Blockchain-Assisted Cross-Domain Data Governance for Social Computing

Shulei Zeng, Yao Sun, and Mugen Peng, *Fellow, IEEE*

***Abstract*—With the vigorous development of the social network, massive data have generated by large amount of users. To utilize these data, social computing is widely used. However, due to the conflict between proprietary data and the use of data, it is a challenge to fully obtain data value in an efficient and legal way. Moreover, how to manage, govern and schedule data, a technical and feasible approach is still a major problem. To this end, as an emerging distributed ledger technology, blockchain is considered a promising technology for data security and privacy and is introduced for cross-domain data governance. In this paper, we propose a blockchain-assisted cross-domain data sharing scheme for social computing data governance. Specifically, permissioned blockchain is introduced to construct trust among different parties. A zero-knowledge proof scheme is designed to verify data ownership confidentially and anonymously. The security of the zero-knowledge proof scheme is also analyzed. The extensive simulations and experiments have proven the effectiveness and efficiency of the mechanism.**

***Index Terms*—Blockchain, social computing, data governance, data privacy, zero-knowledge proof.**

I. Introduction

With the explosive growth of social network [1] in recent years, an enormous amount of data can be generated anywhere and at any time. To fully exploit the value of these data in such algorithm systems, it is necessary to collect data from different sources for exploration on a large scale [2]. However, due to different ownership, the collection, storage, use and sharing of data should be guaranteed by trusted entities [3]. To this end, data governance is proposed to enforce compliance with the use of data [4].

Data governance refers to the process of managing, protecting, and using an organization's data assets effectively and efficiently [5]. The goal of data governance is to ensure the accuracy, completeness, consistency, and security of an organization's data as well as to ensure that the data are used appropriately and in accordance with legal and regulatory requirements [6].

This work was supported in part by the National Key R&D Program of China under No.2021YFB1714100, and in part by the National Natural Science Foundation of China under No.U22B2006, and in part by the Science and Technology Innovation Foundation of China Electronics Technology Group under SXX22641X036, and in part by the Science and Technology on Communication Networks Laboratory Fund Project under FFX22641X009.

S. Zeng and M. Peng is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Science and Technology on Communication Networks Laboratory. (e-mail: zengsl@bupt.edu.cn, pmg@bupt.edu.cn).

Y. Sun is with the School of James Watt School of Engineering, University of Glasgow, Glasgow, U.K. (e-mail: Yao.Sun@glasgow.ac.uk).

While data governance is essential for effective data management, there are still some challenges in implementing and maintaining data governance. A data governance framework should ensure that reasonable and appropriate measures are taken to protect information resources in cross-domain data-sharing scenarios [7]. Disclosed identity reveals the corresponding private data, so that the use of data processing should be anonymous to protect privacy and security. Data ownership is the premise and basis of transactions. Unclear data ownership leads to legal risks. However, identifying the data owner might be a difficult task [8].

To settle the issues of security and privacy in data governance, blockchain [9] might be a feasible method. As a distributed ledger technology, blockchain has achieved a series of satisfactory features: decentralization, invariance, transparency, integrity, fault tolerance and security [10]. A smart contract [11] is a program or a transaction protocol that runs on the blockchain. Therefore, blockchain can achieve self-execution and trusted execution of contracts and promote cooperation between independent and untrusted parties. When applying smart contract technology to data governance, smart contracts inevitably interact with the real world and accept sensitive information from the real world as input. Therefore, they require appropriate data privacy protection in addition to data authenticity.

The security and privacy requirements of data governance in social computing, which are entangled with the privacy issues of smart contracts, are highly challenging due to blockchain's transparency and decentralization. To achieve both security and privacy when applying blockchain to data governance in social computing, this paper proposes a blockchain-assisted cross-domain data sharing scheme (BCDS) to build a heterogeneous data governance system for social computing, which allows entities to provide cross-domain data transmission for task cooperation while guaranteeing data confidence. With the ingenious design, data ownership can be easily verified if needed. The main contributions of this paper are summarized as follows:

- We design BCDS to address the challenges of security and heterogeneity, this scheme supports confidential data sharing and anonymous identity authentication in social computing data governance.
- Based on BCDS, we propose a zero knowledge proof scheme called BCDS-ZK to protect privacy in data sharing. Based on this design, data ownership can be verified confidentially and anonymously among domains through

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

smart contracts.

- We analyze the security of BCDS-ZK through simulation-based proof. We also validate the feasibility of BCDS and obtain insightful results through extensive experiments by adopting Hyperledger Fabric to perform BCDS and obtain the following results: (i) compared with the ECDSA scheme, BCDS has higher computation efficiency, (ii) when the number of users increases, the computational overhead of all mechanisms increases approximately linearly with the increase in the number of users, and (iii) compared with the simulation of a single device, when the BCDS is deployed in a distributed environment, it exhibits a very similar performance.

The rest of this paper is organized as follows: In Section II, we introduce preliminaries on BCDS. In Section III, we provide an overview of the proposed framework and design details. In Section IV, we analyze the security and performance of BCDS. In Section V, we reviews related work for data governance and zero-knowledge proofs. In Section VI, we provide conclusions.

II. Preliminary

This section briefly reviews the techniques of zero-knowledge argument and simulation-based security model, which are key components in the BCDS design.

A. Zero-Knowledge Argument

A zero-knowledge proof is a kind of protocol in which the prover can convince the verifier that some statements are true without revealing any information about the reasons for the establishment. An argument is a kind of proof that only holds when the verifier is bounded in calculation and some computational hardness assumptions hold. In general, a zero-knowledge proof of argument should satisfy the following properties:

- **Perfect Completeness**. For every security parameter, an honest prover can convince an honest verifier with a valid witness.
- **Soundness**. A valid proof is sound if an honest verifier can be convinced only of true statements by a prover who is computationally limited and follows the protocol correctly.
- **Computational Zero-Knowledge**. We say that an effective proof has computational zero knowledge if it does not reveal any information about the witness to any polynomial-time adversary.

B. Simulation-based Security Model

Simulation-based formulation of security is a way of comparing what happens in the 'real world' to what happens in an 'ideal world' where the primitive in question is secure by definition. The definition states that an encryption scheme is secure if they can both learn approximately the same amount of information.

Specifically, the simulation-based security proofs work by constructing a simulator that resides in the ideal world that makes it impossible for the adversary to distinguish between two worlds, then the protocol is considered to be achieving the security attributes defined in the ideal function.

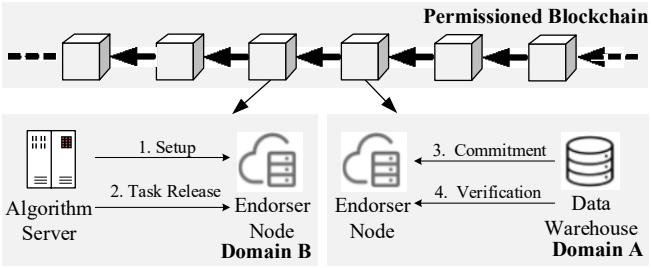


Fig. 1. Overview of the BCDS.

III. Blockchain Assisted Cross-domain Data Sharing Scheme

In this section, we first describe the architecture of the presented cross-domain data-sharing scheme. Then, we introduce details of the proposed BCDS-ZK.

A. Architecture Overview

As shown in Fig. 1, BCDS is based on permissioned blockchains that contain several administrative domains, and a blockchain domain is deployed in an internal network that is composed of multiple physical domain nodes to form a administrative domain. Each domain of blockchains represents a network node within a single organization in the social computing data governance business. It is assumed that there are server devices with powerful computing and storage capabilities in each domain, which can be used to store huge data generated by the social network and have computational capabilities that support the social computing. The consensus of blockchain generally adopts Raft [12] or PBFT (Practical Byzantine Fault Tolerance) [13], which have the ability to resist single points of failure or Byzantine fault tolerance capability with high performance.

The permissioned blockchain involves multiple roles, including client nodes, certificate issuing centers (CA), peer nodes, and endorser nodes. A client node offers an algorithm server, data warehouse or other ordinary node that does not have management authority in the domain's inner network. The CA node is unique in a domain and is responsible for managing the public keys of the domain's ordinary nodes. In addition, the CA cooperates with the endorser node to realize the cross-domain smart contract. A peer node participates in the blockchain network and maintains a copy of the distributed ledger. An endorser node is a type of peer node that is responsible for presiding over the consensus voting of the blockchain and cooperates with the CA node to realize the cross-domain smart contract.

Algorithm 1 BCDS-ZK Algorithm

Setup:
1: Initialize system parameters of BCDS in each domain including $\{\mathbb{G}, P_1, P_2\}$;
2: Pick up a cryptographic hash function H_1;
3: Disclose $\{\mathbb{G}, P_1, P_2, H_1\}$;
4: Generate asymmetric keys pair;
5: Register the public key of each node;
Task Release:
1: Pick a random integer μ;
2: Compute and disclose $H = \mu P_1$;
Commitment:
1: U_A pick a random integer α, calculates $x = H_1(D_A||\alpha)$, $X = xP_1 + \alpha P_2$,$Y = \alpha P_1$,$Z = xP_2$;
2: U_A send $(X||Y||Z||D_A)$ to U_B;
3: U_B decrypt the dataset D_A;
4: U_B executes and send the result m_B back to E_A;
Verification:
1: U_A picks a random integer β and calculates $y = x\beta + \alpha\beta^{-1}$;
2: U_A sends $(y||\beta)$ to U_B;
3: U_B computes $R = \beta^{-1}P_1 + \beta P_3$;
4: U_B computes , $S = \beta^2 Z + X + \beta^{-2}Y$, $T = yR$;
5: If $S = T$ holds, the verification suceed, otherwise fails;
6: U_B executes and send the result m'_B back to E_A.

B. BCDS-ZK: The Zero Knowledge Proof for Blockchain-assisted Data Sharing Scheme

The zero-knowledge proof for the blockchain-assisted data sharing scheme (BCDS-ZK) is a zero-knowledge argument that consists of the following operations: setup, task release, commitment and verification. Alg. 1 describes a generic BCDS-ZK that employs the scheme to preserve privacy for cross-domain data sharing, which runs according to the following steps:

Setup. According to the scheme, each domain of the blockchain is initialized. The system parameters of BCDS in each domain are instantiated, including group $\mathbb{G}$ with order of large prime p. Then, random generators $P_1 \in \mathbb{G}$ and $P_2 \in \mathbb{G}$ and a cryptographic hash function H_1 are picked. $\{\mathbb{G}, P_1, P_2, H_1\}$ is disclosed.

Meanwhile, asymmetric key pair (d, Q) has been generated. The public key Q of the user node in each domain is registered in the public key directory of the CA node. It is only disclosed to the nodes inside the domain, which represents the identity of the node itself. Then, the public key of the endorser node is registered and disclosed to the consortium blockchain for cross-domain smart contracts.

Task Release. We assume that the node of data warehouse U_A, which is located in domain A, stores dataset D_A, which can be used for social computing, and provides it to social computing server node U_B located in domain B . The task release phase is initialized by user node U_B. As a typical cross-domain data sharing, U_B is called the requester in domain B and needs the data that are held in domain A.

To provide privacy protection for the data provider, the requester first announce its requirement to E_B, and then, E_B publishes it as a smart contract to all domains throughout the blockchain network.

Commitment. After the provider responds, the requirement (i.e., send required data) is provided to the requester with the commitment values. The data provider U_A first requests to obtain a temporary anonymous identity certificate from CA_A. CA_A responds with $(r||t)$, where r is a random number that represents the serial number of the temporary anonymous identity certificate. t indicates the time when identity is generated. After receiving the response, U_A generates a random integer α and calculates $x = H_1(D_A||\alpha)$, $X = xP_1 + \alpha P_2$, $Y = \alpha P_1$, and $Z = xP_2$.

U_A sends $(X||Y||Z||r||t||D_A)$ to E_A, where D_A represents the dataset generated by U_A, which is defined according to the smart contract. E_A forwards $(C||L||M||r||t||D_A)$ to E_B through a cross-domain smart contract; E_B forwards them to U_B.

U_B executes the task and sends the execution result m_B back to E_B, and E_B forwards m_B to E_A. Meanwhile, E_B generates a digest of smart contracts and broadcasts it to both domains' blockchains.

Verification. Due to privacy protection by commitment, the provider is anonymous. However, for data sharing benefits such as incentives and reputation, the provider should be able to prove its data ownership as well. U_A first obtains a temporary anonymous identity certificate from CA_A. CA_A responds to the identity $r'||t'$. After receiving the temporary anonymous identity, U_A generates a random integer β and calculates $y = x\beta + \alpha\beta^{-1}$.

U_A sends $(y||\beta||r'||t')$ to E_A. E_A forwards them to E_B, E_B forwards them to requester U_B, and U_B computes $R = \beta^{-1}P_1 + \beta P_2$, $S = \beta^2 Z + X + \beta^{-2}Y$, $T = yR$. If $S = T$ holds, the verification succeeds; otherwise, it fails.

U_B sends verification result m'_B to E_B, and E_B forwards m'_B to E_B. Meanwhile, E_B generates a digest of smart contracts and broadcasts it to both domains' blockchains.

IV. Discussion and Analysis

In this section, we first analyze the security of the scheme. Subsequently, experiments are conducted based on the scheme proposed in this paper to evaluate its performance on several key indicators, including computational overhead and communication overhead, etc.

A. Security Analysis of BCDS-ZK

The BCDS-ZK protocol has perfect completeness, perfect honest verifier zero-knowledge and computational special soundness. Perfect completeness follows directly from the following

$$\begin{aligned} S &= \beta^2 Z + X + \beta^{-2} Y \\ &= (x + \beta^{-2}\alpha)P_1 + (x\beta^2 + \alpha)P_2, \end{aligned} \tag{1}$$

Algorithm 2 Simulator for perfect honest verifier zero-knowledge

1: $\beta \xleftarrow{\$} \mathbb{Z}_p$;
2: $Y, Z \xleftarrow{\$} \mathbb{G}_1$;
3: $y = \beta + \beta^{-1}$.;
4: $X = P_1 + \beta^{-2}P_1 + P_2 + \beta^2 P_2 - \beta^2 Z - \beta^{-2}Y$;
5: Output:$\{\beta, X, Y, Z, y\}$;

$$\begin{aligned} T &= yR \\ &= (x + \beta^{-2}\alpha)P_1 + (x\beta^2 + \alpha)P_2, \end{aligned} \tag{2}$$

and the relation $S = T$ holds whenever the prover knows a witness to the relation and is honest.

To prove perfect honest verifier zero-knowledge, we construct a simulator [14], [15] that produces a distribution of proofs for a given statement $\{P_1, P_2 \in \mathbb{G}\}$, and the proof is indistinguishable from valid proofs produced by an honest prover. The simulator acts as defined in Algorithm 3.

The values β produced by an honest prover interacting with an honest verifier are uniformly distributed random independent elements. The honestly produced $\{Y, Z\}$ are random independent elements that are protected by **DLP**. Thus, the transcript of the proof is identically distributed to an honestly computed proof with uniformly selected challenges. The simulator runs in time $O(V)$ and is thus efficient.

To prove computational special soundness, we construct an efficient extractor ε for BCDS-ZK that, on input $\{P_1, P_2 \in \mathbb{G}\}$, either extracts a witness v_1 such that the relation holds or finds the nontrivial discrete logarithm relation between P_1, P_2.

ε runs the prover to obtain $\{X, Y, Z\}$. Then, ε uses 2 different challenges, β_1 and β_2, and the extractor obtains $y_{(1)}$ and $y_{(2)}$ such that:

$$y_{(i)} = x\beta_{(i)} + \alpha\beta_{(i)}^{-1}. \tag{3}$$

ε computes $x, \alpha \in \mathbb{Z}_p$ by using these $y_{(i)}$ to compute linear combinations of Equation (6). Since $X = xP_1 + \alpha P_2$, we have that x, α is a valid witness for BCDS-ZK. The extractor rewinds the prover 2 times. Extraction is efficient and polynomial.

B. Simulation Setup

To implement the proposed scheme, we implemented a permissioned blockchain based on the Hyperledger Fabric Test Network. In addition, we also designed and developed a series of cross-domain protocols to support data interaction between different domains. In the blockchain prototype system, Docker is used to publish multiple application images of Fabric nodes to build a simple blockchain network with up to 500 nodes. We set two domains in the experiment. Each domain contains several necessary entities, including the CA node, endorser node, user equipment and servers. All devices are connected to one Ethernet in each domain. The operations of the CA node, endorser node, servers and user node in each domain are in one notebook virtual machine with an AMD Ryzen 7 4800H CPU@2.9 GHz and 16 GB memory. The operating system used by each device is Ubuntu 18.04 LTS.

We implement the mechanism within the domain based on sockets, which means the data exchanges between different entities within domain are encapsulated as TCP packets. Cross domain smart contracts are implemented based chaincode of Fabric. To evaluate the performance of the proposed commitment scheme in practice, we implemented BCDS-ZK in Go and integrated it into the go-ristretto library through elliptic curve edwards25519, which has 256-bit security. In addition, we used the machine learning library MNIST as a dataset for social computing.

C. Simulation Results

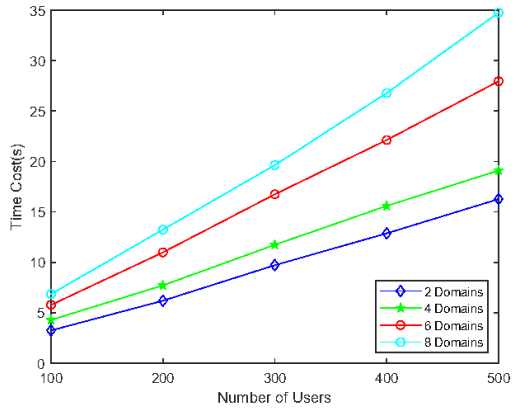
To demonstrate the advantages of the proposed scheme in practice, we first ran BCDS under the setting mentioned in the last section. We record and show the average execution time of each phase in BCDS-ZK compared to ECDSA. As shown in Table I and II, the verification phase only makes up a small part of the computation cost, and the proposed scheme has significant performance advantages. Relatively speaking, generating a proof incurs the heaviest computational overhead because the node needs to calculate the dataset's hash during the commitment phase, which is time-consuming work and only needs to be computed once. In addition, the time cost of calculating the dataset's hash value is proportional to its size.

To further analyze the performance of the proposed scheme, we introduce Hyperledger Fabric as a distributed ledger for sharing data. We compare the average cost time of each phase in BCDS via the number of varying users. We also expanded the blockchain network from 2 domains to 8 domains. In addition, the concurrent queries are expanded to 100-500. Fig. 2 shows the increased time cost of BCDS. Specifically, as shown in Fig. 2a, the time costs of BCDS-ZK on cmmitment phase grows linearly as the number of users increases, as the computational overhead is approximate for each user. As shown in Fig. 2b, the time costs of BCDS-ZK on verification phase have similar curve trends.

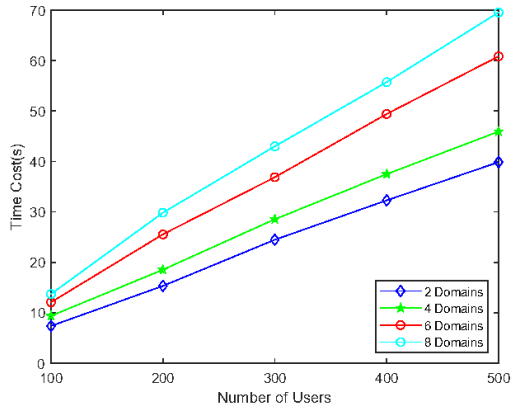
In addition, as Fig. 2a shows the more domains are included, the growth trend of the curve becomes steeper. Note that the time cost increase nonlinearly as the number of domains increases. Fig. 2b shows similar trends. The reason for the trend is that there are a large number of invokes included in BCDS, which require to conduct consensus. It is noted that the consensus time increases with the increase in the number of endorser and validation nodes.

D. Testbed

To verify the feasibility of the proposed scheme, we implemented a production environment on a real testbed containing four host servers. Fig. 3 shows the experimental results of the testbed relative to the simulation of a single device. We set two domains in the testbed. In each domain, one node acts as the prover or verifier with computing and storage capabilities,



(a) Time cost of BCDS-ZK on Commitment phase



(b) Time cost of BCDS-ZK on Verification phase

Fig. 2. Time consumption of BCDS with varying parameters

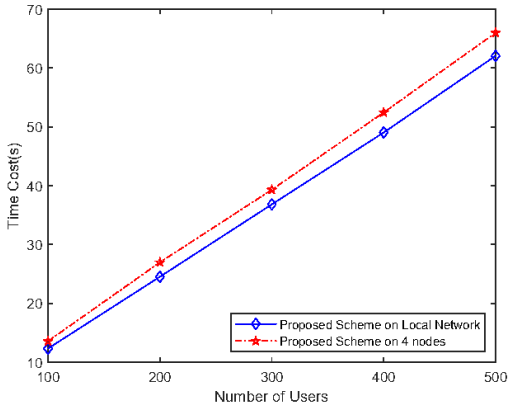


Fig. 3. Execution time of the scheme in real implementation

TABLE I
COMPUTATION OVERHEAD OF BCDS-ZK ON EACH PHASE (UNIT: MILLISECOND)

	Commitment	Verification	Total
BCDS-ZK	24.257	0.314	25.693
ECDSA	26.006	2.461	28.467

TABLE II
COMMUNICATION OVERHEAD OF BCDS-ZK ON EACH PHASE (UNIT: MILLISECOND)

	Commitment	Verification	Total
BCDS-ZK	222.48	221.13	443.61
ECDSA	221.75	222.32	444.07

and the other hosts acts as the consensus node. All devices are connected to one Ethernet in the testbed.

As Fig. 3 shows, the blockchain network deployed in a distributed environment has a very close performance to the blockchain network deployed in a single node because the experiment is based on Hyperledger Fabric, whose network performance mainly depends on the orderer peer. We can conclude that the proposed blockchain solution can run effectively in practical situations and obtains good performance when implementing privacy protection schemes in such implementation experiments.

V. RELATED WORK

In this section, we briefly review and summarize representative research within the scope of this article.

A. Solutions for Data Governance

The research and solutions for data governance have recently received great attention [7]. Abraham *et al.* [5] reviewed the existing research questions of data governance to develop a conceptual framework for data governance. Marijn Janssen *et al.* [4] proposed a framework for data governance for trustworthy big data algorithmic systems by reviewing challenges and approaches to data governance. Vilminko and Pekkola [8] analyzed data ownership, data governance roles and responsibilities in terms of mastering data governance. Alhassan *et al.* [16] reviewed and presented the critical success factors for data governance. These works try to construct data governance framewokrs with different techniques.

As an emerging technology, blockchain has been proposed to build data governance solutions by both industry and academia. Liu *et al.* [17] proposed a new data governance approach that is built upon the blockchain-based decentralized services computing paradigm. Ølnes *et al.* [18] reviewed and discussed the benefits of blockchain technology for data governance. The results suggest that the realization of the data allocation mechanism improves network usage, latency, and blockchain storage and reduces energy consumption. Akkaoui *et al.* [19] presented a secure and efficient blockchain-based data governance framework for sharing health data. Although there have been a dozen of blockchain based data governance solutions proposed recently, they do not solve the data ownership problem. Thus, they may not be suitable to share data

among different parties.

B. Zero-Knowledge Proofs

Zero-knowledge proofs were first invented by Goldwasser *et al.* [20]. In general, zero-knowledge has computational zero-knowledge, while zero-knowledge arguments may have perfect zero-knowledge [21]. Kosba [22] proposed noninteractive zero-knowledge (NIZK) proofs to introduce privacy into smart contracts. Succinct arguments of knowledge (SNARKs) [23] protocols usually have succinct proofs and efficient verifiers with a complex trusted setup. Bünz *et al.* [14] proposed a non-interactive zero-knowledge proof protocol named bulletproof with short proofs and without a trusted setup.

In addition, there are several applications that introduce privacy into blockchains with zero-knowledge proofs. As a privacy-preserving smart contract framework, Hawk [22] provides a compiler tool that solves the problem of transaction privacy and can compile a smart contract. Wan et al. [11] proposed a zero-knowledge SNARK scheme for authenticated data that combined the zk-SNARK technique with a digital signature in an effective way.

To address the challenges in the research mentioned above, we focus on the issue of cross-domain data sharing in social computing, aiming to design a feasible solution to share data through permissioned blockchain and enable data owners to verify ownership, and formally prove the security of solution.

VI. Conclusion

In this paper, we propose a blockchain-assisted data sharing scheme BCDS for cross-domain data governance in social computing. Specifically, the permissioned blockchain is introduced to construct trusted data governance. Further, data ownership can be verified through BCDS-ZK anomymously and privately. Finally, the security and privacy of the proposed blockchain framework and secret sharing scheme are analyzed. The experimental results demonstrate the effectiveness and efficiency of the proposed scheme.

References

[1] L.-L. Shi, L. Liu, Y. Wu, L. Jiang, M. Kazim, H. Ali, and J. Panneerselvam, "Human-centric cyber social computing model for hot-event detection and propagation," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1042–1050, 2019.

[2] J. Bai, G. Huang, S. Zhang, Z. Zeng, and A. Liu, "Ga-dctsp: An intelligent active data processing scheme for uav-enabled edge computing," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 4891–4906, 2023.

[3] B. Yin, H. Yin, Y. Wu, and Z. Jiang, "Fdc: A secure federated deep learning mechanism for data collaborations in the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6348–6359, 2020.

[4] M. Janssen, P. Brous, E. Estevez, L. S. Barbosa, and T. Janowski, "Data governance: Organizing data for trustworthy artificial intelligence," *Government Information Quarterly*, vol. 37, no. 3, p. 101493, 2020.

[5] R. Abraham, J. Schneider, and J. Vom Brocke, "Data governance: A conceptual framework, structured review, and research agenda," *International journal of information management*, vol. 49, pp. 424–438, 2019.

[6] M. Mosley, M. H. Brackett, S. Earley, and D. Henderson, *DAMA guide to the data management body of knowledge*. Technics Publications, 2010.

[7] O. Benfeldt, J. S. Persson, and S. Madsen, "Data governance as a collective action problem," *Information Systems Frontiers*, vol. 22, pp. 299–313, 2020.

[8] R. Vilminko-Heikkinen and S. Pekkola, "Changes in roles, responsibilities and ownership in organizing master data management," *International Journal of Information Management*, vol. 47, pp. 76–87, 2019.

[9] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized business review*, 2008.

[10] Z. Wang, B. Cao, C. Liu, C. Xu, and L. Zhang, "Blockchain-based fog radio access networks: Architecture, key technologies, and challenges," *Digital Communications and Networks*, vol. 8, no. 5, pp. 720–726, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864821001036

[11] B. Cao, Z. Wang, L. Zhang, D. Feng, M. Peng, L. Zhang, and Z. Han, "Blockchain systems, technologies, and applications: A methodology perspective," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 353–385, 2023.

[12] D. Huang, X. Ma, and S. Zhang, "Performance analysis of the raft consensus algorithm for private blockchains," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 172–181, 2019.

[13] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OsDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[14] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *2018 IEEE symposium on security and privacy (SP)*, 2018, pp. 315–334.

[15] Z. Wan, Y. Zhou, and K. Ren, "Zk-authfeed: Protecting data feed to smart contracts with authenticated zero knowledge proof," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1335–1347, 2022.

[16] I. Alhassan, D. Sammon, and M. Daly, "Critical success factors for data governance: a theory building approach," *Information Systems Management*, vol. 36, no. 2, pp. 98–110, 2019.

[17] X. Liu, S. X. Sun, and G. Huang, "Decentralized services computing paradigm for blockchain-based data governance: Programmability, interoperability, and intelligence," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 343–355, 2020.

[18] S. Ølnes, J. Ubacht, and M. Janssen, "Blockchain in government: Benefits and implications of distributed ledger technology for information sharing," *Government information quarterly*, vol. 34, no. 3, pp. 355–364, 2017.

[19] R. Akkaoui, X. Hei, and W. Cheng, "Edgemedichain: A hybrid edge blockchain-based framework for health data exchange," *IEEE Access*, vol. 8, pp. 113 467–113 486, 2020.

[20] 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*, 2019, vol. 18, no. 1, pp. 203–225.

[21] G. Brassard, D. Chaum, and C. Crépeau, "Minimum disclosure proofs of knowledge," *Journal of computer and system sciences*, vol. 37, no. 2, pp. 156–189, 1988.

[22] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE symposium on security and privacy (SP)*, 2016, pp. 839–858.

[23] E. Ben-Sasson, A. Chiesa, D. Genkin, E. Tromer, and M. Virza, "Snarks for c: Verifying program executions succinctly and in zero knowledge," in *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, 2013, pp. 90–108.