BCRS-DS: A Privacy-Protected Data Sharing Scheme for IoT Based on Blockchain and Certificateless Ring Signature

Qi Liu

***Abstract*—The Internet of Things (IoT) has created opportunities for collaboration across organizations and domains through data sharing. However, existing data sharing schemes for IoT face challenges such as privacy breaches, inefficiency, and lack of incentives. To address these issues, blockchain provides a promising infrastructure for data sharing due to its decentralization, auditability, and immutability. Consequently, this paper proposes an IoT data sharing scheme based on blockchain and certificateless ring signature (BCRS-DS), which utilizes the certificateless ring signature algorithm to protect the privacy of participants' identities and further improve the authentication efficiency. Furthermore, BCRS-DS includes a decentralized incentive mechanism designed to motivate participants to share data anonymously. Upon completion of the data sharing process, participants can securely submit proof data to a smart contract, ensuring privacy of their identity while demonstrating eligibility for rewards. The effectiveness and practicality of the proposed BCRS-DS scheme is demonstrated through theoretical analysis and experiments.**

***Index Terms*—data sharing, blockchain, certificateless ring signature, privacy protection**

I. Introduction

Internet of Things (IoT) technology has been widely used in smart transportation, smart healthcare, smart grid, smart agriculture, and other fields [1], [2], [3]. International Data Corporation (IDC) predicts that by 2025, there will be 41.6 billion IoT devices, generating 79.4 ZB of data, during 2018-2025, the data generated by these devices is expected to grow at a Compound Annual Growth Rate (CAGR) of 28.7% [4]. The vast amount of data generated by the IoT has spurred extensive research in both industry and academia [5], [6], with a key focus on how to effectively utilize this data to create greater value. Data sharing within the IoT has undoubtedly emerged as a crucial technology for unlocking the potential of big data [7], [8]. Not only does data sharing enhance existing work efficiency, but it also significantly impacts user intelligence and experience quality in future virtual scenarios [9], [10].

However, there are many challenges to data sharing in IoT. Firstly, the significant costs consumed during data collection and transmission [11], and the fact that data has gradually become a valuable asset [12], make it difficult for data providers (DPs) to be sufficiently incentivized to participate in IoT data sharing without proper incentives [13]. Secondly, most of the traditional data sharing is based on centralized cloud servers that are vulnerable to attacks [14], and the increasing concerns of DPs about privacy leakage and data security have led to their increasing reluctance to share their data [15], [16]. Therefore, protecting the privacy of DPs while providing reasonable incentives in the data sharing process has become an urgent problem.

Blockchain is a decentralized distributed ledger where data is stored on multiple network nodes and there is no central authority controlling the entire system [17], [18]. Blockchain technology was originally introduced as the underlying technology for Bitcoin, but has now been used in a wide variety of fields, including finance [19], supply chain management [20], healthcare [21], and more. The inherent decentralization and immutability nature of blockchain provides a compelling solution to the security problem of IoT data sharing [22] [23], in addition to blockchain's smart contracts that offer another possibility for establishing incentives for data sharing. Smart contracts can automatically enforce predetermined conditions, such as providing rewards to DPs based on the amount of data contributed, and this automated incentive mechanism can help increase the motivation for DPs to participate in data sharing [24]. An emerging trend in academic research involves the examination of the potential of blockchain technology in addressing challenges related to IoT data sharing. Wang et al. have put forward a secure and privacy-preserving data sharing protocol for IoT, based on permissioned blockchain [25]. The authors proposed an efficient anonymous authentication scheme utilizing ElGamal promises and one-of-many proofs to ensure that data visitors are authorized while maintaining their true identities concealed from unauthorized entities. Also, Ma et al. suggested a dynamic traceable data sharing scheme based on blockchain technology [26]. In this approach, the blockchain is responsible for user authentication and the storage of ciphertext indexes and public keys to prevent tampering with shared data. Furthermore, a tracking algorithm is employed to identify and include malicious users in a revocation list embedded within the ciphertext. Furthermore, Alshehri et al. have developed a model called Dynamic Secure Access Control (DSA-Block), integrating blockchain into IoT for secure access control and data sharing [27]. This model stores IoT data in cloud servers for secure storage, protecting the data using differential privacy mechanisms. This integration aims to ensure the security and privacy of IoT data, meeting the evolving needs of data sharing and access control in IoT environments.

Although there are many works based on blockchain to address privacy protection in IoT data sharing, most of them do not focus on the privacy protection of the incentive mechanism, and it is easy for DPs to expose the privacy of their identities when obtaining the rewards, which will greatly reduce the willingness of DPs to participate in data sharing. Therefore, in this paper, we propose an IoT data sharing scheme (BCRS-DS) that can realize the privacy protection of incentive mechanism by combining blockchain and ring signature technology. The main contributions can be summarized as follows.

- We propose BCRS-DS, a blockchain-based data sharing scheme for IoT that facilitates data sharing without the need for a trusted third-party.
- We propose a certificateless ring signature algorithm to safeguard the identity privacy of DPs in the context of data sharing, thereby ensuring their anonymity throughout the entire data sharing process. DPs can prove their eligibility for rewards by submitting proof to the smart contract while maintaining their anonymity during reward reception.
- We confirm the efficiency and feasibility of the proposed scheme through theoretical analysis and extensive experiments.

The rest of the paper is organized in the following way. Section II reviews related work. Section III presents some necessary basics. The formal scheme model of BCRS-DS is given in Section IV and BCRS-DS is described in detail in Section V. The correctness and security of BCRS-DS are analyzed in Section VI, and the performance of BCRS-DS is evaluated in Section VII. Finally, Section VIII summarizes the main conclusions of this paper.

II. Related Work

The utilization of blockchain technology in the context of IoT data sharing has recently garnered significant attention and interest. In [28], Li *et al.* proposed a blockchain-based privacy-preserving IoT private data sharing scheme (BPRPDS) where DPs can be rewarded while sharing data. To safeguard identity privacy, they proposed the use of a deniable ring signature algorithm, specifically designed to prevent anonymity abuse by tracking the actual signer through the denial algorithm. However, the execution of the denial algorithm requires the system manager to interact with each member of the ring. Consequently, if the ring members are offline or unwilling to execute the denial algorithm, it becomes impossible to trace the real signer, leading to a high time cost associated with the denial algorithm. Li *et al.* proposed a secure decentralized data transaction model based on blockchain [29], which solves the trust problem between buyer, seller and agent nodes, and increases user incentives for data transactions. Further to this, a secure cost-aware data caching scheme based on blockchain is proposed for the data caching problem to optimize the placement of cached data and prevent tampering of cached data. The scheme employs the quantum particle swarm optimization (QPSO) algorithm to solve the data cache placement problem with maximum content cache gain under the constraints of transmission cost and edge cache size. However, their proposed data transaction strategy does not consider the real-time changes in user content. Liu *et al.* proposed a blockchain-based data sharing solution in a zero-trust IoT environment [30], which aims to ensure anonymity and entity authentication, data privacy and data trustworthiness, as well as participant incentives and fairness. The solution supports the filtering of forged information through the application of smart contracts, effective voting, and consensus mechanisms, thus effectively preventing the problem of sharing spam information by unauthenticated participants. In [31], Yu *et al.* proposed an efficient and secure IoT data sharing scheme (EB-SDSS) based on blockchain, and they ensure data confidentiality and unforgeability by combining symmetric encryption scheme and edge blockchain, and at the same time, they achieve efficient authentication through certificateless signatures, but the scalability of their scheme is not strong. In [32], Wu *et al.* proposed a blockchain-based anonymized data sharing scheme (BA-DS), and to overcome the anonymity and privacy requirements, they deployed traceable ring signatures and Signature of Knowledge (SoK) to construct labels for each share. In addition, BA-DS records shared tags via blockchain to further track malicious data and revoke lists to eliminate unauthorized manipulation by unauthorized participants. In [33], Wang *et al.* proposed a blockchain-based data sharing scheme for the IoT, in which their scheme relies on a data sharing service provider (DSSP) to forward data sharing requests from data requesters (DRs), DPs use anonymous certificates and zero-knowledge proofs to achieve identity privacy protection, and data sharing requests are recorded on a blockchain for easy auditing and monitoring. However, their scheme relies on a centralized third-party service platform, which is vulnerable to single-point-of-failure attacks. Zhang *et al.* proposed an incentive mechanism for IoT data sharing based on smart contract and data quality (DQ) driven [34]. Specifically, they proposed a smart contract to achieve security in the data sharing process, while the proposed DQ evaluation mechanism ensures the quality of shared data. Second, a nested coalition with two-layer Stackelberg game (TLSNC) scheme is designed to maximize the overall social welfare based on the trust scores obtained in the DQ evaluation process while satisfying the constraints of loose and insufficient computational resources. Moreover, they design a smart contract to automate data sharing transactions and use a Trusted Execution Environment (TEE) to accomplish secure computation of shared data. However, they only focused on if the data quality is improved and did not consider the privacy leakage of IoT data sharing.

Most IoT data sharing schemes either do not consider incentives or focus on designing fair incentives while ignoring the privacy preservation of incentives in data sharing, and only a few studies, e.g., [33], consider the problem, but their schemes rely on additional third parties to forward the data sharing requests, require additional payments for the third parties, and are vulnerable to single-point-of-failure attacks. Therefore, in this paper, we propose an IoT data sharing scheme that protects privacy, enables anonymous access to incentives, and operates without the need for third-party assistance. Our scheme enhances efficiency, minimizes waste, and prevents single points of failure.

III. PRELIMINARIES

In this section, we will review the key knowledge utilized in the paper, focusing on bilinear mapping, computational hardness problems, and certificateless ring signature. Table I illustrates the notation used in BCRS-DS and provides a detailed explanation.

TABLE I
NOTATIONS AND DESCRIPTION

Notations	Description
q	A large prime number,which is the order of $\mathbb{G}_1$
$\mathbb{G}_1$	An additive cyclic group of order q
$\mathbb{G}_2$	A multiplicative cyclic group of order q
P	The generator of $\mathbb{G}_1$
e	A bilinear mapping
H_1, H_2, H_3	Collision-resistant Hash functions
P_0	The public key of KGC
s	The KGC's private key, which is also the master key
X_i	The public key of user i
D_i	The partial private key of user i
(x_i, S_i)	The private key of user i
L_{pk}	A list of public keys $\|L\| = n$
L_{ID}	A list of ID
σ	A certificateless ring signature
Enc_{pk}	Asymmetric encryption algorithm
Dec_{sk}	Asymmetric decryption algorithm

A. Bilinear Mapping

Bilinear mapping is one of the construction tools for constructing cryptographic algorithms, which is widely used in blockchain platforms. The specific concept is as follows: let q be a large prime number, $\mathbb{G}_1$ is an additive cyclic group of order q and $\mathbb{G}_2$ is a multiplicative cyclic group of order q, the map e : $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ as a bilinear mapping, it satisfies the following characteristics:

1) *Bilinearity:* $\forall U, V \in \mathbb{G}_1$ and $a, b \in \mathrm{Z}_q^*$,
$$\begin{aligned} e(aV_1, bV_2) &= e(V_1, V_2)^{ab} \\ e(V, V_1 + V_2) &= e(V, V_1)e(V, V_2) \end{aligned}$$
2) *Nondegeneracy:* $\exists V_1, V_2 \in \mathbb{G}_1, e(V_1, V_2) \neq 1_{\mathbb{G}_2}$, where 1 is the identity element in $\mathbb{G}_2$.
3) *Computability:* $\exists V_1, V_2 \in \mathbb{G}_1$, There are efficient algorithms to compute $e(V_1, V_2)$.

The bilinear mapping e with the above properties can be constructed by Weil pairing or Tate pairing on a hypersingular elliptic curve [35].

B. Computational Hard Problems

1) *Definition 1 Elliptic Curve Discrete Logarithm Problem (ECDLP) :* $\forall R, P \in \mathbb{G}_1, x \in \mathbb{Z}_q^*$, it is difficult to compute x so that the equation $R = xP$ holds.
2) *Definition 2 Computational Diffie-Hellman Problem (CDHP):* $\forall a, b \in \mathbb{Z}_q^*$, give $\varUpsilon \in \mathbb{G}_1$, $a\varUpsilon, b\varUpsilon$, it's hard to compute $ab\varUpsilon \in \mathbb{G}$.
3) *Definition 3 Decisional Diffie-Hellman Problem (DDHP):* $\forall a, b \in \mathbb{Z}_q^*$, give $\varUpsilon \in \mathbb{G}_1$,$a\varUpsilon, b\varUpsilon$, it's hard to decide whether $R = ab\varUpsilon$ holds.

C. Certificateless Ring Signature

Ring signature is an anonymous digital signature technology. In a ring signature, all possible signers form a ring, and each possible signer is called a ring member [36]. Certificateless ring signature is an extension of ring signature which introduces certificateless mechanism. In traditional ring signature, each member needs a valid public key certificate [37]. The certificateless ring signature has the following properties.

1) Correctness: If the message is signed according to the correct signing steps and the signature is not tampered with during propagation, then the ring signature satisfies the signature verification equation.
2) Unforgeability: the probability that an external attack can successfully forge a legitimate signature is negligible, even if he can obtain the signature of any message m from a randomized predicate machine that generates ring signatures, without knowing the private keys of any member.
3) Unconditional anonymity: even if the attacker illegally obtains the private keys of all possible signers, the probability that he can identify the real signer does not exceed $1/n$, where n is the number of all possible signers.

IV. SCHEME MODEL

In this section, we present the scheme model, workflow and security requirements.

A. Scheme Model

Fig. 1 shows the scheme model of BCRS-DS, which has four entities: 1) DP, 2) DR, 3) Blockchain, and 4) Key Generation Center (KGC).

- *DP:* The DP is an IoT device designed to share specific data in exchange for rewards.
- *DR:* The DR is an entity that obtains IoT data by paying a certain incentive, which may be a researcher, a business, or another organization.
- *Blockchain:* Blockchain records data sharing requests and data quality scores, while ensuring the stored information is immutable for future auditing and regulation.
- *KGC:* KGC is a key management center, which is mainly responsible for the generation and distribution of public and private key pairs to ensure the security and effectiveness of the cryptosystem.

B. Workflow

Fig. 2 depicts the interaction between the BCRS-DS entities with the specific workflow shown below.

1) The system is initialized and KGC uploads public parameters to the blockchain.
2) Users register with KGC.

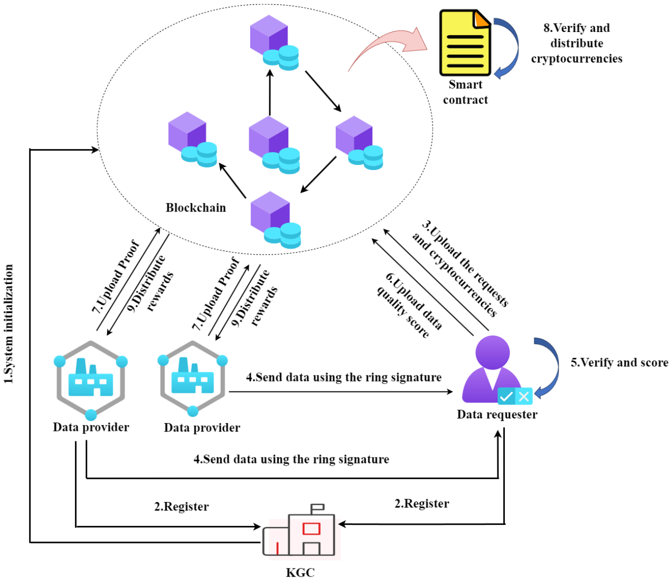

Fig. 1. Scheme model of BCRS-DS.

3) DR publishes a data sharing request on the blockchain and rewards the DP by transferring money to a smart contract.
4) The DP encrypts its private data with the requester's public key after receiving a new data sharing request on the blockchain. It then sends the encrypted data to the DR using a certificateless ring signature.
5) DR receives the ciphertext and signature from multiple DPs, verifies the signature's validity, and then decrypts the ciphertext using its private key to access the data after successful verification.
6) DR evaluates the received data and records the results on the blockchain for monitoring and auditing purposes.
7) The DP retrieves evaluation results from the chain and submits a Proof to the smart contract if eligible for a reward.
8) The smart contract is executed in the TEE to verify the Proof's validity without compromising the DP's privacy.
9) The smart contract rewards users who pass verification.

C. Security Requirements

In the IoT data sharing system, we consider unforgeability and anonymity. According to [38], we consider two types of adversaries.

- *Type I Adversary ($\mathcal{A}_I$):* $\mathcal{A}_I$ can replace any user's public key with a value of his choice, but cannot access the KGC's master secret key.
- *Type II Adversary ($\mathcal{A}_{II}$):* $\mathcal{A}_{II}$ can access the master secret key of KGC but cannot replace the public key of any user.

Definition 4 (Unforgeability of $\mathcal{A}_I$): The unforgeability of BCRS-DS is defined through the game between the adversary $\mathcal{A}_I$ and the challenger $\mathcal{C}$, and BCRS-DS satisfies the unforgeability of $\mathcal{A}_I$ if the probability of $\mathcal{A}_I$ winning in the following game is negligible.

The interaction between the challenger $\mathcal{C}$ and $\mathcal{A}_I$ is as follows.

- *SetUp.* The challenger $\mathcal{C}$ runs the SetUp algorithm to obtain the $pubparas$, then $\mathcal{C}$ sends the $pubparas$ to the adversary $\mathcal{A}_I$. $\mathcal{C}$ keeps the master key s secret. $\mathcal{A}_I$ makes the following query in polynomial time.
- *Hash Queries.* $\mathcal{A}_I$ can submit any queries to $\mathcal{C}$, and $\mathcal{C}$ returns the result to $\mathcal{A}_I$.
- *Public key queries.* $\mathcal{A}_I$ can submit the ID_i of any user to $\mathcal{C}$, and $\mathcal{C}$ is responsible for returning the public key of the corresponding ID_i.
- *Public key replacement queries.* $\mathcal{A}_I$ chooses a new public key $pk_i^{'}$ on ID_i and submits it to $\mathcal{C}$, who replaces the original public key pk_i with $pk_i^{'}$.
- *Partial private key extraction queries.* $\mathcal{A}_I$ submits an identity ID_i to $\mathcal{C}$, and $\mathcal{C}$ returns D_i to $\mathcal{A}_I$.
- *Private key extraction queries.* $\mathcal{A}_I$ chooses an ID_i to submit to $\mathcal{C}$, and if the public key of the ID_i is not replaced, $\mathcal{C}$ returns (x_i, S_i) to $\mathcal{A}_I$, otherwise $\mathcal{C}$ returns $Null$.
- *Certificateless ring signature queries.* The adversary $\mathcal{A}_I$ chooses a set of public key L_{pk} and message m to the challenger $\mathcal{C}$, challenger $\mathcal{C}$ returns to the adversary $\mathcal{A}_I$ a valid signature σ.
- *Forge.* After the above queries, $\mathcal{A}_I$ forgets a certificateless ring signature σ on the message m, which satisfies the following three conditions.
  1) $\mathcal{A}_I$ did not perform a partial private key extraction query in L_{ID}.
  2) The forged signature σ is not obtained by ring signature query.
  3) The forged signature σ is valid.

Definition 5 (Unforgeability of $\mathcal{A}_{II}$): The unforgeability of BCRS-DS is defined through the game between the adversary $\mathcal{A}_{II}$ and the challenger $\mathcal{C}$, and BCRS-DS satisfies the unforgeability of $\mathcal{A}_{II}$ if the probability of $\mathcal{A}_{II}$ winning in the following game is negligible.

The interaction between the challenger $\mathcal{C}$ and $\mathcal{A}_{II}$ is as follows.

- *SetUp.* The challenger $\mathcal{C}$ runs the SetUp algorithm to obtain the $pubparas$, then $\mathcal{C}$ sends the $pubparas$ and s to the adversary $\mathcal{A}_{II}$. $\mathcal{A}_{II}$ makes the following queries in polynomial time.
- *Hash Queries.* $\mathcal{A}_{II}$ can submit any queries to $\mathcal{C}$, and $\mathcal{C}$ returns the result to $\mathcal{A}_{II}$.
- *Public key queries.* $\mathcal{A}_{II}$ can submit the ID_i of any user to $\mathcal{C}$, and $\mathcal{C}$ is responsible for returning the public key of the corresponding ID_i.
- *Private key extraction queries.* $\mathcal{A}_{II}$ submits the ID_i and $\mathcal{C}$ returns the private key corresponding to the ID_i.
- *Certificateless ring signature queries.* $\mathcal{A}_{II}$ selects the list L_{ID}, the corresponding public key list L_{pk}, the message

m is submitted to $\mathcal{C}$, and $\mathcal{C}$ returns a valid signature σ to the adversary.

- *Forge.* Finally, $\mathcal{A}_{II}$ can forge a certificateless signature on message m if the following conditions are satisfied.
  1) $\mathcal{A}_{II}$ never asks for the private key of the user in the L_{ID} list.
  2) The certificateless ring signature σ forged by the adversary is not obtained by ring signature query.
  3) The adversary forged certificateless ring signature σ is valid.

Definition 6 (Unconditional Anonymity): For any Probabilistic Polynomial Time (PPT) adversary, as long as the following form holds, BCRS-DS satisfies the unconditional anonymity.

1) During the user registration phase, the adversary can guess the user's real identity with negligible advantage.
2) In the data sharing phase, the challenger generates public parameters for the adversary, and the adversary can make Hash queries. For any ring signature, the probability that the adversary can guess the true signer is not greater than $1/n$.
3) In the verification and data acquisition phase, the adversary can guess the real identity of the signer with negligible probability.
4) In the get rewards anonymously phase, the adversary cannot obtain additional information, and the probability that the adversary can guess the true signer is not greater than $1/n$.

V. BCRS-DS DETAILS

We detail BCRS-DS in the following steps: 1) System Initialization, 2) User Registration, 3) Data Sharing, 4) Validation and Data Acquisition, and (5) Get Rewards Anonymously.

A. System Initialization

KGC runs the $SetUp$ algorithm and inputs the security parameters λ to generate $\mathbb{G}_1, \mathbb{G}_2, e, q$, where q is a large prime numbers, $\mathbb{G}_1$ is a q order addition cyclic group, P is a generating element of $\mathbb{G}_1$, $\mathbb{G}_2$ is a cyclic group q factorial method, e is a mapping, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$. KGC chooses $s \in \mathbb{Z}_q^*$ as the system master key, and it is also its private key, which is securely retained by KGC. KGC computes $P_0 = sP$ as its own public key. KGC selects three collision-resistant hash functions $\mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3$.

$$\mathrm{H}_1 : \{0,1\}^* \to \mathbb{Z}_q^* \tag{1}$$

$$\mathrm{H}_2 : \{0,1\}^* \to \mathbb{G}_1 \tag{2}$$

$$\mathrm{H}_3 : \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{Z}_q^* \tag{3}$$

The $pubparas = \{\mathbb{G}_1, \mathbb{G}_2, q, e, P, P_0, \mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3\}$ are stored in the blockchain. We define the following algorithms, where $Enc_{pk}()$ denotes the public key encryption algorithm and $Dec_{sk}()$ denotes the decryption algorithm. $\{Enc_{pk}(), Dec_{sk}()\}$ is publicly visible on the blockchain.

B. User Registration

The identity of user i is ID_i, user interacts with KGC, KGC uses the information provided by user to help user generate partial private key, then user generates his private key and public key based on partial private key. The specific algorithm is as follows.

- User i submits his identity information ID_i to KGC, KGC first checks whether ID_i is duplicated, if it is duplicated then it does not do any processing, if it is not duplicated then KGC calculates it as a part of the private key of user i in the following way.

$$Q_i = H_2(ID_i) \tag{4}$$

$$D_i = sQ_i \tag{5}$$

  After the computation, KGC sends (Q_i, D_i) to user i over a secure channel.

- After receiving (Q_i, D_i), user i randomly selects $x_i \in \mathbb{Z}_q^*$ and perform the following calculations.

$$S_i = x_i D_i \tag{6}$$

$$X_i = x_i Q_i \tag{7}$$

$$Y_i = x_i P \tag{8}$$

  User i has private key $sk_i = (x_i, S_i)$ and public key $pk_i = X_i$. Users keep their private key safe and then publish the public key and Y_i on the blockchain.

C. Data Sharing

- DR posts a data sharing request on the blockchain in the form of $req = (seq, M)$, with seq serving as a unique identifier for the task and M providing a brief description of the data required.
- When the DP_π receives a new data sharing request on the blockchain, it uses the asymmetric encryption algorithm $Enc_{pk}()$ to generate the ciphertext $c = Enc_{pk_{DR}}(m)$ if it believes its private data can fulfill the DR's requirements.
- DP_π selects $L_{ID} = \{ID_1, ID_2, ..., ID_n\}$ and $L_{pk} = \{pk_1, pk_2, ..., pk_n\}$, where n is the number of elements in the list, the ID_π and pk_π are contained in L_{ID} and L_{pk}.
- DP_π randomly chooses $T \in \mathbb{G}_1$, $R_i \in \mathbb{G}_1$, $i \in (1, 2, ..., \pi-1, \pi+1, ..., n)$. DP_π performs the following computation to generate the ring signature.

$$T_\pi = x_\pi T \tag{9}$$

$$d = H_3(T \,||\, T_\pi \,||\, X_\pi \,||\, Y_\pi) \tag{10}$$

$$h_i = H_1(c \,||\, R_i \,||\, L_{ID} \,||\, L_{pk}), i \neq \pi \tag{11}$$

$$R_\pi = x_\pi X_\pi - \sum_{i=1, i\neq\pi}^{n} (R_i + h_i X_i) \tag{12}$$

$$h_\pi = H_1(c \,||\, R_\pi \,||\, L_{ID} \,||\, L_{pk}) \tag{13}$$

$$V = (x_\pi + h_\pi) S_\pi \tag{14}$$

  After completing the appeal calculation, the DP_π can generate the signature $\sigma = \{\{R_1, R_2, ..., R_n\}, V, d\}$ and then send the (seq^*, c, σ) to the DR, where seq^* is the unique identification of the message.

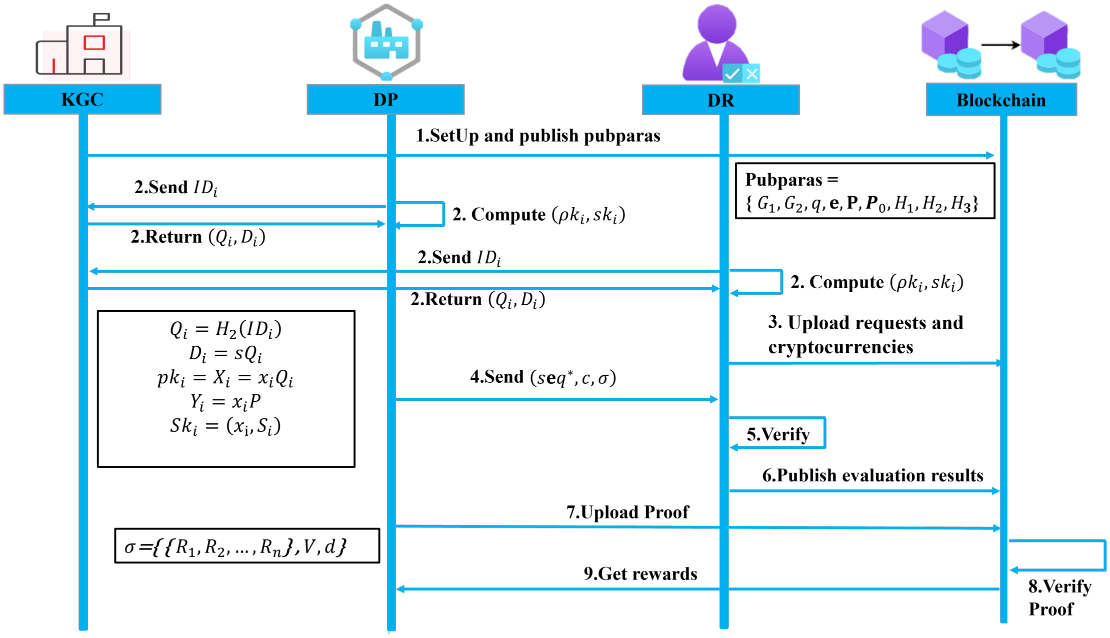
KGC
DP
DR
Blockchain
1.SetUp and publish pubparas
2.Send ID_i
2. Compute (pk_i, sk_i)
2.Return (Q_i, D_i)
Pubparas = { $G_1, G_2, q, \mathbf{e}, \mathbf{P}, P_0, H_1, H_2, H_3$ }
2.Send ID_i
2. Compute (pk_i, sk_i)
2.Return (Q_i, D_i)
3. Upload requests and cryptocurrencies
$Q_i = H_2(ID_i)$
$D_i = sQ_i$
$pk_i = X_i = x_iQ_i$
$Y_i = x_iP$
$Sk_i = (x_i, S_i)$
4.Send (seq^*, c, σ)
5.Verify
6.Publish evaluation results
7.Upload Proof
$\sigma=\{\{R_1, R_2, \ldots, R_n\}, V, d\}$
9.Get rewards
8.Verify Proof

Fig. 2. Interactions between entities in BCRS-DS.

D. Validation and Data Acquisition

- DR receives the message (seq^*, c, σ) sent from multiple DPs, DR first parses the signature as $\sigma = \{\{R_1, R_2, ..., R_n\}, V, d\}$.
- DR computes $h_i^{'} = H_1(c \,||\, R_i \,||\, L_{ID} \,||\, L_{pk})$, and then DR verifies the following equation.

$$e(\sum_{i=1}^{n}(R_i + h_i^{'}X_i), P_0) = e(V, P) \tag{15}$$

- If the equation holds, DR uses the decryption algorithm $Dec_{sk}()$ to decrypt the ciphertext c, resulting in $m = Dec_{sk_{DR}}(c)$.
- After receiving sufficient and satisfactory data, the DR will post an end-of-task announcement on the blockchain regarding the data sharing request. Subsequently, the participants engaging in data sharing will be scored by the DR, and the signature σ and scores of eligible participants will be recorded on the blockchain for supervision audit. These actions serve to ensure transparency and accountability within the data sharing process.

E. Get Rewards Anonymously

Throughout the entire process of providing data, DP maintains anonymity. However, direct acquisition of rewards would compromise user anonymity. If DP's identity privacy is compromised, its willingness to share data will significantly diminish. To address this issue in our proposed scheme, DP uploads a Proof data fragment to the smart contract as evidence of eligibility for receiving rewards and executes the smart contract within a TEE to verify proof validity without compromising its identity privacy.

- When the DP searches the blockchain and finds out it is eligible for a reward, it triggers the smart contract to be executed in the TEE. The DP then offers $Proof = \{seq^*, X_\pi, Y_\pi, T_\pi, T, address\}$ to the smart contract.
- The smart contract finds the corresponding signature information on the blockchain based on the seq^* and computes $d^{'} = H_3(T \,||\, T_\pi \,||\, X_\pi \,||\, Y_\pi)$ verify that the two equations in the following equation are equal.

$$d = d^{'} \tag{16}$$

$$e(T, X_\pi) = e(T_\pi, Q_\pi) \tag{17}$$

- Upon successful validation, the smart contract will transfer funds to the $address$ specified by the DP. Conversely, in case of failed validation, the execution of the smart contract will be terminated.

VI. Security Analysis

In this section, we will carry out the security analysis of BCRS-DS, mainly including the correctness, unforgeability and unconditional anonymity.

Theorem 1 (Correctness): In the ring signature σ, the equations $e(\sum_{i=1}^{n}(R_i + h_i^{'}X_i), P_0) = e(V, P)$ are equal, and the signed message is valid.

Proof: In fact, the equations are equal.

$$
\begin{aligned}
& e(\sum_{i=1}^{n}(R_i + h_i^{'}X_i), P_0) \\
& = e(R_\pi + h_\pi^{'}X_\pi + \sum_{i=1, i\neq\pi}^{n}(R_i + h_i^{'}X_i), P_0) \\
& = e(R_\pi + h_\pi^{'}X_\pi + x_\pi X_\pi - R_\pi, P_0) \\
& = e(X_\pi(x_\pi + h_\pi^{'}), sP) \\
& = e(x_\pi s Q_\pi(x_\pi + h_\pi^{'}), P) \\
& = e(x_\pi D_\pi(x_\pi + h_\pi^{'}), P) \\
& = e(S_\pi(x_\pi + h_\pi^{'}), P) \\
& = e(V, P) \qquad (18)
\end{aligned}
$$

Note that equation (18) holds, indicating that the signer possesses the private key of a ring member and the signed message is valid. ■

Theorem 2 (Unforgeability of $\mathcal{A}_I$): If the CDHP assumption holds, then BCRS-DS is existentially unforgeable under the random oracle model for the adaptive choice of message attack of the first class of adversaries $\mathcal{A}_I$.

Proof: The challenger $\mathcal{C}$ maintains a instance (aP, bP). The purpose of $\mathcal{C}$ is to output the value of $R = abP$. $\mathcal{C}$ can run the $\mathcal{A}_I$ as a subroutine, $\mathcal{A}_I$ interacts with $\mathcal{C}$ as follows.

SetUp. The challenger $\mathcal{C}$ runs the SetUp algorithm to obtain the $pubparas = \{\mathbb{G}_1, \mathbb{G}_2, q, e, P, P_0, H_1, H_2, H_3\}$, where $P_0 = aP$, then $\mathcal{C}$ randomly select $s \in \mathbb{Z}_q^*$ and keeps s in secret. $\mathcal{C}$ sends the $pubparas$ to $\mathcal{A}_I$. $\mathcal{A}_I$ performs the following queries in polynomial time.

H_1 queries. The challenger $\mathcal{C}$ maintains a table T_{H1}, which is initially empty. The adversary $\mathcal{A}_I$ submits a H_1 query on $(c \,||\, R_i \,||\, L_{ID} \,||\, L_{pk})$, if this query already exists in the table T_{H1}, $\mathcal{C}$ returns the results in the table to $\mathcal{A}_I$. If $\mathcal{A}_I$ query submitted no records in the table, the challenger $\mathcal{C}$ randomly chooses $h_i \in \mathbb{Z}_q^*$ and computes $h_i = H_1(c \,||\, R_i \,||\, L_{ID} \,||\, L_{pk})$, returns the h_i to $\kappa \in \mathbb{Z}_q^*$, and adds h_i to T_{H1}.

H_2 queries. The challenger $\mathcal{C}$ maintains a table T_{H2}, which is initially empty. The adversary $\mathcal{A}_I$ submits a H_2 query on ID_i, if this query already exists in the table T_{H2}, $\mathcal{C}$ returns the results in the table to $\mathcal{A}_I$. If ID_i no records in the table T_{H2}, then the challenger $\mathcal{C}$ randomly selected $u_i \in \mathbb{Z}_q^*$, return the $H_2(ID_i) = u_i P$ to $\mathcal{A}_I$, and adds $(ID_i, u_i P)$ to T_{H2}.

H_3 queries. The challenger $\mathcal{C}$ maintains a table T_{H3}, which initially set to an empty table. The adversary $\mathcal{A}_I$ submits a H_3 query on (X_i, Y_i, T_i, T). If the table T_{H3} contains (X_i, Y_i, T_i, T), $\mathcal{C}$ returns the result directly to the $\mathcal{A}_I$. Otherwise,$\mathcal{C}$ randomly chooses $d \in \mathbb{Z}_q^*$, computes $d = H_3((X_i, Y_i, T_i, T))$, adds d to T_{H3} and returns d to $\mathcal{A}_I$.

Public key queries. The challenger $\mathcal{C}$ maintains a table $T_U = (ID_i, D_i, S_i, x_i, X_i, Y_i)$ with T_U initially empty. The adversary $\mathcal{A}_I$ submits a public key query on ID_i, if ID_i already exists in the table T_U, $\mathcal{C}$ returns (X_i, Y_i) to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ randomly chooses $x_i \in \mathbb{Z}_q^*$, and with the help of H_2 queries extract Q_i, then $\mathcal{C}$ adds (X_i, Y_i) to T_U and returns $(X_i = x_i Q_i, Y_i = x_i P)$ to $\mathcal{A}_I$.

Public key replacement queries. The adversary $\mathcal{A}_I$ submits a public key replacement query on ID_i, $\mathcal{C}$ replaces the original (X_i, Y_i) with $(X_i^{'}, Y_i^{'})$ and adds $(X_i^{'}, Y_i^{'})$ to the table T_u.

Partial private key extraction queries. The adversary $\mathcal{A}_I$ submits a partial private key extraction query on ID_i, if $ID_i = ID_j$, $\mathcal{C}$ outputs $\perp$ to terminate the game. If not, $\mathcal{C}$ queries T_U, returns D_i to $\mathcal{A}_I$.

Private key extraction queries. The adversary $\mathcal{A}_I$ submits a private key extraction query on ID_i, $\mathcal{C}$ queries T_U, if the ID_i is within T_U,$\mathcal{C}$ returns (x_i, S_i) to the $\mathcal{A}_I$. Otherwise $\mathcal{C}$ returns $Null$.

Certificateless ring signature queries. The adversary $\mathcal{A}_I$ chooses a set of ID lists L_{ID} and corresponding public key lists $L_{pk} = X_1, X_2, ..., X_n$, where $X_j = bP$. $\mathcal{A}_I$ submits a ring signature query for the ciphertext c. The challenger $\mathcal{C}$ outputs a ring signature for $\mathcal{A}_I$ as follows.

1) $\mathcal{C}$ randomly chooses $T \in \mathbb{G}_1$, $R_i \in \mathbb{G}_1$ where $i \in (1, n)$ and $i \neq j$.
2) $\mathcal{C}$ computes $T_j = x_j T$, $d = H_3(T \,||\, T_\pi \,||\, X_\pi \,||\, Y_\pi)$ and queries d in T_{H3}.
3) $\mathcal{C}$ computes $h_i = H_1(c \,||\, R_i \,||\, L_{ID} \,||\, L_{pk}), i \neq j$ and queries h_i in T_{H_1}.
4) $\mathcal{C}$ computes $R_j = x_j X_j - \sum_{i=1, i\neq j}^{n}(R_i + h_i X_i)$, $h_j = H_1(c \,||\, R_\pi \,||\, L_{ID} \,||\, L_{pk})$, $V = (x_j + h_j)S_j$, through the lookup table T_U.
5) $\mathcal{C}$ returns $\sigma = \{\{R_1, R_2, ..., R_n\}, V, d\}$ on c to $\mathcal{A}_I$.
6) Clearly, the signature σ output by $\mathcal{C}$ is valid because it satisfies the equation $e(\sum_{i=1}^{n}(R_i + h_i X_i), P_0) = e(V, P)$.

If the adversary $\mathcal{A}_I$ is able to forge a valid signature $\sigma = \{c, \{h_1, h_2, \ldots\}, \{R_1, R_2, \ldots\}, V, d\}$ without the partial private key of ID_j being queried, then by the forking lemma [39], the adversary $\mathcal{A}_I$ is able to forge another valid signature $\sigma^{'} = \{c, \{h_1^{'}, h_2^{'}, \ldots\}, , \{R_1, R_2, \ldots\}, V^{'}, d\}$ with a non-negligible probability. Since both signatures are valid, they both satisfy the verification equation, so we can obtain the following equation,

$$e(\sum_{i=1}^{n}(R_i + h_i^{'}X_i), P_0) = e(V^{'}, P) \qquad (19)$$

$$e(\sum_{i=1}^{n}(R_i + h_i^{'}X_i), P_0) = e(V^{'}, P) \qquad (20)$$

From the above two equations, we can obtain,

$$e(P_0, (h_j - h_j^{'})X_j) = e(P, V - V^{'}) \qquad (21)$$

$$e(aP, (h_j - h_j^{'})bP) = e(P, V - V^{'}) \qquad (22)$$

then we get,

$$(h_j - h_j^{'})abP = V - V^{'} \qquad (23)$$

$$abP = (V - V^{'})(h_j - h_j^{'})^{-1} \qquad (24)$$

$\mathcal{C}$ can solve the CDHP problem by running $\mathcal{A}_I$ as a subroutine, but the CDHP problem is hard, so an adversary

$\mathcal{A}$ that can forge a signature does not exist. That is, BCRS-DS suffers from unforgeability for the adaptive choice message attack of the first class of adversaries $\mathcal{A}_I$. ■

Theorem 3 (Unforgeability of $\mathcal{A}_{II}$): If the DDHP assumption holds, BCRS-DS is existentially unforgeable under the stochastic predicate machine model for the adaptive choice of message attack of the second class of adversaries $\mathcal{A}_{II}$.

Proof: The challenger $\mathcal{C}$ maintains a instance (aP, bP, R), where $R \in \mathbb{G}_1$. $\mathcal{C}$ purpose is to determine whether $R = abP$ holds. $\mathcal{C}$ can run the $\mathcal{A}_{II}$ as a subroutine, $\mathcal{A}_{II}$ interacts with $\mathcal{C}$ as follows.

SetUp. The challenger $\mathcal{C}$ runs the SetUp algorithm to obtain the $pubparas = \{\mathbb{G}_1, \mathbb{G}_2, q, e, P, P_0, H_1, H_2, H_3\}$, then $\mathcal{C}$ randomly select $s \in \mathbb{Z}_q^*$. $\mathcal{C}$ sends the $pubparas$ and s to the adversary $\mathcal{A}_{II}$. The adversary $\mathcal{A}_{II}$ performs the following queries in polynomial time.

H_1 queries. The challenger $\mathcal{C}$ maintains a table T_{H1}, which is initially empty. The adversary $\mathcal{A}_{II}$ submits a H_1 query on $(c\,||\,R_i\,||\,L_{ID}\,||\,L_{pk})$, if this query already exists in the table T_{H1}, $\mathcal{C}$ returns the results in the table to $\mathcal{A}_{II}$. If $\mathcal{A}_{II}$ query submitted no records in the table, the challenger $\mathcal{C}$ randomly chooses $h_i \in \mathbb{Z}_q^*$ and computes $h_i = H_1(c\,||\,R_i\,||\,L_{ID}\,||\,L_{pk})$, returns the h_i to $\kappa \in \mathbb{Z}_q^*$, and adds h_i to T_{H1}.

H_2 queries. The challenger $\mathcal{C}$ maintains a table T_{H2}, which is initially empty. The adversary $\mathcal{A}_{II}$ submits a H_2 query on ID_i, if $ID_i = ID_j$, $\mathcal{C}$ sets $Q_j = aP$, then sends Q_i to $\mathcal{A}_{II}$ and adds Q_i to T_{H2}. If this query ID_i already exists in the table T_{H2}, $\mathcal{C}$ returns the results in the table to $\mathcal{A}_{II}$. If ID_i no records in the table T_{H2}, then the challenger $\mathcal{C}$ randomly selected $u_i \in \mathbb{Z}_q^*$, return the $H_2(ID_i) = u_iP$ to $\mathcal{A}_{II}$, and adds (ID_i, u_iP) to T_{H2}.

H_3 queries. The challenger $\mathcal{C}$ maintains a table T_{H3}, which initially set to an empty table. The adversary $\mathcal{A}_{II}$ submits a H_3 query on (X_i, Y_i, T_i, T). If the table T_{H3} contains (X_i, Y_i, T_i, T), $\mathcal{C}$ returns the result directly to the $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}$ randomly chooses $d \in \mathbb{Z}_q^*$, computes $d = H_3(X_i, Y_i, T_i, T)$, adds d to T_{H3} and returns d to $\mathcal{A}_{II}$.

Public key queries. The challenger $\mathcal{C}$ maintains a table $T_U = (ID_i, D_i, S_i, x_i, X_i, Y_i)$ with T_U initially empty. The adversary $\mathcal{A}_{II}$ submits a public key query on ID_i, if $ID_i = ID_j$, $\mathcal{C}$ sets $(X_j = R, Y_j = bP)$, then $\mathcal{C}$ sends (X_i, Y_i) to $\mathcal{A}_{II}$ and adds (X_i, Y_i) to T_U. If ID_i already exists in the table T_U, $\mathcal{C}$ returns (X_i, Y_i) to $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}$ randomly chooses $x_i \in \mathbb{Z}_q^*$, and with the help of H_2 queries extract Q_i, then $\mathcal{C}$ adds (X_i, Y_i) to T_U and returns $(X_i = x_iQ_i, Y_i = x_iP)$ to $\mathcal{A}_{II}$.

Private key extraction queries. The adversary $\mathcal{A}_{II}$ submits a private key extraction query on ID_i, $\mathcal{C}$ queries T_U, if $ID_i = ID_j$, $\mathcal{C}$ stop game. If the ID_i is within T_U, $\mathcal{C}$ returns (x_i, S_i) to the $\mathcal{A}_{II}$. Otherwise $\mathcal{C}$ queries T_{H_2}, randomly chooses $x_i \in \mathbb{Z}_q^*$ computes $D_i = sQ_i, S_i = x_iD_i$. Then $\mathcal{C}$ returns (x_i, S_i) to $\mathcal{A}_{II}$ and adds (x_i, S_i) to T_U.

Certificateless ring signature queries. The adversary $\mathcal{A}_{II}$ chooses a set of ID lists L_{ID} and corresponding public key lists $L_{pk} = X_1, X_2, , ..., X_n$. $\mathcal{A}_{II}$ submits a ring signature query for the ciphertext c. The challenger $\mathcal{C}$ outputs a ring signature for $\mathcal{A}_I$ as follows.

1) $\mathcal{C}$ randomly chooses $T \in \mathbb{G}_1$, $R_i \in \mathbb{G}_1$ where $i \in (1, n)$ and $i \neq j$.
2) $\mathcal{C}$ computes $T_j = x_jT$, $d = H_3(T\,||\,T_\pi\,||\,X_\pi\,||\,Y_\pi)$ and queries d in T_{H_3}.
3) $\mathcal{C}$ computes $h_i = H_1(c\,||\,R_i\,||\,L_{ID}\,||\,L_{pk}), i \neq j$ and queries h_i in T_{H_1}.
4) $\mathcal{C}$ computes $R_j = x_jX_j - \sum_{i=1,i\neq j}^{n}(R_i + h_iX_i)$, $h_j = H_1(c\,||\,R_\pi\,||\,L_{ID}\,||\,L_{pk})$, $V = (x_j + h_j)S_j$, through the lookup table T_U.
5) $\mathcal{C}$ returns $\sigma = \{\{R_1, R_2, ..., R_n\}, V, d\}$ on c to $\mathcal{A}_{II}$.
6) Clearly, the signature σ output by $\mathcal{C}$ is valid because it satisfies the equation $e(\sum_{i=1}^{n}(R_i + h_iX_i), P_0) = e(V, P)$.

Finally, $\mathcal{A}_{II}$ forgers a valid signature σ on the ciphertext c. This forged signature cannot be obtained by ring signature queries, and the adversary never makes a partial private key extraction queries for the members in the L_{ID}. Because $Q_j = aP, Y_j = x_jP = bP, X_j = x_jQ_k = abP$, $X_j = R$, so $\mathcal{C}$ can determine $R = abP$. This solves the DDHP problem, but DDHP is hard, which contradicts the assumption, so $\mathcal{A}_{II}$ cannot forge a valid signature and BCRS-DS is unforgeable against $\mathcal{A}_{II}$. ■

Theorem 4 (Unconditional Anonymity): For any PPT adversary, BCRS-DS satisfies unconditional anonymity.

Proof: We demonstrated the anonymity of BCRS-DS through four stages.

1) In the user registration phase, the user's public key is X_i, where $X_i = x_iQ_i$, because x_i is randomly chosen among $\mathbb{Z}_q^*$, so X_i in $\mathbb{G}_1$ is obeying a random distribution, which means that X_i and the random elements in $\mathbb{G}_1$ are computationally indistinguishable, and so do not give away additional information.
2) In the data sharing phase, R_i $(i \neq \pi)$ in the ring signature $\sigma = \{\{R_1, R_2, ..., R_n\}, V, d\}$ generated by DP_π is randomly chosen so R_i $(i \neq \pi)$ is subject to a random distribution in $\mathbb{G}_1$. Since x_π is randomly chosen in $\mathbb{Z}_q^*$, $R_\pi = x_\pi X_\pi - \sum_{i=1,i\neq\pi}^{n}(x_i + h_iX_i), V = (x_\pi + h_\pi)S_\pi$ are randomly chosen in $\mathbb{G}_1$. So R_π and V are also random distributed in $\mathbb{G}_1$, so they do not compromise the privacy of the signer. $d = H_3(T\,||\,T_\pi\,||\,X_\pi\,||\,Y_\pi)$, and since T and T_π are kept secret and the hash function H_3 is unidirectional, d does not give away the signer's privacy either. So the probability that an adversary outside the ring members can correctly guess the identity of the true signer does not exceed $1/n$.
3) In the validation and data acquisition phase, DR validates the equation $e(\sum_{i=1}^{n}(R_i + h_i^{'}X_i), P_0) = e(V, P)$ without obtaining any additional information either.
4) In the get rewards anonymously phase, DP_π to provide $\{seq^*, X_\pi, Y_\pi, T_\pi, T\}$ to the smart contract, but since the smart contract is executed in TEE, the sensitive information is handled encrypted, and it does not disclose DP_π's private information either.

To summarize, unconditional anonymity of DP is maintained at all stages in BCRS-DS, and the probability that an adversary other than a member of the ring can correctly guess the identity of the true signer is no more than $1/n$, and so BCRS-DS is unconditionally anonymous. ■

VII. Performance Evaluation

This section assesses the performance of BCRS-DS specifically in relation to computation cost and communication cost. BCRS-DS was simulated using Java Pairing Based Cryptography (JPBC) [40] on a machine equipped with an AMD R7-6800H CPU (8 core 4.7 Ghz) and 32GB RAM. Under the same conditions, we simulate [28], [33] and [41].

A. Computational Cost

We conducted 50 experiments to calculate the average. Fig 3 compares the key generation time for BCRS-DS, [28], [33] and [41], with the number of keys generated on the horizontal axis and the time spent on the vertical axis. In Fig. 3, it is evident that under similar conditions, the key generation time cost is highest for [33], while BCRS-DS also incurs a relatively higher key generation time cost. This is because certificateless signature generation requires an additional portion of the key. The time and cost for key generation in [28] is minimal because their scheme allows users to generate their own keys without involving the KGC.

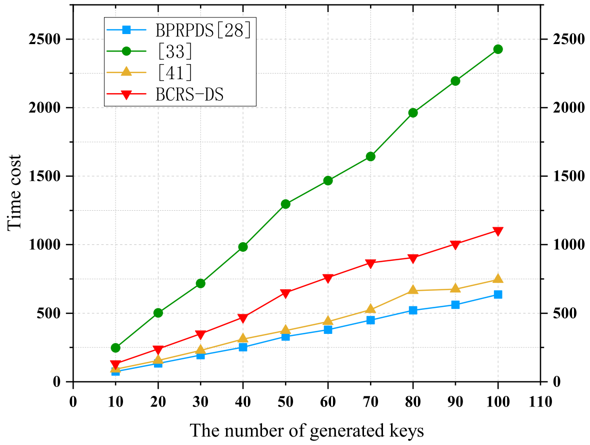

Fig. 3. Comparison of key generation time of different schemes.

In Fig. 4, the comparison of signature or certificate generation time between BCRS-DS and existing schemes is presented. It is evident from the figure that the certificate generation time cost of [33] remains constant even with an increase in ring membership. This is attributed to the utilization of an anonymous certificate technique, obviating the need to form a ring set with other public keys. However, it is important to note that [33] requires the generation of a new key each time an anonymous certificate is generated, hence incurring additional costs. Notably, when the number of ring members is below 30, our scheme exhibits lower computational expenses than [33]. Furthermore, it is observed that the signature time cost of [28] and [41] surpasses that of BCRS-DS. The results of the validation time comparison between BCRS-DS and existing schemes are presented in Figure 5. It is observed that the validation time of [33] remains constant with an increase in the number of members in the ring. Moreover, it is found that the validation time cost is lower for BCRS-DS when the number of members in the ring is less than 40. In the context of practical applications, a ring size of 30 or less is considered sufficient, as evidenced by the use of a ring size of 11 in Monroe [42]. Additionally, the validation time cost of both [28] and [41] is found to be higher than that of BCRS-DS.

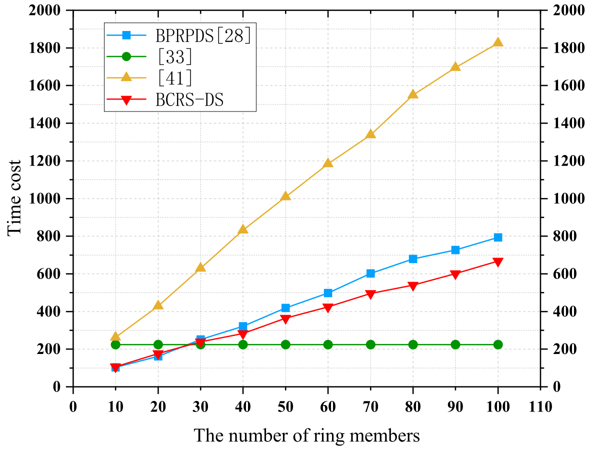

Fig. 4. Comparison of signature or certificate generation time of different schemes.

The slightly higher time cost of key generation for BCRS-DS is due to the nature of certificate-less signatures, but key generation only needs to be performed once. The time cost of BCRS-DS has a better performance when the number of members in the ring is small, although [33] favors the time cost when there are more members in the ring, but the scheme needs a new key each time to generate anonymous certificates, which creates an additional waste of resources. Therefore, overall the computational cost of BCRS-DS has an excellent performance.

B. Communication Cost

We use the length of the public key and the length of the ring signature to measure the communication cost [43]. Fig. 6 compares the signature length of BCRS-DS with those of [28], [33], and [41]. The signature length of [33] remains constant, whereas the signature lengths of both [28], [41], and BCRS-DS increase linearly with the number of ring members. BPRPDS has a slightly higher signature length than ours, and the signature length of BCRS-DS is lower than that of [33] when the number of ring members is 30 or less.

Table II compares the public key and private key length of BCRS-DS with [28], [33], and [41], where $|\mathbb{G}_1| = 512$ B,

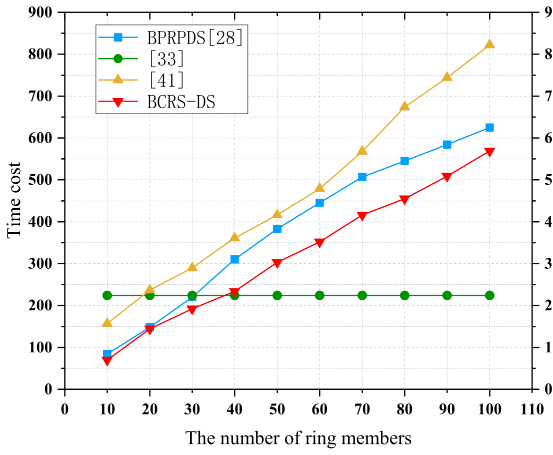

Fig. 5. Comparison of verification time of different schemes.

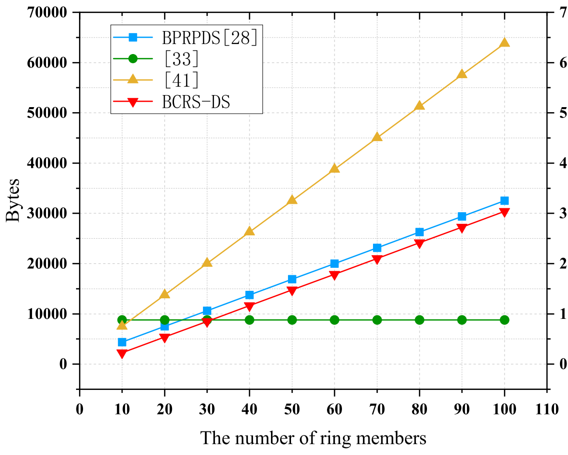

Fig. 6. Length of ring signature or certificate.

$\left|\mathbb{Z}_q^*\right| = 160$ B. BCRS-DS's public key size aligns with that of [28], [33] and [41], while the private key size exceeds that of [28], [41] by the value of $|\mathbb{G}_1|$, resulting in a marginal increase in communication overhead. The private key length of [33] is longer than any other scheme. The private key length of our scheme is 512 B more than [28], [41], but the signature length is smaller than [28], [41]. Taking ring size 20 as an example, the signature length of [28] is 2,117 B more than our scheme, and the signature length of [41] is 8,383 B more than our scheme, and this gap will keep getting bigger as the number of ring members increases. Compared with scheme [33], the private key length of our scheme outperforms that scheme, and the signature length is better than [33] for ring size of 30 or less.

Overall, the communication overhead of BCRS-DS outperforms the scheme [28], [41]. Below ring size 30, the communication overhead of BCRS-DS is superior to [33].

TABLE II
COMPARE THE LENGTH OF THE KEY

Schemes	Size of the public key	Size of the private key
[28]	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{Z}_q^*\rvert$
[33]	$\lvert\mathbb{G}_1\rvert$	$3\lvert\mathbb{G}_1\rvert + \lvert\mathbb{Z}_q^*\rvert$
[41]	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{Z}_q^*\rvert$
BCRS-DS	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_1\rvert + \lvert\mathbb{Z}_q^*\rvert$

VIII. Conclusion

This paper introduces BCRS-DS, a novel approach that eliminates the dependence on traditional trusted third parties, achieves decentralized data sharing. Furthermore, we present a certificate-less ring signature algorithm designed to safeguard the privacy of the DP while enabling anonymous rewards, thereby significantly boosting the motivation for data sharing among participants. Lastly, we provide theoretical analysis and experimental evidence to validate the efficacy and strong performance of the proposed scheme.

References

[1] T. Moulahi, R. Jabbar, A. Alabdulatif, S. Abbas, S. El Khediri, S. Zidi, and M. Rizwan, "Privacy-preserving federated learning cyber-threat detection for intelligent transport systems with blockchain-based security," *Expert Systems*, vol. 40, no. 5, p. e13103, 2023.

[2] T. Mazhar, H. M. Irfan, I. Haq, I. Ullah, M. Ashraf, T. A. Shloul, Y. Y. Ghadi, Imran, and D. H. Elkamchouchi, "Analysis of challenges and solutions of iot in smart grids using ai and machine learning techniques: A review," *Electronics*, vol. 12, no. 1, p. 242, 2023.

[3] J. Liu, L. Shu, X. Lu, and Y. Liu, "Survey of intelligent agricultural iot based on 5g," *Electronics*, vol. 12, no. 10, p. 2336, 2023.

[4] M. Shirer and C. MacGillivray, "The growth in connected iot devices is expected to generate 79.4 zb of data in 2025, according to a new idc forecast," *IDC. com. https://www. idc. com/getdoc. jsp*, 2019.

[5] X. Hao, W. Ren, Y. Fei, T. Zhu, and K.-K. R. Choo, "A blockchain-based cross-domain and autonomous access control scheme for internet of things," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 773–786, 2022.

[6] P. Zhang, M. Sun, Y. Tu, X. Li, Z. Yang, and R. Wang, "Device-edge collaborative differentiated data caching strategy towards aiot," *IEEE Internet of Things Journal*, 2023.

[7] Y. Chen, J. Li, F. Wang, K. Yue, Y. Li, B. Xing, L. Zhang, and L. Chen, "Ds2pm: A data-sharing privacy protection model based on blockchain and federated learning," *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 12 112–12 125, 2023.

[8] Y. Shen, S. Shen, Q. Li, H. Zhou, Z. Wu, and Y. Qu, "Evolutionary privacy-preserving learning strategies for edge-based iot data sharing schemes," *Digital Communications and Networks*, vol. 9, no. 4, pp. 906–919, 2023.

[9] S. A. Khowaja, P. Khuwaja, K. Dev, I. H. Lee, W. U. Khan, W. Wang, N. M. F. Qureshi, and M. Magarini, "A secure data sharing scheme in community segmented vehicular social networks for 6g," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 890–899, 2022.

[10] F. Yang, Y. Qiao, M. Z. Abedin, and C. Huang, "Privacy-preserved credit data sharing integrating blockchain and federated learning for industrial 4.0," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8755–8764, 2022.

[11] Z. Sun, F. Qi, L. Liu, Y. Xing, and W. Xie, "Energy-efficient spectrum sharing for 6g ubiquitous iot networks through blockchain," *IEEE Internet of Things Journal*, vol. 10, no. 11, pp. 9342–9352, 2023.

[12] W. Dai, C. Dai, K.-K. R. Choo, C. Cui, D. Zou, and H. Jin, "Sdte: A secure blockchain-based data trading ecosystem," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 725–737, 2019.

[13] H. Du, J. Wang, D. Niyato, J. Kang, Z. Xiong, and D. I. Kim, "Ai-generated incentive mechanism and full-duplex semantic communica-

tions for information sharing," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 9, pp. 2981–2997, 2023.

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

[15] J. Yu, S. Liu, M. Xu, H. Guo, F. Zhong, and W. Cheng, "An efficient revocable and searchable ma-abe scheme with blockchain assistance for c-iot," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2754–2766, 2023.

[16] S. Liu, J. Yu, Y. Xiao, Z. Wan, S. Wang, and B. Yan, "Bc-sabe: Blockchain-aided searchable attribute-based encryption for cloud-iot," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7851–7867, 2020.

[17] X. Wang, H. Zhu, Z. Ning, L. Guo, and Y. Zhang, "Blockchain intelligence for internet of vehicles: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, 2023.

[18] M. Xu, S. Liu, D. Yu, X. Cheng, S. Guo, and J. Yu, "Cloudchain: A cloud blockchain using shared memory consensus and rdma," *IEEE Transactions on Computers*, vol. 71, no. 12, pp. 3242–3253, 2022.

[19] M. Qin, X. Zhang, Y. Li, and R. M. Badarcea, "Blockchain market and green finance: The enablers of carbon neutrality in china," *Energy Economics*, vol. 118, p. 106501, 2023.

[20] L. A. Risso, G. M. D. Ganga, M. Godinho Filho, L. A. de Santa-Eulalia, T. Chikhi, and E. Mosconi, "Present and future perspectives of blockchain in supply chain management: a review of reviews and research agenda," *Computers & Industrial Engineering*, p. 109195, 2023.

[21] V. Merlo, G. Pio, F. Giusto, and M. Bilancia, "On the exploitation of the blockchain technology in the healthcare sector: A systematic review," *Expert Systems with Applications*, vol. 213, p. 118897, 2023.

[22] M. Xu, C. Liu, Y. Zou, F. Zhao, J. Yu, and X. Cheng, "wchain: A fast fault-tolerant blockchain protocol for multihop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6915–6926, 2021.

[23] T. M. Tan and S. Saraniemi, "Trust in blockchain-enabled exchanges: Future directions in blockchain marketing," *Journal of the Academy of marketing Science*, vol. 51, no. 4, pp. 914–939, 2023.

[24] H. Xie, J. Zheng, T. He, S. Wei, and C. Hu, "Tebds: A trusted execution environment-and-blockchain-supported iot data sharing system," *Future Generation Computer Systems*, vol. 140, pp. 321–330, 2023.

[25] Z. Wang, Q. Chen, and L. Liu, "Permissioned blockchain-based secure and privacy-preserving data sharing protocol," *IEEE Internet of Things Journal*, 2023.

[26] R. Ma, L. Zhang, Q. Wu, Y. Mu, and F. Rezaeibagha, "Be-trdss: Blockchain-enabled secure and efficient traceable-revocable data-sharing scheme in industrial internet of things," *IEEE Transactions on Industrial Informatics*, 2023.

[27] S. Alshehri, O. Bamasaq, D. Alghazzawi, and A. Jamjoom, "Dynamic secure access control and data sharing through trusted delegation and revocation in a blockchain-enabled cloud-iot environment," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4239–4256, 2023.

[28] T. Li, H. Wang, D. He, and J. Yu, "Blockchain-based privacy-preserving and rewarding private data sharing for iot," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 15 138–15 149, 2022.

[29] C. Li, S. Liang, J. Zhang, Q.-e. Wang, and Y. Luo, "Blockchain-based data trading in edge-cloud computing environment," *Information Processing & Management*, vol. 59, no. 1, p. 102786, 2022.

[30] Y. Liu, X. Hao, W. Ren, R. Xiong, T. Zhu, K.-K. R. Choo, and G. Min, "A blockchain-based decentralized, fair and authenticated information sharing scheme in zero trust internet-of-things," *IEEE Transactions on Computers*, vol. 72, no. 2, pp. 501–512, 2022.

[31] J. Yu, B. Yan, H. Qi, S. Wang, and W. Cheng, "An efficient and secure data sharing scheme for edge-enabled iot," *IEEE Transactions on Computers*, pp. 1–14, 2023.

[32] T. Wu, W. Wang, C. Zhang, W. Zhang, L. Zhu, K. Gai, and H. Wang, "Blockchain-based anonymous data sharing with accountability for internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 5461–5475, 2023.

[33] C. Wang, S. Wang, X. Cheng, Y. He, K. Xiao, and S. Fan, "A privacy and efficiency-oriented data sharing mechanism for iots," *IEEE Transactions on Big Data*, vol. 9, no. 1, pp. 174–185, 2023.

[34] C. Zhang, T. Shen, and F. Bai, "Toward secure data sharing for the iot devices with limited resources: A smart contract-based quality-driven incentive mechanism," *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 12 012–12 024, 2023.

[35] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology—CRYPTO 2001: 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19–23, 2001 Proceedings*. Springer, 2001, pp. 213–229.

[36] S. Zhang, X. Ying, and B. Wang, "A privacy protection scheme based on linkable ring signature for user payment of peer-to-peer uniform-price double auction transaction in the microgrid day-ahead market," *International Journal of Electrical Power & Energy Systems*, vol. 147, p. 108806, 2023.

[37] B. Samra and S. Fouzi, "New efficient certificateless scheme-based conditional privacy preservation authentication for applications in vanet," *Vehicular Communications*, vol. 34, p. 100414, 2022.

[38] K.-A. Shim, "Security models for certificateless signature schemes revisited," *Information sciences*, vol. 296, pp. 315–321, 2015.

[39] J. Herranz and G. Sáez, "Forking lemmas for ring signature schemes," in *Progress in Cryptology-INDOCRYPT 2003: 4th International Conference on Cryptology in India, New Delhi, India, December 8-10, 2003. Proceedings 4*. Springer, 2003, pp. 266–279.

[40] A. De Caro and V. Iovino, "jpbc: Java pairing based cryptography," in *2011 IEEE symposium on computers and communications (ISCC)*. IEEE, 2011, pp. 850–855.

[41] Y. Zhao, Y. Yu, Y. Li, G. Han, and X. Du, "Machine learning based privacy-preserving fair data trading in big data market," *Information Sciences*, vol. 478, pp. 449–460, 2019.

[42] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen, "Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero," in *Computer Security–ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II 22*. Springer, 2017, pp. 456–474.

[43] 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 of Things Journal*, vol. 8, no. 24, pp. 17 236–17 260, 2021.