we will first build and start a substrate chain and then access the built substrate chain through MetaMask. Then we will develop, compile, deploy, and debug the smart contract on remix, an online IDE for smart contracts. Remix will access our MetaMask wallet through Injected Web3 and interact with the substrate chain through MetaMask.

As shown in Fig. 7, we use execution cost, transaction cost, and response time of the contract to evaluate the performance on the blockchain. We measured the transaction cost of the entire transaction and the execution cost of the contract code on the Ethereum virtual machine, respectively. The execute cost and transaction cost of the Deploy stage are relatively high, at 0.75×10^5 and 0.95×10^5 gas, respectively. This cost is acceptable because the contract only needs to be deployed once. In the Enc stage, the execution cost and transaction cost are 0.2×10^5 and 0.3×10^5 gas, respectively, which is about twice as large as the Dec stage. This is because additional signatures are required in the Enc stage, which affects gas consumption. On the other hand, the reaction time of the Deploy stage is around 10 s. In contrast, the Enc stage and the Dec stage have relatively low response time, 4 s and less than 2 s, respectively. Our scheme works well in Ethereum, while the response time and gas cost are acceptable.

VIII. Conclusion

In this article, we review the architecture of anonymous data-sharing schemes and clarify the security requirements for building efficient anonymous data-sharing schemes. To overcome the existing technical barriers, we propose a novel

blockchain-based anonymous data-sharing paradigm to ensure security, anonymity, data privacy, and authenticity. In addition, we introduce a blockchain infrastructure to store revocation lists to ensure accountability. This article aims to draw more attention to the security and privacy issues in blockchain-based anonymous data sharing. In this section, we also discuss some open issues for future work. Since blockchain is not scalable for real-time monitoring applications (i.e., IoT), it is a challenging problem to perform real-time data detection. Monitoring IoT devices by introducing blockchain-based and fog computing solutions has potential. For the computation resource-constrained devices, reducing the computation complexity and communication cost is a challenging issue to be addressed. There is potential by introducing a lightweight anonymous data-sharing scheme with less computation and communication overhead. On the other hand, in real-world applications, deniability becomes more vital for some special cases, such as the anonymous broadcast platform, or as an alternative to the forgetness mechanism of the social networks. The expired data should not disclose the sender's information even if the key has been exposed. It is a challenge to the existing data-sharing schemes which are against the impersonation attack and with deniability. The potential could be with the time-locked deniable cryptosystem with forwarding and backward security. Finally, the feasibility of integrating our scheme with current platforms such as Ether 2.0 could be beneficial.

REFERENCES

[1] Q. Huang, L. Wang, and Y. Yang, "Secure and privacy-preserving data sharing and collaboration in mobile healthcare social networks of smart cities," *Security Commun. Netw.*, vol. 2017, pp. 1–12, Aug. 2017.

[2] B. Martens, A. Streel, I. Graef, T. Tombal, and N. Duch-Brown, "Business-to-business data sharing: An economic and legal analysis," JRC Digital Economy Working Paper 2020-05, Eur. Comm., Seville, Spain, 2020.

[3] J. Hathaliya, R. Gupta, S. Tanwar, and P. Sharma, "A smart contract-based secure data sharing scheme in healthcare 5.0," in *Proc. IEEE Globecom Workshops*, 2021, pp. 1–6.

[4] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of Vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, Mar. 2019.

[5] C. Zhang, L. Zhu, C. Xu, J. Ni, C. Huang, and X. Shen, "Location privacy-preserving task recommendation with geometric range query in mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4410–4425, Dec. 2022.

[6] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.

[7] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 72–80, Apr. 2020.

[8] W. Zhang et al., "Optimizing federated learning in distributed industrial IoT: A multi-agent approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3688–3703, Dec. 2021.

[9] H. Deng, Z. Qin, L. Sha, and H. Yin, "A flexible privacy-preserving data sharing scheme in cloud-assisted IoT," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11601–11611, Dec. 2020.

[10] D. Unal, A. Al-Ali, F. O. Catak, and M. Hammoudeh, "A secure and efficient Internet of Things cloud encryption scheme with forensics investigation compatibility based on identity-based encryption," *Future Gener. Comput. Syst.*, vol. 125, no. 99, pp. 433–445, 2021.

[11] Y. Bao, W. Qiu, and X. Cheng, "Secure and lightweight fine-grained searchable data sharing for IoT-oriented and cloud-assisted smart healthcare system," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2513–2526, Feb. 2022.

[12] *Open Data: Unlocking Innovation and Performance With Liquid Information*. San Francisco, CA, USA: McKinsey Company, 2013.

[13] *New View of HHS Breach Data Spotlights Need for Complete Privacy Protection*. Protenus, Baltimore, MD, USA, 2021.

[14] P. Ruth, D. Xu, B. Bhargava, and F. Regnier, "E-notebook middleware for accountability and reputation based trust in distributed data sharing communities," in *Proc. Trust Manag. Second Int. Conf. iTrust*, 2004, pp. 161–175.

[15] A. Baldwin, "Enhanced accountability for electronic processes," in *Proc. Int. Conf. Trust Manag.*, 2004, pp. 319–332.

[16] S. Sundareswaran, A. Squicciarini, and D. Lin, "Ensuring distributed accountability for data sharing in the cloud," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 4, pp. 556–568, Jul./Aug. 2012.

[17] C. Zhang, C. Xu, H. Wang, J. Xu, and B. Choi, "Authenticated keyword search in scalable hybrid-storage blockchains," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, 2021, pp. 996–1007.

[18] "Global blockchain in healthcare market." 2018. [Online]. Available: https://bisresearch.com/industry-report/global-blockchain-in-healthcare-market-2025.html

[19] C. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3516–3526, Jun. 2019.

[20] Y. Gao, Y. Chen, X. Hu, H. Lin, Y. Liu, and L. Nie, "Blockchain based IIoT data sharing framework for sdn-enabled pervasive edge computing," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5041–5049, Jul. 2021.

[21] H. Wang, S. Ma, H. Dai, M. Imran, and T. Wang, "Blockchain-based data privacy management with nudge theory in open banking," *Future Gener. Comput. Syst.*, vol. 110, no. 99, pp. 812–823, 2020.

[22] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, and M. Guizani, "BPDS: A blockchain based privacy-preserving data sharing for electronic medical records," in *Proc. IEEE Global Commun. Conf.*, 2018, pp. 1–6.

[23] L. Hirtan, P. Krawiec, C. Dobre, and J. Batalla, "Blockchain-based approach for e-Health data access management with privacy protection," in *Proc. 24th IEEE Int. Workshop Comput. Aided Model. Design Commun. Links Netw.*, 2019, pp. 1–7.

[24] 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.*, early access, Oct. 10, 2022, doi: 10.1109/JSAC.2022.3213341.

[25] K. Yu, L. Tan, M. Aloqaily, H. Yang, and Y. Jararweh, "Blockchain-enhanced data sharing with traceable and direct revocation in IIoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7669–7678, Nov. 2021.

[26] H. Huang, X. Chen, and J. Wang, "Blockchain-based multiple groups data sharing with anonymity and traceability," *Sci. China Inf. Sci.*, vol. 63, no. 3, 2020, Art. no. 130101.

[27] S. Wang, K. Liang, J. Liu, J. Chen, J. Yu, and W. Xie, "Attribute-based data sharing scheme revisited in cloud computing'," *IEEE Trans. Inf. Forensics Security*, vol. 11, pp. 1661–1673, 2016.

[28] S. Xu, J. Ning, X. Huang, J. Zhou, and R. H. Deng, "Server-aided bilateral access control for secure data sharing with dynamic user groups," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4746–4761, 2021.

[29] S. Anil and E. Gokce, "A federated semantic metadata registry framework for enabling interoperability across clinical research and care domains," *J. Biomed. informat.*, vol. 46, no. 5, pp. 784–794, 2013.

[30] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "SiRiUS: Securing remote untrusted storage," in *Proc. Netw. Distrib. Syst. Security Symp.*, 2003, pp. 1–25.

[31] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in *Proc. Conf. File Storage Technol.*, 2003, pp. 1–14.

[32] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proc. 25th Annu. Int. Cryptol. Conf. Adv. Cryptol.*, 2005, pp. 258–275.

[33] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. 29th IEEE Int. Conf. Comput. Commun. Joint Conf. IEEE Comput. Commun. Soc.*, 2010, pp. 534–542.

[34] Q. Liu, G. Wang, and J. Wu, "Time-based proxy re-encryption scheme for secure data sharing in a cloud environment," *Inf. Sci.*, vol. 258, pp. 355–370, Feb. 2014.

[35] Q. Zhang, X. Zhang, Q. Zhang, W. Shi, and H. Zhong, "Firework: Big data sharing and processing in collaborative edge environment," in *Proc. 4th IEEE Workshop Hot Topics Web Syst. Technol.*, 2016, pp. 20–25.

[36] K. Opuni-Boachie et al., "A secured proxy-based data sharing module in IoT environments using blockchain," *Sensors*, vol. 19, no. 5, p. 1235, 2019.

[37] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "DAC-MACS: Effective data access control for multiauthority cloud storage systems," *IEEE Trans. Inf. Forensics Security*, vol. 8, pp. 1790–1801, 2013.
[38] B. John, S. Amit, and W. Brent, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy*, 2007, pp. 321–334.
[39] J. Hong, K. Xue, and W. Li, "Comments on 'DAC-MACS: Effective data access control for multiauthority cloud storage systems/security analysis of attribute revocation in multiauthority data access control for cloud storage systems," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 1315–1317, 2015.
[40] C. Lan, C. Wang, H. Li, and L. Liu, "Comments on 'attribute-based data sharing scheme revisited in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2579–2580, 2021.
[41] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage systems," in *Proc. 8th ACM Symp. Inf. Comput. Commun. Security*, 2013, pp. 523–528.
[42] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 131–143, Jan. 2013.
[43] N. Satoshi. "Bitcoin: A peer-to-peer electronic cash system." Decentralized Business Review. 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[44] W. Nikolaus, M. Thierry, J. Marco, and P. Fabian, "Privacy-preserving data sharing infrastructures for medical research: Systematization and comparison," *BMC Med. Informat. Decis. Making*, vol. 21, no. 1, pp. 1–13, 2021.
[45] K. Peng, M. Li, H. Huang, C. Wang, S. Wan, and K.-K. R. Choo, "Security challenges and opportunities for smart contracts in Internet of Things: A survey," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12004–12020, Aug. 2021.
[46] J. Liu, M. Au, W. Susilo, and J. Zhou, "Linkable ring signature with unconditional anonymity," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 157–165, Jan. 2014.
[47] A. De Caro and V. Iovino, "JPBC: Java pairing based cryptography," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2011, pp. 850–855.
[48] M. A. Ferrag and L. Shu, "The performance evaluation of blockchain-based security and privacy systems for the Internet of Things: A tutorial," *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17236–17260, Dec. 2021.

Tong Wu (Member, IEEE) received the Ph.D. degree in computer science from the University of Wollongong, Wollongong, NSW, Australia, in 2020.

She is currently with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China; Yangtze Delta Region Academy of Beijing Institute of Technology, Zhejiang, China; and Defense Innovation Institute, Chinese Academy of Military Science, Beijing. Her research interests include applied cryptography, cloud security, and security and privacy in blockchain.

Weijie Wang received the B.S. degree from Xidian University, Xi'an, China, in 2020. He is currently pursuing the master's degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China.

His research interests include federal learning, security, and privacy in blockchain.

Chuan Zhang (Member, IEEE) received the Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2021.

From September 2019 to September 2020, he worked as a visiting Ph.D. student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is currently an Assistant Professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include secure data services in cloud computing, applied cryptography, machine learning, and blockchain.

Weiting Zhang (Member, IEEE) received the Ph.D. degree in communication and information systems with Beijing Jiaotong University, Beijing, China, in 2021.

From November 2019 to November 2020, he was a visiting Ph.D. student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is currently an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include Industrial Internet of Things, edge intelligence, and machine learning for wireless networks.

Liehuang Zhu (Senior Member, IEEE) received the Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2004.

He is currently a Professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include security protocol analysis and design, group key exchange protocols, wireless sensor networks, and cloud computing.

Keke Gai (Senior Member, IEEE) received the B.Eng. degree in automation from the Nanjing University of Science and Technology, Nanjing, China, in 2004, the M.E.T. degree in educational technology from the University of British Columbia, Vancouver, BC, Canada, in 2010, the M.B.A. degree in business administration and the M.S. degree in information technology from Lawrence Technological University, Southfield, MI, USA, in 2009 and 2014, respectively, and the Ph.D. degree in computer science from Pace University, New York, NY, USA, in 2018.

He is currently with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China, and Yangtze Delta Region Academy of Beijing Institute of Technology, Zhejiang, China. His research interests include cyber security, edge computing, cloud computing, blockchain, and reinforcement learning.

Haotian Wang is currently pursuing the undergraduate degree with the College of Arts and Science, University of Pennsylvania, Philadelphia, PA, USA.

His research interests include blockchain-based applications, political economy, and biochemistry.