Received 2 September 2024; revised 31 October 2024; accepted 18 November 2024. Date of publication 21 November 2024; date of current version 4 December 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3504403

Eisdspa: An Efficient and Secure Blockchain-Based Donation Scheme With Privacy Protection and Auditability

YONG ZHOU [1], HONG LEI [1], AND ZIJIAN BAO [2]

[1] The School of Cyberspace Security (School of Cryptography), Hainan University, Haikou 570228, China
[2] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China

CORRESPONDING AUTHORS: H. LEI AND Z. BAO (e-mail: leiluono1@163.com; zibao@polyu.edu.hk)

This work was supported in part by the National Key R&D Program of China under Grant 2021YFB2700600, and in part by the Research Startup Fund of Hainan University under Grant KYQD(ZR)-21071.

ABSTRACT Charity donations are a critical mechanism for social resource distribution. However, traditional donation systems, typically centralized, are prone to issues such as data redundancy, vulnerability to single-point failures, and a deficiency in transparency and traceability. Although blockchain-based donation programs have emerged to address trust issues inherent in centralized models, they often neglect critical security concerns like privacy protection and identity authentication. This paper introduces Eisdspa, a blockchain-based donation system designed to offer identity authentication, auditability, and privacy protection. Specifically, we introduce an identity credential system that facilitates anonymous donations, shielding the identities of both donors and donees through the use of BBS+ signatures and zero-knowledge proofs of knowledge (ZKPoKs). Additionally, we ensure the integrity of goods donations by offering robust auditability and protecting user privacy with Pedersen commitments and ZKPoKs. We formally define the privacy aspects of Eisdspa and conduct a security analysis of the system under the random oracle model. A prototype implementation of the scheme, along with a comparative analysis with existing solutions, highlights the benefits of Eisdspa. Moreover, we assess the computational efficiency of Eisdspa, with experimental results indicating its high performance in computational overhead.

INDEX TERMS Auditability, authentication, donation system, privacy protection.

I. INTRODUCTION

CHARITY donations possess the potential to generate profound social impact and serve as a crucial conduit for the equitable distribution of societal resources [1]. The philanthropic landscape has witnessed remarkable expansion, underscoring the increasing necessity for effective charitable giving. Nevertheless, a range of pressing issues has surfaced in recent years. There exists a risk that certain charities may manipulate data during the donation process to further their own interests, raising significant concerns regarding integrity and accountability. Moreover, the opacity in the flow and utilization of goods complicates the ability to trace the trajectory from donation to intended beneficiaries, creating a disconnection between donors and recipients.

Traditional goods donation systems [2], [3], [4] often rely on centralized frameworks, typically hosted on cloud platforms, which are inherently susceptible to single-point failures. This reliance not only undermines reliability but also exacerbates issues of mismanagement and data manipulation, ultimately compromising public trust. Additionally, the donation process frequently lacks transparency, making it difficult to trace the flow and utilization of resources. In light of these challenges, it is imperative to explore enhancements to existing protocols that can improve transparency and accountability in philanthropic endeavors, thereby bolstering public confidence in charitable organizations. To mitigate these concerns, numerous blockchain-based solutions have been proposed. References [5], [6], [7] address the limitations of conventional charitable frameworks, promoting increased accountability and trust in charitable giving. Such solutions tackle critical issues, including the risk of information manipulation by charitable organizations and the opacity surrounding the allocation of donated goods. Notable examples include decentralized shipping platforms on the Ethereum network [5] and traceable donation systems that utilize Trusted Execution Environments (TEE) [6] and consortium blockchains [7]. However, these solutions do have some drawbacks, facing privacy and security challenges such as the risk of donation information leakage [8] and the exposure of the identities of donors and donees [9]. To address these concerns, some schemes incorporate cryptographic primitives to build privacy-protecting tokens [10] or credentials [11], [12]. While these cryptographic approaches provide noteworthy advantages, there are still unresolved issues that require further exploration and meticulous scrutiny.

Our motivations arise from an in-depth exploration of blockchain technology, BBS+ signatures and ZKPoKs in the framework of system architecture. Blockchain serves as a decentralized ledger that enhances transparency and reliability, making it pivotal for validating transactions and addressing trust issues across various applications [13], [14], [15], [16]. Its decentralized nature eliminates single points of failure, while immutable characteristics ensure that once data is recorded, it cannot be altered, safeguarding against fraud and manipulation. Additionally, blockchain facilitates seamless traceability, enabling stakeholders to track transactions and verify authenticity, which not only enhances accountability but also supports effective auditability, ensuring compliance and integrity throughout the transaction lifecycle. BBS+ signatures have been integrated into decentralized identity credential systems [17], [18], [19], particularly within anonymous credential frameworks for identity authentication. This integration allows users to selectively disclose their identity attributes, thereby enhancing privacy and providing individuals with greater control over their personal information. ZKPoKs [20], [21], [22] enhance user privacy by enabling one party to prove knowledge of information without revealing the information itself.

The aforementioned problems caused by the existing blockchain-based goods donation schemes and motivations induce three challenges. C1: Authentication. To mitigate fraudulent activities such as counterfeit donations and impersonation by participants, the solution must accurately verify the identities of all participants while facilitating anonymous donations to ensure equitable distribution of donated goods. C2: Privacy Protection. It is imperative to safeguard sensitive donation data, including the quantity of donations, the entitlement to receive them, and identity information, to prevent potential harm to individuals. However, a significant challenge arises in balancing privacy protection with the need for donors to access data regarding the distribution details of the projects in which they are involved. This tension underscores the complexities of ensuring both data confidentiality and appropriate transparency in the donation process. C3: Auditability. Auditors should rigorously evaluate and verify the donation and distribution processes to ensure that charitable organizations act with integrity, avoiding any ulterior motives, and to confirm that the number of donations recorded aligns with the actual quantity of goods distributed.

To address the three challenges previously outlined, we present a blockchain-based scheme termed Eisdspa for goods donation which incorporates identity authentication, privacy protection, and auditability. Our scheme provides user identity credentials utilizing the BBS+ signature to safeguard personally identifiable information. This approach ensures the legitimacy of participants while facilitating anonymous donations, effectively addressing challenge C1. Furthermore, our scheme constructs ZKPoKs to maintain the confidentiality of donation information throughout both the donation and distribution processes, thereby tackling challenge C2. This guarantees that sensitive data remains secure while allowing authorized auditors to verify the integrity of the donation process. Additionally, Eisdspa features a robust auditability mechanism to ensure the accuracy of the goods donation and distribution processes, thereby addressing challenge C3. The primary contributions of the proposed scheme are as follows:

- We propose and instantiate Eisdspa, a blockchain-based goods donation scheme centered on authentication, privacy protection, and auditability. Our work includes formal security models that encompass these critical components, and we provide rigorous proofs demonstrating that Eisdspa meets all security requirements defined in the formalized model.
- We design an identity authentication mechanism utilizing BBS+ signatures to enable anonymous donations, while leveraging ZKPoKs to ensure both privacy and auditability of the donation process.
- We conduct a series of experiments on the proposed Eisdspa system, comparing it with existing schemes in terms of overall performance, including computational and gas costs, as well as functional implementation. The results indicate that our system provides substantial advantages, delivering both high efficiency and comprehensive functionality.

The remainder of this paper proceeds as follows. In Section II, we review the related work concerning identity credentials, auditability, and donation schemes based on blockchain. In Section III, we provide a concise overview of preliminary. In Section IV, we introduce our system model, system components, and design objectives. In Sections V and VI, we describe the proposed Eisdspa and the details of ZKPoKs. In Section VII, we formally analyze the security of Eisdspa. In Section VIII, we demonstrate the implementation of Eisdspa, evaluate its efficiency and computational cost, and compare it with existing solutions. Finally, we conclude our work and potential future directions in Section IX.

II. RELATED WORK

A. CREDENTIAL BASED ON BLOCKCHAIN

As a fundamental cryptographic primitive, numerous credential schemes have emerged based on blockchain technology, especially with threshold-based credentials.

Sonnino et al. [11] proposed Coconut, a selective disclosure credential scheme that accommodates both public and private attributes, re-randomization, distributed threshold issuance, and multiple unlinkable disclosures of specific attributes. In contrast to conventional certificate systems, Coconut enables potential integration with blockchain-based frameworks, thereby enhancing both security and convenience. Li et al. [12] generated decentralized anonymous credentials (DACs) for each entity by applying threshold signatures [23] within a decentralized digital forensics framework. This approach allows data providers and investigators to authenticate identities prior to each transaction in digital forensics. The DAC generation process is decentralized and does not rely on a single trusted entity, thereby enhancing robustness and security within the decentralized digital forensics system. Doerner et al. [19] offered an anonymous credential scheme with threshold issuance based on the secure multiparty signature protocol of the BBS+ signature scheme. It demonstrated that composable security against malevolent attackers is possible when only signatures produced by a semi-honest protocol are verified.

While threshold-based credentials enhance system security, they encounter notable challenges, primarily due to their reliance on a fixed set of issuers during initial setup. This limitation requires a complete re-execution of the setup algorithm whenever an issuer is added or removed, restricting adaptability and complicating real-world scenarios. Eisdspa formalizes a system where users directly engage with an authority to request credentials, with the issuance process securely documented on a blockchain for verification, which mitigates the drawbacks of traditional threshold-based systems while preserving robust security.

B. BLOCKCHAIN-BASED AUDITING

Many papers have proposed auditability schemes that capitalize on the transparency and decentralization features of blockchain. By considering the potential threats posed by malevolent blockchain miners and auditors [15], [16], [24] utilize the decentralized nature of blockchain technology to distribute central responsibilities and ensure data integrity. However, the transparency inherent in blockchain also introduces risks of privacy breaches. Furthermore, the significant computational overhead required for processing and verifying large volumes of data in extensive audit tasks leads to inefficiencies, thereby limiting the applicability of these schemes primarily to small-scale audit scenarios.

Consequently, some works investigate blockchain auditing schemes that incorporate cryptographic techniques, such as zero-knowledge proofs and commitments. Zhang et al. [14] developed a blockchain-based sealed bid scheme, BSS, which utilizes a bid comparison circuit constructed from homomorphic encryption. This scheme provides anonymous and auditable verification of bid outcomes by integrating zero-knowledge proofs and commitments. The auctioneer can demonstrate to all bidders that the bid cipher of the winner corresponds to the plaintext of the generated cipher. Similarly, Chen et al. [25] proposed the auditable decentralized confidential payment system (ADCP) on the blockchain, which supports two levels of auditability: regulatory compliance and supervision. The former ensures that a given set of transactions adheres to regulatory policies through interactions with involved users, while the latter allows for inspection of any confidential transactions without user interaction, striking a balance between privacy and auditability. However, these auditing mechanisms often rely on overly complex zero-knowledge proofs, leading to significant computational overhead. To address this challenge, we have optimized the zero-knowledge proofs required for auditability, resulting in a substantial reduction in computational overhead for auditors while safeguarding sensitive information.

C. DONATION SYSTEM BASED ON BLOCKCHAIN

Blockchain offers significant advantages for charitable activities, leading to the development of various goods donation systems in numerous research studies. Kaur et al. [5] proposed an efficient architecture and algorithms for a user-centered solution using Ethereum to grant an access management policy to donors, donees, and charity organizations. This enhances the efficiency of the contribution process during disaster or pandemic events. To address trust issues, donors can access a report detailing their contribution history, which can be beneficial for communities affected by crises. Almaghrabi and Alhogail [26] proposed a blockchain-based donation traceability framework intended to enable all parties concerned to track the movement of donations for charities from the time they are provided by donors to the time they reach the intended donees. Although both [5] and [26] alleviate trust concerns, they leave sensitive information vulnerable, posing a significant risk of privacy breaches for donors and donees.

Li et al. [6] proposed a secure auditability donation system, Astraea, which integrated a distribute smart contract with the SGX Enclave [27] to distributed donations and prove the integrity of the donation number and donation sum while preserving donation privacy. Additionally, the system features a donation smart contract designed to facilitate deposit refunds and protect against theft and collusion attacks from malicious collectors and intermediaries. However, the audit mechanism employed in this framework suffers from inefficiency, requiring auditors to completely traverse all item data, thereby imposing an excessive computational burden.

Wang et al. [10] proposed a decentralized approach based on the utilization of tokens, systematically addressing the need for charitable distributions. Their solution includes two implementations–one for smart cards and another for smartphones–designed to meet the requirements of charitable organizations while ensuring robust accountability, scalability, and the privacy of donees. Kaur et al. [5] highlighted a system that leveraged smart contracts to automate resource and donation distribution and verification processes during emergencies. However, this system did not adequately address identity authentication and anonymity issues, raising privacy concerns for donors and donees and undermining trust in the automated processes. Saraswat et al. [28] introduced UpHaaR, a blockchain-based charity scheme that treated fund allocations and scheme data as ledger entries in the blockchain to mitigate fake fundraisers to protect the security and privacy of numerous users. However the deficiency of an identity authentication mechanism within the system compromises its credibility, rendering it unsuitable for widespread practical application due to diminished user trust. We propose a novel blockchain-based framework for goods donation, integrating for authentication, privacy preservation, and robust auditability capabilities employing BBS+ signatures for anonymous donations and authentication, and ZKPoKs to ensure both the privacy and auditability of Eisdspa.

III. PRELIMINARIES

A. ZERO-KNOWLEDGE PROOF-OF-KNOWLEDGE

A zero-knowledge proof of knowledge (ZKPoK) [29] is a two-party protocol between a prover and a verifier, which enables the prover to convince the verifier that it knows some secret piece of information without revealing anything about the secret apart from what is already disclosed in the claim [30]. For instance, $\mathsf{ZKPoK}\{(x) : y = g^x\}$ represents a ZKPoK of the integer $x \in \mathbb{Z}_p$ such that $y = g^x$ holds, where g is an element of the group $\mathbb{G} \;=\; < g >$. The norm is that the letter in the parenthesis (x) represents the quantity of which knowledge is being proven, while all other values are known to the verifier.

B. PEDERSEN COMMITMENT

Let $\mathbb{G}$ be a cyclic group with two generators g_0, g_1. A Pedersen commitment of an integer $a \in \{1, 2, \ldots, n\}$ is generated as follows: pick a random integer r and return the commitment $com = g_0^a g_1^r$ [31]. The Pedersen commitment is characterized by three features: ① Perfectly hiding: com reveals nothing about a; ② Additive homomorphism: $\prod_{i=0}^{n} com_i = g_0^{\sum_{i=0}^{n} a_i} g_1^{\sum_{i=0}^{n} r_i}$; ③ Computationally binding: a Pedersen commitment com cannot be opened to two different integers under the discrete logarithm assumption.

C. BILINEAR MAP

Let $\mathbb{G}_0$ and $\mathbb{G}_1$ be two multiplicative cyclic groups of prime order p, with g and h being their generators, respectively. Then the properties of the bilinear map [32] $e : \mathbb{G}_0 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ are as follows:

- Bilinear: $u \in \mathbb{G}_0$, $v \in \mathbb{G}_1$, and $a, b \in \mathbb{Z}_p$, then $e(g^u, g^v) = e(g, h)^{uv}$;
- Computability: for any $u \in \mathbb{G}_0$ and $v \in \mathbb{G}_1$, there exists an efficient algorithm to compute $e(u, v)$;
- Non-degenerate: there exist $u \in \mathbb{G}_0$, $v \in \mathbb{G}_1$, s.t. $e(u, v) \neq 1$, and $e(u, v)$ is the generator of $\mathbb{G}_T$.

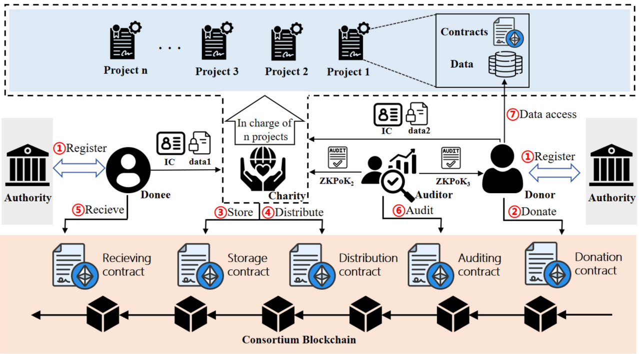

FIGURE 1. The system architecture of Eisdspa.

D. THE BBS+ SIGNATURE

The BBS+ signature [17] is a multi-message, secure digital signature mechanism which enables selective disclosure of any subset of the signed messages while supporting the proof of signature knowledge.

Let $g, g_0, g_1, \ldots, g_l$ be generators of $\mathbb{G}_0$, and h be the generator of $\mathbb{G}_1$. Choose $u \in \mathbb{Z}_p^*$ as the secret key of the signature scheme and compute the public key $Z = h^u$. A BBS+ signature on messages $m_1, m_2, \ldots, m_l$ is (t, a, s), where $a, s \in \mathbb{Z}_p^*$ and $t = (g g_0^s g_1^{m_1} \cdots g_l^{m_l})^{\frac{1}{u+a}}$. This signature can be verified as: $e(g g_0^s g_1^{m_1} \cdots g_l^{m_l}, h) = e(t, Zh^a)$. The BBS+ signature can be proved unforgeable against adaptively chosen message attacks under the q-SDH assumption [33], [34].

E. CONSORTIUM BLOCKCHAIN

Blockchain [35] is a distributed database system that links data chronologically in blocks following a consensus among a group of untrusted users. It is decentralized, unalterable, and immune to manipulation. A consortium chain is a blockchain that is collaboratively maintained and controlled by authorized organizations. Only qualified entities are permitted access to the chain. Compared to public blockchains, consortium blockchains provide superior permission management and privacy protection. Several well-known consortium blockchains are Hyperledger Fabric [36], R3 Corda [37], Fisco Bcos [38], etc.

IV. PROBLEM STATEMENT

A. SYSTEM MODEL

Eisdspa involves six entities: authority, donor, charity organization, donee, auditor, and the consortium blockchain. We display the architectural vision of Eisdspa in Fig. 1 and list the key notations in Table 1.

Authority (AU) serves as a trustworthy third party to issue the Identity Credential (IC) for authorized users and confirms the identity of users who must register with AU. At the stage of donating or receiving commodities, users utilize IC to authenticate themselves to CO.

Donor (DO) is a person who donates some goods to one of the projects of a charity organization so that they

TABLE 1. **Key notations of Eisdspa.**

Notation	Definition
AU, AO	Authority, Auditor
DO, DE	Donor, Donee
CO, PR	Charity organization, charity project
PPT	Probabilistic polynomial-time
num	The number of donation
$elig$	The number of receiving
λ	The security parameter
pk, sk	Public key, secret key
IC	Identity credential
$\mathcal{A}, \mathcal{C}, \Sigma$	Adversary, Challenger, the Eisdspa scheme
$negl(\lambda)$	The negligible function
mes, mes'	The basic information about goods donated or received.
com_{num}, res	The commitment and token of the number of goods donated by the donor.
com_{elig}, kes	The commitment and token of the number of goods received by the donee.
mul_{com}	The commitment of total number of goods in PR, which is the identity element of $\mathbb{G}_0$ initially.
mul_{tok}	The token of total number of goods in PR, which is the identity element of $\mathbb{G}_0$ initially.

can be managed and distributed. When DO delivers goods, he initiates a donation transaction tx_{do} to the consortium blockchain. To ensure transparency, we have additionally given DO special permissions to track the distribution progress of their donated projects.

Donee (DE) is the receiver of goods. They must first finish identity registration with AU in order to create their own identity credentials, just like DO. DE initiates a receiving transaction tx_{re} to acquire goods after completing the authentication process.

Charity Organization (CO) is responsible for the distribution and storage of goods. Each CO will manage a series of charity projects (PRs), and a project manager will be assigned to each PR. CO will store and upload the goods data after receiving the encrypted information from DOs and DEs. CO creates a commitment and a token of the amount donated and distributed for the auditor to validate the authenticity of the goods data. During this process, CO will begin executing a storage transaction tx_{st} and a distribution transaction tx_{di}.

Auditor (AO) is responsible for auditing the entire donation process. There are two tasks: ① affirming that the goods are correctly distributed and CO has not misappropriated the goods or tampered with the data; ② confirming that the donations made by DO match the amount received by CO.

Consortium Blockchain (CB) is responsible for recording and tracking the status information of goods, specifically the five statuses of donation, storage, distribution, receipt, and audit. Therefore, every charity project in the consortium blockchain has related smart contracts, which are used to record the status of goods in the project.

B. COMMUNICATION MODEL

This paper employs a communication model that primarily uses point-to-point channels as the default mechanism for secure information exchange. To address specific needs, we also incorporate broadcast channels for the transmission of commitments and zero-knowledge proofs, allowing for efficient communication to multiple recipients simultaneously. Additionally, authenticated communication channels are utilized to ensure data integrity and authenticity, with specific use cases outlined in the following sections. This model leverages established techniques for efficient implementation, enhancing both reliability and security in our framework.

C. SYSTEM COMPONENTS

The scheme is constructed as shown below for *Setup*, *Register*, *Donate*, *Store*, *Distribute&Receive*, *Audit*, and *Authorize*.

- *Setup Phase.* To set up the system, AU generates cryptographic parameters that are shared by all parties. Each entity generates a key pair (pk, sk). The algorithm satisfies the following syntax.
  - $params \leftarrow Setup(1^{\lambda})$: On input of the security parameter 1^{λ}, it returns the public parameter *params*. Additionally, donors, donees, auditors, authorities, charity organizations, and charity project managers get their own pair of keys.
- *Register Phase.* The user (DO or DE) completes identity registration through this phase, sending the identity information *material* to AU and getting an IC. Furthermore, AU compiles *material* of DE to determine the number *elig* for receiving goods and creates the relevant proof. The algorithms satisfy the following syntax.
  - $C \leftarrow RegisterC(s, m')$: The user inputs two random numbers s and m' and returns a commitment C.
  - $IC \leftarrow RegisterIC(C, material, a)$: On input of the commitment C, the identity information *material*, and a random number a, AU outputs a legitimate identity credential IC for the user.
  - $(elig, Ec, \sigma_{Ec}) \leftarrow AssigElig(material, w_1, w_2)$: On input of the identity information *material* of DE and two random numbers w_1 and w_2, AU outputs the receiving goods number *elig*, the proof Ec for *elig* and the relevant signature σ_{Ec}. In addition AU sends *elig*, Ec and σ_{Ec} to DE through the encrypted channel.
- *Donate Phase.* DO verifies the legitimacy of IC to CO. If validation is successful, DO will make a donation. The algorithm satisfies the following syntax.
  - $(com_{num}, res) \leftarrow Donate(num, r)$: DO inputs the number num of goods and the random number r. It outputs a commitment com_{num} and a token res. Additionally, DO sends num and a random number

r' to CO through the encrypted channel and initiates the donation transaction tx_{do}.

- *Store Phase.* At this stage, DO gives the goods to CO for safekeeping. After receiving num and r' from DO, CO will initiate the storage transaction tx_{st} after receiving the goods and generate a proof for pk_{DO} so that DO may display the donation motion to the charity project in the latter authorize phase. The related storage algorithms are as follows.
  - $(mul_{com}, mul_{tok}, com'_{num}, res') \leftarrow Store(num, r', mul_{com}, mul_{tok})$: CO inputs num, r', the commitment mul_{com} and the token mul_{tok} for the total number of goods. It outputs the updated commitment mul_{com} and the updated token mul_{tok} for the current total number of goods in PR. In addition, it outputs a new commitment com'_{num} and a new token res' for the number of this donation.
  - $(Ac, \sigma_{Ac}) \leftarrow ProofGen(v, pk_{DO}, pk_{PR})$: CO inputs a random number $v \in \mathbb{Z}_p^*$, pk_{DO} and pk_{PR}. It outputs the proof Ac of pk_{DO} and the signature σ_{Ac} of CO on Ac.
- *Distribute & Receive Phase.* At this stage, CO distributes the goods to DE who receives them. It is necessary to confirm the legitimacy of IC when DE seeks permission to obtain goods from CO. Following verification, DE sends Ec, σ_{Ec} and k' to CO through the encrypted channel, where k' is a random number. The distribution algorithm and the receiving algorithm are as follows.
  - $(mul_{com}, mul_{tok}, com'_{elig}, kes') \leftarrow Distribute(Ec, \sigma_{Ec}, k', mul_{com}, mul_{tok})$: On inputs the proof Ec of receiving number $elig$, the signature σ_{Ec}, k', the commitment mul_{com} and token mul_{tok} for the total number of goods, it outputs the updated commitment mul_{com} and the updated token mul_{tok} for the current total number of goods of PR, a commitment com'_{elig} and a token kes' for the number of receiving $elig$.
  - $(com_{elig}, kes) \leftarrow Receive(elig, k)$: DE inputs $elig$ and a random number k. It outputs a new commitment com_{elig} and a new token kes for $elig$.
- *Audit Phase.* There are two cases in this phase: either an audit of a donation or an audit of the total amount of goods in PR.
  - $result_1 \leftarrow AuditDO(com_{num}, com'_{num})$: AO inputs commitments of the number of donations com_{num}, com'_{num} generated during the donate phase and the store phase. If check passes, it returns $result_1 = 1$, and $result_1 = 0$ otherwise.
  - $result_2 \leftarrow AuditPR(mul_{com}, mul_{tok})$: AO inputs the commitment mul_{com} and the token mul_{tok} of the total number. If check passes, it returns $result_2 = 1$, and $result_2 = 0$ otherwise.
- *Authorize Phase.* The scheme provides a privacy dataset for each PR that contains the distribution details of the goods contributed by DOs, enabling DOs to be informed of the distribution details. Nevertheless, only DOs contributing to this project can access the data.
  - $\{0, 1\} \leftarrow AuthorizeProcess(Ac, \sigma_{Ac}, pk_{DO})$: The PR manager inputs the proof Ac, the signature σ_{Ac}, and the public key pk_{DO} to confirm pk_{DO} is legitimate. If the check passes, it returns 1 and authorizes DO to access privacy datasets, and 0 otherwise.

D. DESIGN OBJECTIVES

We formalize the security requirements and formal definitions of security models for Eisdspa. The following is an introduction to the security objectives.

- Authentication. In order to confirm the authenticity of identities, DO and DE must perform identity authentication prior to making donations or receiving commodities.
- Privacy Protection. There are two primary aspects. (1) Amount protection: Both the amount of goods provided by DO and the amount of goods obtained by DE will be kept secret. (2) Anonymity: Even when contributing or receiving goods, the identities of DO and DE will remain confidential.
- Auditability. We stipulate that CO must not falsify its donation and start fraudulent donations, nor may it misappropriate part of the products for personal benefit and tamper with the donation information. It is necessary to identify any missing parts from the commodities. This means that the number of goods provided by DO minus the amount obtained by DE equals the number of goods accessible to CO.

We provide formal definitions of security models with respect to the aforementioned requirements by constructing the cryptographic game. Considering $\mathcal{A}$ as an adversary assaulting Eisdspa, we can represent attack behaviors as adversarial queries to oracles that are implemented by a challenger $\mathcal{C}$. The oracles that are accessible to $\mathcal{A}$ are listed as follows.

- $\mathcal{O}_{sign}$: $\mathcal{A}$ queries this oracle with an identity credential IC and a message m to get a signature. $\mathcal{C}$ keeps track of this type of query by maintaining a set $\mathcal{S}_{\sigma,m}$, which is initially empty. Upon receiving a fresh query, if $(\cdot, m) \in \mathbb{S}_{\{\sigma,m\}}$, it returns $\perp$. Otherwise it returns a fresh valid signature σ and stores (m, σ) in $\mathbb{S}_{\{\sigma,m\}}$.
- $\mathcal{O}_{donate}$: $\mathcal{A}$ queries this oracle with an identity credential IC to donate some goods and conduct a donation transaction. Upon receiving a fresh query, $\mathcal{C}$ returns the donation number num and a donation transaction.
- $\mathcal{O}_{distribute}$: $\mathcal{A}$ queries this oracle with an identity credential IC to distribute some goods and conduct a distribution transaction. $\mathcal{C}$ keeps track of this type of query by maintaining a set $\mathcal{S}_{IC}$, which is initially empty. On receiving a fresh query, $\mathcal{C}$ checks if $IC \in \mathcal{S}_{IC}$. If not, $\mathcal{C}$ returns $elig$ and a distribution transaction and stores IC in the set $\mathcal{S}_{IC}$. Else, $\mathcal{C}$ returns $\perp$.

We define the advantage of $\mathcal{A}$ in $\Sigma = (Setup, Register, Donate, Store, Distribute\&Receive, Audit, Authorize)$ to win in PPT when engaging with a game as below.

Definition 1 (Authentication): Let $IC_{DO} = (IC_{DO1}, \ldots, IC_{DO}n)$ be the set of all donors, and $IC_{DE} = (IC_{DE1}, \ldots, IC_{DE}n)$ be the set of all donees. Σ achieves authentication if the PPT adversary $\mathcal{A}$ has negligible advantage in the following game.

Authentication$(\Sigma, \lambda, \mathcal{A})$:
$params \leftarrow Setup(1^{\lambda})$
$(m_i, \sigma_i)_{i=1}^{p} \leftarrow \mathcal{A}^{\mathcal{O}_{Sign}}(IC_{DO}, IC_{DE}, params)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}((m_i, \sigma_i)_{i=1}^{p}, params)$

On the jth inquiry $\mathcal{O}_{Sign}$, if $(\cdot, m_j) \in \mathbb{S}_{\{\sigma,m\}}$, it returns $\perp$. Otherwise it returns a fresh valid signature σ_j and stores (m_j, σ_j) in $\mathbb{S}_{\{\sigma,m\}}$. If $(\cdot, m) \notin \mathbb{S}_{\{\sigma,m\}}$ and $Verify(m, \sigma) = 1$, $\mathcal{A}$ wins in the game. The advantage of $\mathcal{A}$ in winning the game can be defined as $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Authentication}$.

Definition 2.1 (Amount protection): Σ has the ability to conceal the number of donations if the PPT adversary $\mathcal{A}$ has negligible advantage in the following $Game_{Num}$.

Privacy_num$(\Sigma, \lambda, \mathcal{A})$:
$params \leftarrow Setup(1^{\lambda})$
$num_0, num_1 \leftarrow \mathcal{A}(params)$; $b \leftarrow \{0, 1\}$
$com_{num_b} \leftarrow Donate(num_b, r)$
$b' \leftarrow \mathcal{A}(com_{num_b}, num_0, num_1)$

$\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Privacy\_num}$ is the advantage of $\mathcal{A}$ in winning the game, then:

$$\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Privacy\_num} = \left|\mathsf{Pr}\left[\mathcal{A}\ wins - \frac{1}{2}\right]\right| = \left|\mathsf{Pr}\left[b' = b\right] - \frac{1}{2}\right|.$$

Similarly, the advantage of $\mathcal{A}$ in the game of guessing the number of receiving can be defined as $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Privacy\_elig}$, and

$$\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Privacy\_elig} = \left|\mathsf{Pr}\left[\mathcal{A}\ wins - \frac{1}{2}\right]\right| = \left|\mathsf{Pr}\left[b' = b\right] - \frac{1}{2}\right|.$$

If the PPT adversary $\mathcal{A}$ has negligible advantage in the game, Σ has the ability to conceal the number of receiving.

Definition 2.2 (Anonymity): Let $num = \{num_1, \ldots, num_n\}$ be the set of all donation numbers. Σ is anonymous for DO if the PPT adversary $\mathcal{A}$ has negligible advantage in the following $Game_{Anoy}$.

Anonymity$(\Sigma, \lambda, \mathcal{A})$:
$params \leftarrow Setup(1^{\lambda})$
$IC_1, \ldots, IC_n \leftarrow \mathcal{A}(params)$
$b \leftarrow \{1, \ldots, n\}$, $tx_b \leftarrow Donate(IC_b, num_b)$
$b' \in \{1, \ldots, n\}$, $b' \leftarrow \mathcal{A}(tx_b)$

$\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Anonymity\_DO}$ is the advantage of $\mathcal{A}$ in winning the game, then:

$$\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Anonymity\_DO} = \left|\mathsf{Pr}\left[\mathcal{A}\ wins - \frac{1}{n}\right]\right| = \left|\mathsf{Pr}\left[b' = b\right] - \frac{1}{n}\right|.$$

Similarly, the advantage of $\mathcal{A}$ in the game of guessing the identity of DE can be defined as $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Anonymity\_DE}$ as shown below. If the PPT adversary $\mathcal{A}$ has negligible advantage in the game, Σ is anonymous for DE.

$$\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Anonymity\_DE} = \left|\mathsf{Pr}\left[\mathcal{A}\ wins - \frac{1}{n}\right]\right| = \left|\mathsf{Pr}\left[b' = b\right] - \frac{1}{n}\right|$$

Definition 3 (Auditability): Σ is auditable if the PPT adversary $\mathcal{A}$ has negligible advantage in the following $Game_{Audit}$.

Audit$(\Sigma, \lambda, \mathcal{A})$:
$params \leftarrow Setup(1^{\lambda})$
For $i = 1$ to n: $IC_i \leftarrow RegisterIC(material_i, a_i, s_i)$
$(num_1, \ldots, num_d) \leftarrow \mathcal{A}^{\mathcal{O}_{donation}}(IC_1, \ldots, IC_n)$
$(elig_1, \ldots, elig_t) \leftarrow \mathcal{A}^{\mathcal{O}_{distribution}}(IC_1, \ldots, IC_n)$

$\mathcal{A}$ queries d times donation oracle and t times distribution oracle. $\mathcal{C}$ keeps track of distribution queries by maintaining a set $\mathbb{S}_{IC}$, which is initially empty. On the jth inquiry $\mathcal{O}$, if $IC_j \in \mathbb{S}_{IC}$, it returns $\perp$. Otherwise, it returns $elig_j$ and stores IC_j in the set $\mathbb{S}_{IC}$. $\mathcal{A}$ wins the game if $sum < (\sum_{i=1}^{d} num_i - \sum_{j=1}^{t} elig_j)$ where sum is the remaining goods number of $\mathcal{C}$.

V. PROPOSED SCHEME

In this section, we present the instantiation of Eisdspa and the details.

A. AN EFFICIENCY INSTANTIATION

- *Setup Phase.*
  - $params \leftarrow Setup(1^{\lambda})$: AU takes as input the security parameter 1^{λ} and generates three groups $\mathbb{G}_0$, $\mathbb{G}_1$ and $\mathbb{G}_T$ of prime order p. g, g_0, g_1 are generators of $\mathbb{G}_0$ and h is the generator of $\mathbb{G}_1$. Additionally, it chooses two hash functions. $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ and $H_2 : \{0,1\}^* \rightarrow \{0,1\}^l$. Each entity gets a public-private key pair (pk, sk) where $sk \in \mathbb{Z}_p$ and $pk = h^{sk}$. It returns the public parameters $params = \{e, \mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T, g, g_0, g_1, h, p, H_1, H_2, pk\}$.
- *Register Phase.*
  - $C \leftarrow RegisterC(s, m')$: The user chooses two random numbers $s, m' \in \mathbb{Z}_p^*$ and computes a commitment $C = g_0^s g_1^{m'}$. Then he sends his identity information *material* and C to AU through the encrypted channel, along with the following zero-knowledge proof.

$$\mathsf{ZKPoK}_0\left\{\left(s, m'\right) : C = g_0^s g_1^{m'}\right\} \tag{1}$$

  - $IC \leftarrow RegisterIC(C, material, a)$: AU gets (*material*, C) based on the encrypted channel, then verifies the authenticity of *material* and the verification of ZKPoK_0. AU returns failure if the verification of *material* and ZKPoK_0 outputs invalid. Otherwise AU chooses $a \in \mathbb{Z}_p^*$ and computes $m'' = H_1(material||a)$ and $t = (gCg_1^{m''})^{\frac{1}{a+u}}$, where u is the private key of AU and the public key $Z = h^u$. AU sends IC to the user through the encrypted channel, where $IC =$

  (t, a). After obtaining the IC, the user calculates $m = m' + H_1(material||a)$ and verifies $e(t, Zh^a) \overset{?}{=} e(gg_0^s g_1^m, h)$. If it succeeds, the user stores IC as the identity credential.
  - $(elig, Ec, \sigma_{Ec}) \leftarrow AssigElig(material, w_1, w_2)$: AU combines the identifying information *material* with the local village committee to examine the actual circumstances about DE and gives him a qualifying number *elig* for the quantity of goods he gets. Additionally, AU chooses two random numbers w_1, $w_2 \in \mathbb{Z}_p^*$, computes $Ec_{1a} = h^{w_1}, Ec_{1b} = pk_{DE} \cdot pk_{CO}^{w_1}$, $Ec_{2a} = h^{w_2}$, $Ec_{2b} = elig \cdot pk_{CO}^{w_2}$, and generates a signature σ_{Ec} on the proof $Ec = (Ec_{1a}, Ec_{1b}, Ec_{2a}, Ec_{2b})$ to CO. AU sends $(elig, Ec, \sigma_{Ec})$ to DE through the encrypted channel.
- *Donate Phase.* Following the register phase, after obtaining the IC, DO proves the validity of IC based on ZKPoK_1. If the verification of ZKPoK_1 outputs invalid, DO fails to make donations. Otherwise DO executes the following algorithm and launches the donation transaction tx_{do}.

$$\mathsf{ZKPoK}_1\big\{(t, a, s, m) : e\big(t, Zh^a\big) = e\big(gg_0^s g_1^m, h\big)\big\} \quad (2)$$

  - $(com_{num}, res) \leftarrow Donate(num, r)$: DO randomly chooses $r \in \mathbb{Z}_p^*$ and computes the commitment $com_{num} = g_0^{num} g_1^r$ and a token $res = g_0^r$ for secure auditing. Additionally, DO chooses another random number $r' \in \mathbb{Z}_p^*$ and sends *num* and r' to CO through the encrypted channel. DO launches a donation transaction $tx_{do} = (mes, com_{num}, res)$, where *mes* is the fundamental details about the goods and $mes = (id, H_2(list), time, pk_{PR})$. Here *id* is the serial number of the donation, *list* is the detailed inventory of the goods, *time* is the timestamp of the donation, and pk_{PR} is the public key of the charity project.
- *Store Phase.*
  - $(mul_{com}, mul_{tok}, com'_{num}, res') \leftarrow Store(num, r', mul_{com}, mul_{tok})$: After getting *num*, r' from DO, CO computes a new commitment $com'_{num} = g_0^{num} g_1^{r'}$ and a new token $res' = g_0^{r'}$ for the donation number. Additionally, CO updates the commitment and token for the current total number of goods of PR by processing $mul_{com} = mul_{com} \cdot com'_{num}$ and $mul_{tok} = mul_{tok} \cdot res'$, where they both are the identity element of $\mathbb{G}_0$ initially.
  - $(Ac, \sigma_{Ac}) \leftarrow ProofGen(v, pk_{DO}, pk_{PR})$: CO randomly chooses $v \in \mathbb{Z}_p^*$ and computes $Ac_1 = h^v$, $Ac_2 = pk_{DO} \cdot (pk_{PR})^v$ by which PR authorizes DO for private database access in the authorize phase. Then CO signs $Ac = (Ac_1, Ac_2)$ and gets the signature σ_{Ac}. CO launches the storage transaction $tx_{st} = (mes, com'_{num}, res', mul_{com}, \sigma_{Ac}, mul_{tok}, Ac)$.
- *Distribute & Receive Phase.* Before receiving the goods from CO, DE proves the validity of IC based on ZKPoK_1. If the verification of ZKPoK_1 outputs invalid, DE fails to get goods. Otherwise DE sends (Ec, σ_{Ec}, k') to CO by encrypted channel where k' is randomly chosen by DE, and the following algorithms will be executed.
  - $(mul_{com}, mul_{tok}, com'_{elig}, kes') \leftarrow Distribute (Ec, \sigma_{Ec}, k', mul_{com}, mul_{tok})$: CO firstly verifies the authenticity of Ec through σ_{Ec}. If it is true, CO computes $pk_{DE} = \frac{Ec_{1b}}{(Ec_{1a})^{sk_{CO}}}$ to identify the right DE and $elig = \frac{Ec_{2b}}{(Ec_{2a})^{sk_{CO}}}$ to get the receiving number *elig* of the DE. Then CO computes a commitment $com'_{elig} = g_0^{elig} g_1^{k'}$ and a token $kes' = g_0^{k'}$ for the number. Additionally, CO updates the commitment and token for the current total number of goods in PR by processing $mul_{com} = \frac{mul_{com}}{com'_{elig}}$ and $mul_{tok} = \frac{mul_{tok}}{kes'}$. CO launches the distribution transaction $tx_{di} = (mes', com'_{elig}, kes')$, where mes' is the basic information on the goods received and is similar to *mes*, which is in the donate phase.
  - $(com_{elig}, kes) \leftarrow Receive(elig, k)$: DE randomly chooses $k \in \mathbb{Z}_p^*$. Then DE computes a new commitment $com_{elig} = g_0^{elig} g_1^k$ and a new token $kes = g_0^k$ of receiving number *elig* and launches the receiving transaction $tx_{re} = (mes', com_{elig}, kes)$.
- *Audit Phase.*
  - $result_1 \leftarrow AuditDO(com_{num}, com'_{num})$: When AO submits an audit to a donation of DO, ZKPoK_2 is executed. If verification passes, it returns $result_1 = 1$, otherwise it returns $result_1 = 0$. Additionally, AO launches an audit transaction $tx_{au_2} = (pk_{DO}, result_1, time)$.

$$\mathsf{ZKPoK}_2\left\{\begin{array}{l} \big(num, r, r'\big) : com_{num} = g_0^{num} g_1^r \\ \wedge\ com'_{num} = g_0^{num} g_1^{r'} \end{array}\right\} \quad (3)$$

  - $result_2 \leftarrow AuditPR(mul_{com}, mul_{tok})$: When AO submits an audit to a project PR, ZKPoK_3 is executed, where *sum* is the total number of goods that are remaining in the PR. x is related to the random numbers r' and k' sent by the donors and donees during all storage phases and distribution phases. Assuming there are t_1 storages and t_2 distributions, then $x = \sum_{i=1}^{t_1} r'_i - \sum_{j=1}^{t_2} k'_j$, which is secret for AO. If verification passes it returns $result_2 = 1$, otherwise $result_2 = 0$. Lastly, AO launches the auditing transaction $tx_{au_1} = (pk_{PR}, result_2, time)$.

$$\mathsf{ZKPoK}_3\left\{\begin{array}{l} (sum, x) : mul_{com} = g_0^{sum} g_1^x \\ \wedge\ mul_{tok} = g_0^x \end{array}\right\} \quad (4)$$

- *Authorize Phase.*
  - $\{0, 1\} \leftarrow AuthorizeProcess(Ac, \sigma_{Ac}, pk_{DO})$: The PR manager in charge verifies the authenticity of Ac through the signature σ_{Ac}. If it is true, PR confirms

if $\frac{Ac_2}{(Ac_1)^{sk_{PR}}} = pk_{DO}$. Returns 1 and authorizes the DO to access the privacy datasets of the PR if it holds, and 0 otherwise.

VI. DETAILS OF THE ZERO-KNOWLEDGE PROOF-OF-KNOWLEDGE PROTOCOLS

A. DETAILS OF ZKPOK$_0$

ZKPoK$_0$ can be done using standard proof of representation of discrete logarithms.

- (Commitment) The user generates $r_s, r_{m'} \in \mathbb{Z}_p^*$ randomly, computes $T = g_0^{r_s} g_1^{r_{m'}}$ and sends T to AU.
- (Challenge) AU chooses a random challenge $c \in \mathbb{Z}_p^*$ and sends c to the user.
- (Response) The user computes $z_{r_s} = r_s - cs$, $z_{r_{m'}} = r_{m'} - cm'$ and sends $(z_{r_s}, z_{r_{m'}})$ to AU.
- (Verification) AU outputs 1 if $T = C^c g_0^{z_{rs}} g_1^{z_{r_{m'}}}$ and 0 otherwise.

B. DETAILS OF ZKPOK$_1$

To conduct ZKPoK$_1$, the prover first computes $T = tg^{r_1}, R = g_1^{r_1} g^{r_2}$ for some randomly generated $r_1, r_2 \in \mathbb{Z}_p^*$. Then he conducts the following equations.

$$\begin{aligned} e\big(t, Zh^a\big)e\big(g^{r_1}, Zh^a\big) &= e\big(gg_0^s g_1^m, h\big)e\big(g^{r_1}, Zh^a\big) \\ &= e\big(tg^{r_1}, Zh^a\big) \\ &= e\big(T, Zh^a\big) \end{aligned}$$

So it can be deduced that

$$\frac{e(T, Z)}{e(g, h)} = e(g_0, h)^s e(g_1, h)^m e(g, Z)^{r_1} e(g, h)^{ar_1} e(T, h)^{-a}.$$

And ZKPoK$_1$ is equivalent to ZKPoK$'_1$ as follows.

$$\text{ZKPoK}'_1 \left\{ \begin{array}{l} (r_1, r_2, t, a, s, m, \ \ t_1, t_2) : \\ \frac{e(T,Z)}{e(g,h)} = e(g_0, h)^s e(g_1, h)^m \\ e(g, Z)^{r_1} e(g, h)^{ar_1} \\ e(T, h)^{-a} \\ \wedge\ R = g_1^{r_1} g^{r_2} \end{array} \right\} \tag{5}$$

Then he conducts the following protocol with verifier.

- (Commitment) The prover randomly generates $\rho_s, \rho_m, \rho_{r_1}, \rho_{r_2}, \rho_{ar_1}, \rho_{ar_2}, \ \rho_a \in \mathbb{Z}_p^*$, computes $T_1 = e(g_0, h)^{\rho_s} e(g_1, h)^{\rho_m} e(g, Z)^{\rho_{r_1}} e(g, h)^{\rho_{ar_1}} e(T, h)^{-\rho_a}$ and $T_2 = g_1^{\rho_{r_1}} g^{\rho_{r_2}}$. Then prover sends (T_1, T_2) to verifier.
- (Challenge) The verifier chooses a random challenge $c \in \mathbb{Z}_p^*$ and sends c to the prover.
- (Response) The prover computes $z_s = \rho_s - cs, z_m = \rho_m - cm, z_a = \rho_a - ca, z_{r_1} = \rho_{r_1} - cr_1, z_{r_2} = \rho_{r_2} - cr_2, z_{ar_1} = \rho_{ar_1} - car_1, z_{ar_2} = \rho_{ar_2} - car_2$ and sends $(z_s, z_m, z_a, z_{r_1}, z_{r_2}, z_{ar_1}, z_{ar_2})$ to the verifier.
- (Verification) The verifier outputs 1 If $T_1 = e(T, Z)^c e(g, h)^{-c} e(g_0, h)^{z_s} e(g_1, h)^{z_m} e(g, Z)^{z_{r_1}} e(g, h)^{z_{ar_1}} e(T, h)^{-z_a}$ and $T_2 = R^c g_1^{z_{r_1}} g^{z_{r_2}}$, and 0 otherwise.

C. DETAILS OF ZKPOK$_2$

ZKPoK$_2$ can be done using standard proof of representation of discrete logarithms together with equality of discrete logarithms.

- (Commitment) The prover generates $r_1, r_2, r_3 \in \mathbb{Z}_p^*$, computes $T_1 = g_0^{r_1} g_1^{r_2}, T_2 = g_0^{r_1} g_1^{r_3}$ and sends (T_1, T_2) to the verifier.
- (Challenge) The verifier chooses a random challenge $c \in \mathbb{Z}_p^*$ and sends c to the prover.
- (Response) The prover computes $z_{r_1} = r_1 - cnum, z_{r_2} = r_2 - cr, z_{r_3} = r_3 - cr'$ and sends $(z_{r_1}, z_{r_2}, z_{r_3})$ to verifier.
- (Verification) The verifier outputs 1 if $T_1 = (com_{num})^c g_0^{z_{r1}} g_1^{z_{r2}}$ and $T_2 = (com'_{num})^c g_0^{z_{r1}} g_1^{z_{r3}}$, and 0 otherwise.

D. DETAILS OF ZKPOK$_3$

ZKPoK$_3$ can be done using standard proof of representation of discrete logarithms together with equality of discrete logarithms.

- (Commitment) The prover randomly generates $r_1, r_2 \in \mathbb{Z}_p^*$, computes $T_1 = g_0^{r_1} g_1^{r_2}, T_2 = g_0^{r_2}$ and sends (T_1, T_2) to the verifier.
- (Challenge) The verifier chooses a random challenge $c \in \mathbb{Z}_p^*$ and sends c to the prover.
- (Response) The prover computes $z_{r_1} = r_1 - csum, z_{r_2} = r_2 - cx$ and sends (z_{r_1}, z_{r_2}) to the verifier.
- (Verification) The verifier outputs 1 if $T_1 = (mul_{com})^c g_0^{z_{r1}} g_1^{z_{r2}}$, $T_2 = (mul_{tok})^c g_0^{z_{r2}}$ and 0 otherwise.

VII. SECURITY PROOFS

Theorem 1. Eisdspa provides identity authentication under the soundness of the ZKPoK and the q-SDH assumption.

Proof: $t = (gg_0^s g_1^m)^{\frac{1}{a+u}}$ in IC. During the authentication process, if and only if $\mathcal{A}$ is successful in forging a legitimate identity credential $IC^* = (t^*, a)$, a valid signature σ^* and a validated authenticity by ZKPoK$_1$ can be generated. The structure of IC is predicated on the BBS+ signature, which is unforgeable against adaptive chosen message attack under the q-SDH assumption, so $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Authentication} \leq negl(\lambda)$, completing the proof. ■

Theorem 2. Σ has the ability to conceal the number of donations and receiving under the Diffie Hellman (DH) assumption.

Proof: Given $(\mathbb{G}, g^r)$, the advantage of computing r is negligible in DH assumption. That is $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{DH-r} \leq negl(\lambda)$. $\mathcal{A}$ generates two donation numbers num_0 and num_1 in $Game_{Num}$ and sends them to $\mathcal{C}$. Given (g_0, g_1), $\mathcal{C}$ can generate two commitments $(com_{num_0}, com_{num_1})$ and two tokens (res_0, res_1). We hide the donation number in (com_{num_0}, res_0) and (com_{num_1}, res_1). Consequently, $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Privacy\_num} \leq \mathsf{Adv}_{\Sigma,\mathcal{A}}^{DH-r}$. So $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Privacy\_num}$ is negligible under DH assumption. Similarly, $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Privacy\_elig} \leq negl(\lambda)$, completing the proof. ■

Theorem 3. Eisdspa is anonymous under the q-Strong Diffie-Hellman (q-SDH) assumption.

TABLE 2. Computational costs (ms).

Scheme		DO	CO	DE	AO
Astraea [6]	theory	$T_{hash} + 3T_{mul} + T_{add} + T_{sig}$	$2T_{hash} + (3n+1)T_{sig} + 2nT_{dec} + (3n+3)T_{mul} + (5n-2)T_{add}$	$T_{sig} + T_{enc} + T_{hash}$	$(2n+4)T_{mul} + (2n+2)T_{add} + 2T_{hash}$
	practice	0.442	162.7n+161.2	0.097	0.24n+0.4656
BBD [5]	theory	$T_{register} + T_{create}$	$T_{access} + T_{approve}$	$T_{register} + T_{finalize}$	✗
	practice	119	134n	93	✗
UpHaaR [28]	theory	$T_{signcryption}$	$nT_{unsigncryption} + nT_{keygen}$	$T_{verify} + T_{unsigncryption}$	$nT_{keygen} + nT_{sc}$
	practice	43.64	41.56n	83.44	6.56n
Ours	theory	$T_{1p} + T_{hash} + 3T_{mul} + T_{add}$	$2nT_{1v} + n(12T_{mul} + 3T_{add} + 2T_{hash} + T_{sig})$	$T_{1p} + 2T_{mul} + T_{add} + T_{hash}$	$T_{2v} + T_{3v}$
	practice	20.59	56.72n	19.39	10.23

T_{hash}: the time of a hash; T_{mul}: the time of a scale multiplication in $\mathbb{G}$; T_{add}: the time of a point add in $\mathbb{G}$; T_{sig}: the time of signing; T_{enc}: the time of encryption; T_{1p}: the prove time of ZKPoK_1; T_{1v}: the verify time of ZKPoK_1; T_{2v}: the verify time of ZKPoK_2; T_{3v}: the verify time of ZKPoK_3; n: the number of DO and DE; $T_{signcryption}$: the time of a signcryption; $T_{unsigncryption}$: the time of an unsigncryption; $T_{unsigncryption}$: the time of an unsigncryption; T_{keygen}: the time of a session key establishment; T_{verify}: the time of a session key verification; T_{sc}: the time of executing the smart contract for UpHaaR; $T_{register}$: the time of user registration; T_{create}: the time of donation creation; T_{access}: the time of giving access; $T_{approve}$: the time of approving request; $T_{finalize}$: the time of finalizing request.

Proof: Given the elements $\{g, g^a, g^{a^2}, \ldots, g^{a^q}\}$ in $\mathbb{G}_0$ and $\{h, h^a\}$ in $\mathbb{G}_1$, it is hard to calculate $g^{\frac{1}{a+e}}$ where $e \in \mathbb{Z}_p^*$ under q-SDH assumption. That is $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{q-SDH} \leq negl(\lambda)$. We define two events. ① Event1: $\mathcal{A}$ successfully guessed b by breaking the q-SDH assumption in $Game_{Anoy}$. ② Event2: $\mathcal{A}$ successfully guessed b randomly in $Game_{Anoy}$. We have

$$\begin{aligned}\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Anonymity\_DO} &= \left|\Pr[\mathcal{A}\ wins] - \frac{1}{n}\right| \\ &= \left|\Pr\left[b' = b\right] - \frac{1}{n}\right| \\ &= \left|\Pr[\text{Event1}] + \Pr[\text{Event2}] - \frac{1}{n}\right| \\ &= \left|\mathsf{Adv}_{\Sigma,\mathcal{A}}^{q-SDH} + \frac{1}{n} - \frac{1}{n}\right| \leq negl(\lambda).\end{aligned}$$

As demonstrated by the same premise, $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{Anonymity\_DE} \leq negl(\lambda)$, completing the proof. ■

Theorem 4. Eisdspa is auditable if the Elliptic Curve Discrete Logarithm Problem (ECDLP) assumption holds.

Proof: We assume that the malicious parties cannot initiate a receiving transaction tx_{re} on behalf of a donee in Eisdspa. $\mathcal{B}$ is a PPT adversary towards ECDLP and an elementary wrapper from $\mathcal{A}$. Now we assume that $\mathcal{A}$ has non-negligible advantage in $Game_{Audit}$ and show how to construct $\mathcal{B}$ breaking the ECDLP assumption with non-negligible advantage.

1. $\mathcal{B}$ calls $\mathcal{A}$ as a subroutine;
2. $\mathcal{A}$ initiates some queries to $\mathcal{O}_{donation}$ and $\mathcal{O}_{distribution}$;
3. $\mathcal{B}$ outputs what $\mathcal{A}$ outputs.

Given that $\mathcal{A}$ runs in polynomial time, the outcome is performed in polynomial time by $\mathcal{B}$. If $\mathcal{A}$ prevails, it means that the total amount of goods remaining is $sum < (\sum_{i=1}^{d} num_i - \sum_{j=1}^{t} elig_j)$. In this case, $\mathcal{B}$ has a non-negligible advantage to break the ECDLP assumption. This is because when $\mathcal{A}$ takes some of the goods unauthorized approval, $\mathcal{A}$ has to figure out the private key of DE and then launches tx_{re} to conceal the conduct. If $\mathcal{A}$ has a non-negligible advantage in winning the $Game_{Audit}$, then $\mathcal{B}$ can break the ECDLP assumption. Consequently, the advantage of $\mathcal{A}$ wins is negligible under the ECDLP assumption condition, and Σ is auditable. ■

VIII. EFFICIENCY ANALYSIS

Based on the Fisco Bcos [38] platform, we have developed a prototype of Eisdspa and have completed in-depth testing on smart contract gas consumption. Along with thoroughly testing the proof and validation time overheads for ZKPoK_0, ZKPoK_1, ZKPoK_2, and ZKPoK_3, we have also assessed the computational cost of each participating entity in Eisdspa. Our experiments are conducted on a Linux VM-12-16-ubuntu 20.04 system with a 320 GiB hard drive and an Intel Xeon Platinum 8255C CPU running at 2.50 GHz. To evaluate gas usage and zero-knowledge proof implementations, we utilize the Foundry testing framework, which leverages Python and Rust languages, in conjunction with the Fisco Bcos 2.9.2 blockchain.

A. COMPUTATIONAL COSTS

We analyze the computational overhead of the four entities–CO, DO, DE, and AO–and compared the results with Astraea [6], BBD [5] and UpHaaR [28], as indicated in Table 2 to assess the computational overhead of each entity.

During the donation phase, DO runs ZKPoK_1 for identity authentication requiring T_{1p}, and DO conducts vector multiplication with point-add operations in $\mathbb{G}_0$. This computes the commitment com_{num} and the token res for the donation number, requiring $2T_{mul} + T_{add}$ and T_{mul} operations, respectively.

TABLE 3. Gas costs and monetary costs.

Transaction		donate	store	distribute	receive	audit
Cost	GAS (gas)	110286	162196	150071	110248	75910
	USD[1]($)	6.06	8.91	8.25	6.06	4.17

[1] The USD costs were estimated based on the prices (gas to Gwei and ETH to USD) on October. 1, 2024.

The hashing time T_{hash} for the basic goods information is also calculated. Throughout the store and distribute phases, CO runs ZKPoK_1 for identity authentication requiring $2nT_{1v}$. And CO updates mul_{com} and mul_{tok} twice, requiring a total of $6T_{mul}+2T_{add}$ operations. Additionally, it also calculates σ_{Ac} in T_{Sig} time. In addition, CO computes the new commitment and token of the donation number and receiving number twice and hashes the goods information twice, taking $6T_{mul}+2T_{add}+2T_{hash}$ time. Before receiving goods, DE runs ZKPoK_1 for verifying identity requiring T_{1p} and uses $2T_{mul}+T_{add}$ and T_{hash} to calculate the commitment and token of the receiving number and the hash value of the received goods information, respectively. During the audit phase, AO only needs to confirm ZKPoK_2 and ZKPoK_3, taking $T_{2v}+T_{3v}$. This is in stark contrast to Astraea, which requires the cumulative calculation of every proof of donation or distribution.

As demonstrated in Table 2, DO and DE have a larger time overhead than Astraea because, in contrast to Astraea, DO and DE must complete identity registration and authentication. In addition, the cyclic groups $\mathbb{G}_0$ and $\mathbb{G}_1$ of elliptic curves as well as bilinear pairings–which are more complicated than straightforward scalar multiplicative additions–provide the foundation for the addition and vector multiplication operations. Even if CO requires DE and DO to complete identity authentication again, there is still very little overhead associated with it. Additionally, compared to BBD and UpHaaR, our approach offers superior computation time for all members with minimal overhead. The computational cost of AO in Astraea, UpHaaR increase linearly for large-scale donors and donees, and its auditability is passive and necessitates traversing through every donation transaction. The overhead of AO is also proportional to the number of users in the system. In our scheme, AO only needs to proactively audit ZKPoK_2 and ZKPoK_3 and confirm their accuracy.

B. GAS COSTS

In Eisdspa, there are five transactions: tx_{do}, tx_{st}, tx_{di}, tx_{re} and tx_{au}. We constructed testing contracts based on Foundry for each of these transactions and evaluated how much gas and USD they consumed. The costs of the five transactions are displayed in Table 3.

We also performed a comparison with Astraea [6], as Fig. 2. illustrates. The results show that three transactions, tx_{do}, tx_{di}, and tx_{au}, use at least 1/3 less gas than Astraea, while tx_{st} and tx_{re} use somewhat more. In particular, the ability of auditing is proactive in Eisdspa; it merely requires witnessing the last transaction and does not require going through every donation or distribution action repeatedly. As a consequence, a lot of gas is used during the process to record data structures called mul_{com} and mul_{tok}. Furthermore, our other transactions do not consume more gas compared to Astraea.

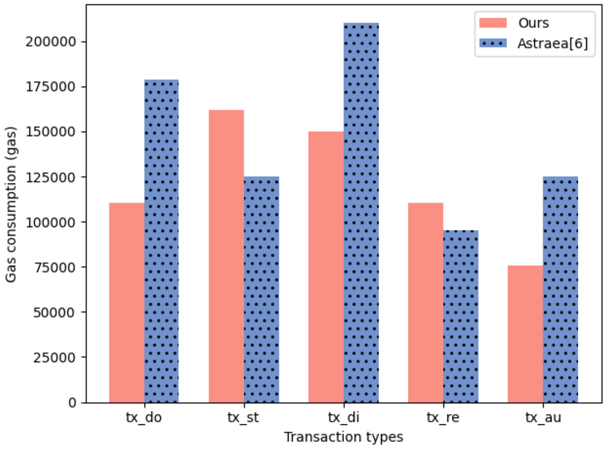

FIGURE 2. Gas comparison.

TABLE 4. The prove time and verify time of ZKPoKs (ms).

ZKPoKs	ZKPoK_0	ZKPoK_1	ZKPoK_2	ZKPoK_3
Proof time	0.92	14.72	3.75	3.31
Verify time	1.20	18.78	5.11	4.62

C. ZKPOK IMPLEMENTATIONS AND ASSESSMENT

We implement ZKPoK_0, ZKPoK_1, ZKPoK_2, and ZKPoK_3 with secure lengths exceeding 80 bits, using the Python programming language and the Charm-Crypto 0.50 library. The cyclic group of integers is based on a 1024-bit modulus. We then carefully examine the proof and validation time overheads of each ZKPoK. For each proof, we calculate the average time over 1000 experiments. The time overheads for the corresponding validation and proving processes are displayed in Table 4. In ZKPoK_1, random number blinding is necessary to ensure user anonymity. Additionally, because it relies on zero-knowledge proofs with bilinear pairings, it incurs a relatively large overhead. For the remaining three ZKPoKs, the time overheads for the proving and verification phases are extremely low. However, the time required for the verification phase exceeds that of the proving phase in each case. Specifically, the verifier must integrate responses from the prover to ascertain the equality of the results.

D. COMPARISON OF FUNCTIONALITY

We compare our scheme with existing works in terms of the five functional realizations of authentication, privacy protection, auditability, access control, and anonymity in Table 5. Astraea and Aid focus mainly on the interaction

TABLE 5. Comparison of functionalities.

Scheme	Auth	Priv	Audi	Acce	Anon
Astraea [6]	✗	✓	✓	✓	✗
Organ [7]	✓	✓	✓	✗	✗
Aid [10]	✓	✓	✓	✓	✗
UpHaaR [28]	✓	✓	✗	✓	✗
BBD [5]	✗	✓	✓	✓	✗
Ours	✓	✓	✓	✓	✓

Auth = Authentication; Priv = Privacy Protection; Audi = Auditability; Acce = Access Control; Anon = Anonymity.

between DO and CO in access control; however, DOs are only able to access information about the goods they have provided and are not able to fully understand the progress of the project. Additionally, the restriction on access is too weak in UpHaaP; any user has access to critical data, which is detrimental to privacy. BBD offers strong privacy features but lacks authentication and anonymity, making it less favorable for donors and donees when donation projects are public. Our solution strikes a compromise between access restriction and privacy protection, allowing DO to see the distribution details of the projects to which they donated but denying them access to those who do not donate. Furthermore, although many methods fail to implement the anonymous donation and receipt function, our system achieves this based on the BBS+ signature and ZKPoKs to enable anonymous users to donate and receive goods.

IX. CONCLUSION AND FUTURE DIRECTIONS

In this work, we present a blockchain-based goods donation system with auditability, authentication, and privacy protection. In order to accomplish the three objectives, we construct dependable ZKPoKs with BBS+ signatures, realize safe authenticity and donation privacy, and facilitate proactive and effective auditability utilizing the additive homomorphism of the Pedersen commitment. We formally define and prove privacy and security. The outcomes of the experiment demonstrate its outstanding practicability and efficiency. However, as our research relies on bilinear pairings with marginally more overhead, our authentication is less effective. Therefore, we will concentrate on creating lightweight authentication in our next effort to increase user authentication efficiency. Furthermore, given the poor performance of the single chain, we are considering investigating cross-chain regulation in the future to implement a model that combines the regulatory chain and the donation business chain.

REFERENCES

[1] A. Carnegie. "Carnegie corporation of New York." Accessed: Aug. 15, 2024. [Online]. Available: https://www.carnegie.org/

[2] C. L. Anita Alfaro. "Donorsnap." Accessed: Aug. 15, 2024. [Online]. Available: https://info.gartnerdigitalmarkets.com/donorsnap-gdm-lp?category=donation-management&utm_source=GetApp

[3] L. Vera. "Betterworld." Accessed: Aug. 15, 2024. [Online]. Available: https://betterworld.org/blog/nonprofits/16-best-fundraising-platforms-for-charity-and-boosting-impactful-giving/

[4] A. Glazer and G. Weiner. "Whole whale." Accessed: Aug. 15, 2024. [Online]. Available: https://www.wholewhale.com/tips/best-donation-platforms-nonprofits/

[5] M. Kaur, P. D. Kaur, and S. K. Sood, "Blockchain oriented effective charity process during pandemics and emergencies," *IEEE Trans. Comput. Soc. Syst.*, vol. 11, no. 1, pp. 431–441, Feb. 2024.

[6] M. Li et al., "Astraea: Anonymous and secure auditing based on private smart contracts for donation systems," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 4, pp. 3002–3018, Jul./Aug. 2023.

[7] D. Hawashin, R. Jayaraman, K. Salah, I. Yaqoob, M. C. E. Simsekler, and S. Ellahham, "Blockchain-based management for organ donation and transplantation," *IEEE Access*, vol. 10, pp. 59013–59025, 2022.

[8] M. Breuer, U. Meyer, S. Wetzel, and A. Mühlfeld, "A privacy-preserving protocol for the kidney exchange problem," in *Proc. 19th Workshop Privacy Electron. Soc.*, 2020, pp. 151–162.

[9] R. Vanholder et al., "Organ donation and transplantation: A multi-stakeholder call to action," *Nat. Rev. Nephrol.*, vol. 17, no. 8, pp. 554–568, 2021.

[10] B. Wang, W. Lueks, J. Sukaitis, V. G. Narbel, and C. Troncoso, "Not yet another digital ID: Privacy-preserving humanitarian aid distribution," in *Proc. IEEE Symp. Security Privacy (SP)*, 2023, pp. 645–663.

[11] A. Sonnino, M. Al-Bassam, S. Bano, S. Meiklejohn, and G. Danezis, "Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers," in *Proc. Netw. Distrib. Syst. Security (NDSS) Symp.*, 2019, pp. 24–27.

[12] M. Li et al., "Anonymous, secure, traceable, and efficient decentralized digital forensics," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 5, pp. 1874–1888, May 2024.

[13] L. Lu et al., "iQuery: A trustworthy and scalable blockchain analytics platform," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 6, pp. 4578–4592, Nov./Dec. 2023.

[14] Z. Zhang et al., "A blockchain-based privacy-preserving scheme for sealed-bid auction," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 5, pp. 4668–4683, Sep./Oct. 2024.

[15] J. Shu, X. Zou, X. Jia, W. Zhang, and R. Xie, "Blockchain-based decentralized public auditing for cloud storage," *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2366–2380, Oct.–Dec. 2022.

[16] M. R. Kabir, F. A. Sobhani, N. Mohamed, and M. Ashrafi, "Impact of integrity and internal audit transparency on audit quality: The moderating role of blockchain," *Manag. Account. Rev.*, vol. 21, no. 1, pp. 203–233, 2022.

[17] M. H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-TAA," in *Proc. Int. Conf. Secur. Cryptogr. Netw.*, 2006, pp. 111–125.

[18] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Proc. Annu. Int. Cryptol. Conf.*, 2004, pp. 56–72.

[19] J. Doerner, Y. Kondi, E. Lee, A. Shelat, and L. Tyner, "Threshold BBS+ signatures for distributed anonymous credential issuance," in *Proc. IEEE Symp. Security Privacy (SP)*, 2023, pp. 773–789.

[20] U. Fiege, A. Fiat, and A. Shamir, "Zero knowledge proofs of identity," in *Proc. 19th Annu. ACM Symp. Theory Comput.*, 1987, pp. 210–217.

[21] H. Wang, Y. Guo, R. Bie, and X. Jia, "Verifiable arbitrary queries with zero knowledge confidentiality in decentralized storage," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 1071–1085, 2024.

[22] D. Mouris and N. G. Tsoutsos, "Zilch: A framework for deploying transparent zero-knowledge proofs," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3269–3284, 2021.

[23] V. Shoup, "Practical threshold signatures," in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2000, pp. 207–220.

[24] H. Han, R. K. Shiwakoti, R. Jarvis, C. Mordi, and D. Botchie, "Accounting and auditing with blockchain technology and artificial intelligence: A literature review," *Int. J. Account. Inf. Syst.*, vol. 48, Mar. 2023, Art. no. 100598.

[25] Y. Chen, X. Ma, C. Tang, and M. H. Au, "PGC: Decentralized confidential payment system with auditability," in *Proc. 25th Eur. Symp. Res. Comput. Secur.*, 2020, pp. 591–610.

[26] A. Almaghrabi and A. Alhogail, "Blockchain-based donations traceability framework," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9442–9454, 2022.
[27] N. C. Will and C. A. Maziero, "Intel software guard extensions applications: A survey," *ACM Comput. Surv.*, vol. 55, no. 14s, pp. 1–38, 2023.
[28] D. Saraswat, F. Patel, P. Bhattacharya, A. Verma, S. Tanwar, and R. Sharma, "UpHaaR: Blockchain-based charity donation scheme to handle financial irregularities," *J. Inf. Secur. Appl.*, vol. 68, Aug. 2022, Art. no. 103245.
[29] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, New York, NY, USA: ACM, 2019, pp. 203–225.
[30] F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven, "Better zero-knowledge proofs for lattice encryption and their application to group signatures," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Security*, 2014, pp. 551–572.
[31] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Proc. Annu. Int. Cryptol. Conf.*, 1991, pp. 129–140.
[32] A. Joux, "A one round protocol for tripartite Diffie–Hellman," in *Proc. Int. Algorithmic Number Theory Symp.*, 2000, pp. 385–393.
[33] S. Tessaro and C. Zhu, "Revisiting BBS signatures," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2023, pp. 691–721.
[34] J. Ni, M. H. Au, W. Wu, X. Luo, X. Lin, and X. S. Shen, "Dual-anonymous off-line electronic cash for mobile payment," *IEEE Trans. Mobile Comput.*, vol. 22, no. 6, pp. 3303–3317, Jun. 2023.
[35] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," in *Proc. Decent. Bus. Rev.*, 2008, Art. no. 21260.
[36] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, 2018, pp. 1–15.
[37] R. G. Brown, J. Carlyle, I. Grigg, and M. Hearn, *Corda: An Introduction*, R3 CEV, New York, NY, USA, 2016, p. 14.
[38] (Shanghai Free Trade Zone's Financial Innovation Operational Office, Shanghai, China). *FISCO BCOS Documentation*. Accessed: Aug. 15, 2024. [Online]. Available: https://fisco-bcos-documentation.readthedocs.io/zh-cn/latest/

YONG ZHOU received the bachelor's degree from the School of Cyber Security (School of Cryptography), Hainan University, Haikou, China, in 2018, where she is currently pursuing the master's degree in cyberspace security with the School of Cyber Security (School of Cryptography). Her areas of research interest are blockchain security and cryptography.

HONG LEI received the bachelor's and master's degrees from Beihang University, Beijing, China, in 2006 and 2009, respectively, and the Ph.D. degree from Michigan State University in May 2015, where he continued as a Postdoctoral Fellow with Smart Microsystem Laboratory. He joined Schweitzer Engineering Laboratory in 2016, and then joined the Department of Electrical and Computer Engineering, Portland State University as a Tenure-Track Assistant Professor in July 2018. He was appointed as the Associate Dean of Oxford-Hainan Blockchain Research Institute in June 2019. He is currently a Professor with Hainan University, and doing researches on TEE and blockchain.

ZIJIAN BAO received the Ph.D. degree from the Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China, in December 2023. He is currently a Postdoctoral Fellow with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His major research interests include applied cryptography and blockchain security.