From Isolation to Integration: A Reputation-Backed Auditable Model for Cohort Data Sharing

Jie Zhang, *Student Member, IEEE*, Xiaohong Li, *Member, IEEE*, Ruitao Feng, Shanshan Xu, Zhe Hou, Hanwei Wu,and Guangdong Bai, *Member, IEEE*

Abstract—Data sharing is vital to breaking data silos and maximizing information value. However, practical implementations often rely on cloud servers, raising trust concerns that prevent Data Centers (DCs) from sharing sensitive data. Motivated by the need to ensure both the quality and quantity of shared data, we proposed a reputation-driven, auditable data-sharing model that uses blockchain to enable secure, distributed sharing. Our model faces two primary challenges: (1) ensuring data quality in distributed settings, where existing cloud-based audit schemes requiring high computational resources are unsuitable, and (2) promoting active sharing of scarce data, where current incentive mechanisms fail to encourage proactive participation. To address these, we introduce the Secure Auditable Sharing Protocol (SASP) and the fair Reputation-driven Proactive Sharing Mechanism (RPSM). SASP enhances ElGamal encryption and integrates efficient hashing techniques for privacy-preserving audits of ciphertext integrity and deduplication without relying on costly bilinear mappings. RPSM tackles the challenge of selfish DCs by incorporating a committee mechanism and consensus algorithm, ensuring fair incentives to encourage active participation. Our implementation and real-world case study demonstrate that the proposed model effectively guarantees the quality and quantity of shared data, offering a novel solution to the data silo problem in distributed architectures.

Index Terms—Data Sharing, Integrity and Deduplication Audits, Reputation-Based Mechanism, Permissioned Blockchain, IND-CPA

1 Introduction

Cohort research is instrumental in uncovering the etiology of diseases, forming the foundation for many advancements in medical science [1]. Cohort data centers (DCs), such as those hosted by hospitals and research institutions [2], play a pivotal role in storing and managing such valuable data. Facilitating cross-center data sharing is crucial to enhancing the utility of cohort data and maximizing its research potential.

However, the highly sensitive, private, and scarce nature of medical cohort data creates significant barriers to sharing. DCs are often reluctant to share their data beyond internal usage due to privacy and security concerns, resulting in the so-called "data island" dilemma. In this scenario, data remains siloed within individual DCs, preventing broader collaboration and hindering scientific progress.

Notably, since 2018, only a single institution has made its cohort data publicly available [2], underscoring the severity of this issue. Similar challenges exist in other domains, such as finance and government, where high-value, sensitive data faces comparable constraints [3]–[5]. This lack of sharing exacerbates the trust deficit in these sectors and limits the potential for large-scale, impactful research [6]–[9].

Motivation. Addressing the "data island" dilemma requires a robust solution that simultaneously ensures the **quality** and **quantity** of shared data. Current approaches predominantly focus on one aspect while neglecting the other, leading to notable deficiencies.

For one thing, **ensuring the quality of shared data is critical, yet distributed architectures lack effective solutions for secure and efficient data auditing.** While encryption is widely adopted to protect sensitive data [10]–[15], sharing ciphertext alone raises concerns about data integrity and relevance [16]. Existing ciphertext auditing schemes [14], [17]–[20] are often computationally intensive and tailored to centralized cloud storage, rendering them unsuitable for distributed environments. There is an urgent need for lightweight and secure auditing mechanisms tailored for distributed architectures.

For another, **effective data sharing also requires a fair and motivating incentive mechanism.** Current reputation-based schemes [21]–[23] encourage minimal compliance rather than proactive sharing, as penalty mechanisms alone fail to incentivize meaningful contributions [23], [24]. Innovative mechanisms that reward active participation and value creation are essential, particularly in scenarios involving high-value, scarce data.

- *This work is supported by the National Key Research and Development Program of China under Grant 2021YFF1201102.*
- *Jie Zhang, Xiaohong Li and Hanwei Wu are with the College of Intelligence and Computing, Tianjin University, Tianjin, China. (email: {jackzhang, xiaohongli, wuhanwei}@tju.edu.cn).*
- *Ruitao Feng is with the Faculty of Science and Engineering, Southern Cross University, Australia (e-mail: ruitao.feng@scu.edu.au).*
- *Shanshan Xu is with the School of Geographic Sciences, East China Normal University, Shanghai, China. (email: s.xu.ecnu@gmail.com).*
- *Zhe Hou is with the School of Information and Communication Technology, Griffith University, Nathan, Australia. (email: z.hou@griffith.edu.au).*
- *Guangdong Bai is with the School of Electrical Engineering and Computer Science, The University of Queensland, Australia (e-mail: g.bai@uq.edu.au).*
- *Ruitao Feng, Zhe Hou and Guangdong Bai are the corresponding authors.*

Proposed Approach. Considering the significant value and privacy concerns of data in healthcare, finance, and government, we present a comprehensive data-sharing model that addresses both quality and quantity challenges through cryptographic innovation and incentive mechanism design. Our solution leverages Permissioned Blockchain (PB) technology, harnessing its decentralization, immutability, and auditability properties [25]–[29]. The PB architecture—enhanced by certificate authentication and smart contract capabilities—provides an ideal foundation for regulated data sharing environments [30]. Our technical contribution unfolds in two complementary dimensions:

First, we introduce the **Secure Auditable Sharing Protocol (SASP)**, which replaces computationally expensive bilinear mapping-based auditing with an enhanced ElGamal cryptosystem [31] integrated with efficient hashing and locality-sensitive hashing techniques [32]. SASP enables ciphertext deduplication and integrity verification while maintaining encrypted data storage locally—only tags and signatures require blockchain storage, significantly reducing DC overhead.

Second, we develop the **Reputation-driven Proactive Sharing Mechanism (RPSM)**, which establishes fair access policies through dynamic reputation matching and committee-based incentive structures. Drawing inspiration from reputation mechanics [24], [33]–[35], RPSM incorporates a novel **Proof-of-Reputation (PoR)** consensus protocol—a Proof-of-Stake variant [36] where reputation values replace monetary stakes in committee elections (formally defined in Section 5.2.3).

The integration of SASP and RPSM within a PB framework creates a complete reputation-backed auditable sharing model that facilitates secure circulation of sensitive data across healthcare, financial, and governmental domains.

Model Overview. Our architecture maintains sensitive data in encrypted form at source DCs while storing verification tags on-chain. Participants earn reputation through successful data sharing activities, with access privileges governed by reputation-based thresholds. Blockchain infrastructure ensures audit trail immutability and automates smart contract-based verification of data deduplication and integrity.

We validate our approach through both theoretical analysis and practical implementation: extensive security proofs demonstrate formal guarantees, while real-world case studies confirm practical viability in resolving data quality and quantity dilemmas in sharing scenarios. Our model has up to 75.44% reduction in theoretical computation cost on-chain compared to existing schemes. Our case study in real-world demonstrates that our model can perform approximately 1000 audits in 2 seconds with about 25% computing power.

Contributions. Our contributions are as follows:

- For scenarios ensuring data residency within DCs, we propose a fully controlled distributed data sharing model, enabling proactive incentives and data quality auditing.
- To ensure the quality of data circulating within the model, we propose a secure, auditable data sharing protocol called SASP, which provides ciphertext deduplication and integrity auditing capabilities for the model.
- To ensure the quantity of data circulating within the model, we propose a fair reputation-driven proactive sharing mechanism called RPSM, which provides proactive incentives for data sharing within the model.
- Our reputation-backed auditable sharing model has undergone extensive security and performance analyses, in particular a real-world case, which results validate its effectiveness and feasibility.

TABLE 1
Comparison of our proposed model with existing works

Properties	[11]	[15]	[12]	[22]	[13]	[18]	[14]	[20]	Ours
Decentralization store		✓		✓	✓				✔
Data protection	✓		✓	✓	✓		✓	✓	✔
Data sharing	✓	✓	✓	✓			✓		✔
Deduplication auditing							✓	✓	✔
Integrity auditing						✓		✓	✔
Restricted access		✓			✓				✔
Incentive sharing				✓					✔
SC evaluation	✓	✓	✓		✓		✓		✔

Organizational Structure. This paper organizes its technical narrative through a logically structured progression:

- **Section 2** establishes the foundational context of distributed sharing models and security threats that inform our threat model design in Section 4
- **Section 3** formalizes the mathematical assumptions and cryptographic primitives that directly enable the security proofs in Section 6
- **Section 4** defines the system model and six formal security objectives that govern the design requirements implemented in Section 5
- **Section 5** details SASP and RPSM mechanisms with explicit technical dependencies on the cryptographic foundations from Section 3
- **Sections 6-7** provide formal proofs for all definitions in Section 4 and computational validation of the implementations from Section 5
- **Section 8** demonstrates real-world validation requiring integration of the theoretical foundations and practical implementations from preceding sections
- **Sections 9-11** synthesize findings from all technical sections to discuss limitations and future directions

2 Background and Related Work

This section reviews three critical foundations for our model: distributed data-sharing models (Section 2.1), blockchain fundamentals (Section 2.2), and security threats (Section 2.3). These collectively establish the technical context essential for comprehending the problem formulation in Section 4 and the subsequent model design in Section 5.

2.1 Distributed Data-Sharing Models

Distributed data-sharing models address limitations of cetralized architectures. Current research bifurcates based on data ontology accessibility:

- **Model-Parameter Sharing (e.g., Federated Learning)** [37]: Prioritizes *data sovereignty* by sharing model parameters instead of raw data, enabling collaborative model training while maintaining data control. However, compared to raw data sharing, it leads to suboptimal model performance and is inherently vulnerable to privacy leakage. Moreover, it lacks effective mechanisms for auditing the quality of data used in local model updates.

- **Data-Ontology Sharing (e.g., Decentralized Data Marketplaces)** [38]: Focuses on enabling *data confidentiality* by distributed infrastructures (e.g., cloud/edge nodes) [12], [20], [21] to facilitate targeted data sharing and permit controlled access to the data entities. However, it faces critical challenges in incentivizing and privacy-preserving for sensitive cohort data sharing.

2.2 Blockchain Foundations

Blockchain leverages *decentralization* [27], *immutability* [26], and *programmability* [28] for secure sharing platforms [11], [12], [15], [22]. The intrinsic cryptographic primitives further reinforce data security guarantees. Existing research focuses on enhancing data confidentiality through advanced cryptographic methods such as: attribute-based encryption for fine-grained access control [10], [39], proxy re-encryption to enable secure delegation of access [11], [12], [40], and identity-based encryption for hierarchical access structures [13], [14]. Despite their privacy-preserving efficacy, these methods inadequately address *ciphertext auditing*, *secure deduplication*, and *computational efficiency* [16]. Scalability enhancements like sharding [41], zero-trust frameworks [42], enhanced consensus [23], [24], [43] et. also lack mechanisms for *data integrity audit* and *collaboration incentivization*. Consequently, two critical research dimensions emerge: **Data-Oriented Security Protocols** and **User-Oriented Incentive Mechanisms**.

2.2.1 Data-Oriented Secure Sharing Protocols

Secure sharing protocols focus on protecting data confidentiality and ensuring data integrity by various cryptographic techniques. Attribute-based and proxy re-encryption schemes enable secure data sharing but introduce significant computational overheads [11], [12], [22]. Identity-based encryption facilitates scalable access control but lacks provisions for deduplication [13]. Integrity audits for cloud-stored data rely on bilinear mappings, which incur high computational costs and fail to address deduplication of shared data [18]–[20], while [44] employs Merkle tree to reduce auditing overhead, but this approach incurs security compromises. Emerging solutions, such as lattice-based Merkle tree constructions [45], address post-quantum integrity proofs but overlook secure deduplication mechanisms. Notably, only Tian *et al.* [14] addressed blockchain-stored data deduplication auditing — albeit with pairing operations unsuitable for resource-constrained environments. These gaps underscore the critical need for lightweight, comprehensive protocols in high-value data sharing.

2.2.2 User-Oriented Incentive Mechanisms

Fair incentive mechanisms aim to address the reluctance of participants to share sensitive data. Token-based schemes, which face regulatory and legal uncertainties [46]. Penalty-driven schemes, which discourage proactive sharing [47]. Reputation mechanisms, widely used in peer-to-peer and blockchain networks to foster trust [33], [35], [48], and have been integrated with blockchain systems to enhance trustworthiness and incentivize data sharing [21]–[24]. Critically, none integrate *data quality verification* with incentive allocation—essential for high-value cohort data.

TABLE 2
Description of notations

Symbols	Description
DC	Data center for medical cohort data
RV,DV,SV	Reputation value, Data volume, Shared data volume
BDC, SDC	Big scale DC and small scale DC
PB,SC	Permissioned blockchain and Smart contract
PoR	Proof of reputation consensus
SASP	Secure auditable share protocol
RPSM	Reputation-driven proactive sharing mechanism
p, q, λ	Large prime q of length λ and $p = 2q + 1$ be a safe prime
$\mathbb{Z}_p^*, \mathbb{Z}_q^*$	The multiplicative groups of integers modulo p and q
$\mathbb{G}, g$	A multiplicative subgroup of $\mathbb{Z}_p^*$ with prime order q and its generator
$H()$	Hash function mapping $\{0,1\}^*$ to $\mathbb{Z}_q^*$
$Enc(), Dec()$	Encryption and decryption algorithms
$Sig(), Ver()$	Digital signature and verify algorithms
$DC_{\mathbf{i}}, sk_i, PK_i$	The DC uniquely identified by **i** and its public-private key pair
D, D_{sk}, D_{PK}	The cohort data D and its public-private key pair
B_i, C_i, T_i	The i-th block of D, its ciphertext and its tag
$\{B\}, \{C\}, \{T\}$	The set of B_i, C_i, T_i
Obj_D, TH_D	The generated ID of D ant its access threshold
R_{id}^t	Reputation value of DC_{id} at round t
N^t, nm^t	Number of current DCs and committee members at round t
TH_{com}^t	Threshold for becoming a committee at round t
α, β, σ	Penalty factor, decay factor and repetition rate
$\mathcal{TagList}$	Shared data tags set stored on-chain
$\mathcal{Policy}$	Access thresholds set stored on-chain
$\vert D\vert$,$\vert\mathbb{G}\vert$	Length of shared data D and element in $\mathbb{G}$
H	Evaluation of hash function
E	Evaluation of exponent operation
P	Evaluation of bilinear mapping operation

2.3 Security Threats

Models confront two threat levels at the network and architectural levels. **Network-level Attacks** compromise system availability through Distributed Denial of Service attacks [49], mitigated by rate limiting with request throttling [35]; and Collusion attacks [50], addressed via incentive mechanisms [24], [34], [43]. **Blockchain-specific Attacks** threaten data integrity via Replay attacks [51], prevented by timestamp-based nonce verification [14], [17]; and Sybil attacks [52], countered through identity generation challenges [43]. Critically, existing defenses lack *auditing capabilities for shared data*, allowing risks like duplicated/invalid ciphertexts to persist undetected. Concurrently, incentive mechanisms remain vulnerable to *free-riding behaviors*, further discouraging high-value data sharing.

A comparative summary is in Table 1 Unlike cloud-focused schemes, we address sensitive cohort data via distributed storage, restricted access, and incentives. Performance analysis (Section 7) confirms acceptable overhead.

3 PRELIMINARIES

In this section, we briefly introduce the formal mathematical assumptions and cryptographic foundations essential for our model's security guarantees (Section 6) and operational mechanisms (Section 5). Table 2 standardizes notation.

3.1 Assumptions

3.1.1 Decisional Diffie-Hellman (DDH)

Let $\mathbb{G}$ be a multiplicative subgroup of $\mathbb{Z}_p^*$ with prime order q and g be a generator of $\mathbb{G}$. For any Probabilistic Polynomial Time (PPT) adversary $\mathcal{A}$, the DDH assumption [53] is: given $g, g^x, g^y, g^z \in \mathbb{G}$ where $x, y, z \in$

$\mathbb{Z}_q^*$, $|\Pr[\mathcal{A}(g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(g, g^x, g^y, g^{xy}) = 1]| \leq negl(\lambda)$ is negligible in the security parameter λ. That is, the tuples (g, g^x, g^y, g^z) and (g, g^x, g^y, g^{xy}) are computationally indistinguishable.

3.1.2 IND-CPA Security under DDH

Building upon the DDH assumption, we define IND-CPA security for our encryption scheme:

Security Game:

1) $\mathcal{C}$ runs $\mathsf{Init}() \rightarrow Para = (p, q, \lambda, g, H)$, sends to $\mathcal{A}$
2) $\mathcal{A}$ chooses target DC_{id}^*
3) $\mathcal{C}$ computes $(sk^*, PK^*) \leftarrow \mathsf{DC-KeyGen}(1^\lambda, index^*)$
4) $\mathcal{A}$ submits m_0, m_1 with $|m_0| = |m_1|$
5) $\mathcal{C}$ picks $b \xleftarrow{\$} \{0,1\}$, computes $C^* \leftarrow \mathsf{Enc}(PK^*, m_b)$
6) $\mathcal{A}$ makes adaptive oracle queries (excluding challenge ciphertext)
7) $\mathcal{A}$ outputs b', wins if $b' = b$

Definition (IND-CPA Security): *Our encryption scheme is IND-CPA secure under the DDH assumption if for all PPT adversaries* $\mathcal{A}$,

$$\left|\Pr[b' = b] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$$

3.1.3 Data Center Thirsty and Restrictive (DCTR)

Let $\{DC_i\}$ denote a set of data centers with heterogeneous data volumes. For any DC_i possessing data volume $DV_i \in \mathbb{Z}$, the DCTR assumption is that DC_i desires to increase DV_i, potentially limit the increase of other DCs' DVs, the fastest way to increase DV_i is to access other DCs' shared data.

3.2 Locality-Sensitive Hashing

Given a distance metric d, e.g. Euclidean distance, a Locality-Sensitive Hashing (LSH) function hashes close items to the same hash value with higher probability than the items that are far apart. We use the common p-stable LSH function [32] to map the plaintext block to an integer, described as $\mathsf{LSH}(\cdot) \rightarrow v \in \mathbb{Z}_p^*$, where, $\mathsf{LSH}(\cdot)$ is (r, p)-sensitive if any two points s, t satisfy:

- If $d(s,t) \leq r$, then $\Pr[\mathsf{LSH}(s) = \mathsf{LSH}(t)] \geq p$
- If $d(s,t) \geq r$, then $\Pr[\mathsf{LSH}(s) = \mathsf{LSH}(t)] \leq p$

where $d(s,t)$ is the distance between the point s and t.

3.3 Bloom filter

A Bloom filter [54] is an m-bit array initialized with all bits set to 0. Given a set S, the Bloom filter uses k independent hash functions to insert the i-th element $s_i \in S$ by setting all corresponding hash positions in the array to 1. If all queried positions are 1, the element s is considered to exist in the filter with a false positive probability approximated by $\Pr_{\mathrm{FP}} \approx \left(1 - e^{-\frac{k|S|}{m}}\right)^k$, where $|S|$ denotes the cardinality of set S. The filter provides an efficient membership test operation $\mathsf{Ins}(\cdot) \rightarrow \{0,1\}$, where 1 indicates probabilistic membership and 0 guarantees non-membership.

3.4 ElGamal Cryptosystem

The ElGamal scheme [31] is a discrete logarithm-based public key cryptosystem and signature protocol. The algorithm comprises the following components:

- $\mathsf{KeyGen}(1^\lambda) \rightarrow (PK, sk)$. On input security parameter λ, an element $sk \in \mathbb{Z}_q^*$ is randomly selected as the private key and public key $PK \equiv g^{sk}$ are computed. Publish PK.
- $\mathsf{Enc}(PK, m) \rightarrow c$. With the input of a message m and PK, an element $\rho \in \mathbb{Z}_q^*$ is randomly selected, and calculate $y_1 \equiv g^\rho, \quad y_2 \equiv m \cdot PK^\rho$, then output the ciphertext $c := (y_1, y_2)$.
- $\mathsf{Dec}(sk, c) \rightarrow m$. With the input of sk and ciphertext c , the message m is outputed by calculating the $m' \equiv y_2 \cdot y_1^{-sk} \equiv m \cdot PK^\rho / g^{\rho \cdot sk}$.
- $\mathsf{Sig}(sk, m) \rightarrow \pi$. With the input of sk and m, an element $k \in \mathbb{Z}_q^*$ is randomly selected, and calculate $r \equiv g^k, \quad s \equiv k^{-1}(H(m) - sk \cdot r) \mod q$. Output $\pi := (r, s)$.
- $\mathsf{Ver}(m, \pi) \rightarrow 1/0$. With the input of m and sign π, the verification result "1" or "0" is outputed by calculating whether $g^{H(m)} \overset{?}{\equiv} PK^r \cdot r^s$.

Note that we will denote $a \equiv b$ implies $a \equiv b \mod p$ to omit the $\mod p$ operation to reduce clutter.

4 PROBLEM FORMULATION

Building upon the background from Section 2 and the cryptographic foundations from Section 3, this section formally establishes our problem formulation through presenting the system model and design goals, defines adversarial capabilities within our threat model, articulates six formal security definitions that govern our security proofs in Section 6, introduces the SASP and RPSM core components whose detailed implementations are elaborated in Section 5.

4.1 Model Overview and Design Goals

4.1.1 Model Overview

As illustrated in Fig. 1, the model consists of four major entities:

- **Data Center (DC)** is the centralized cohort data centre and owns the cohort data, categorized into Big Volume DCs (BDCs) and Small Volume DCs (SDCs) based on data volume. DCs are evaluated by Reputation Values (RVs), which determine their roles and access privileges.
- **Permissioned Blockchain (PB)** is a blockchain where any DC needs permission to join. Participating DCs can communicate securely through "channels", while the formed P2P network can facilitate data transmission using the gRPC[1] protocol. The blockchain stores sharing tags and policies, ensuring immutability and traceability.
- **Committee** is a dynamic group Com of high-RV DCs selected via the Proof of Reputation (PoR) consensus in each round. Committee members enjoy privileged access to shared data, bypassing access policies, while Com entry thresholds (TH_{com}) limit participation.
- **Smart Contract (SC)** is the autonomous executables deployed on the blockchain , with the Audit SC performing

[1] https://github.com/grpc/grpc-go.git

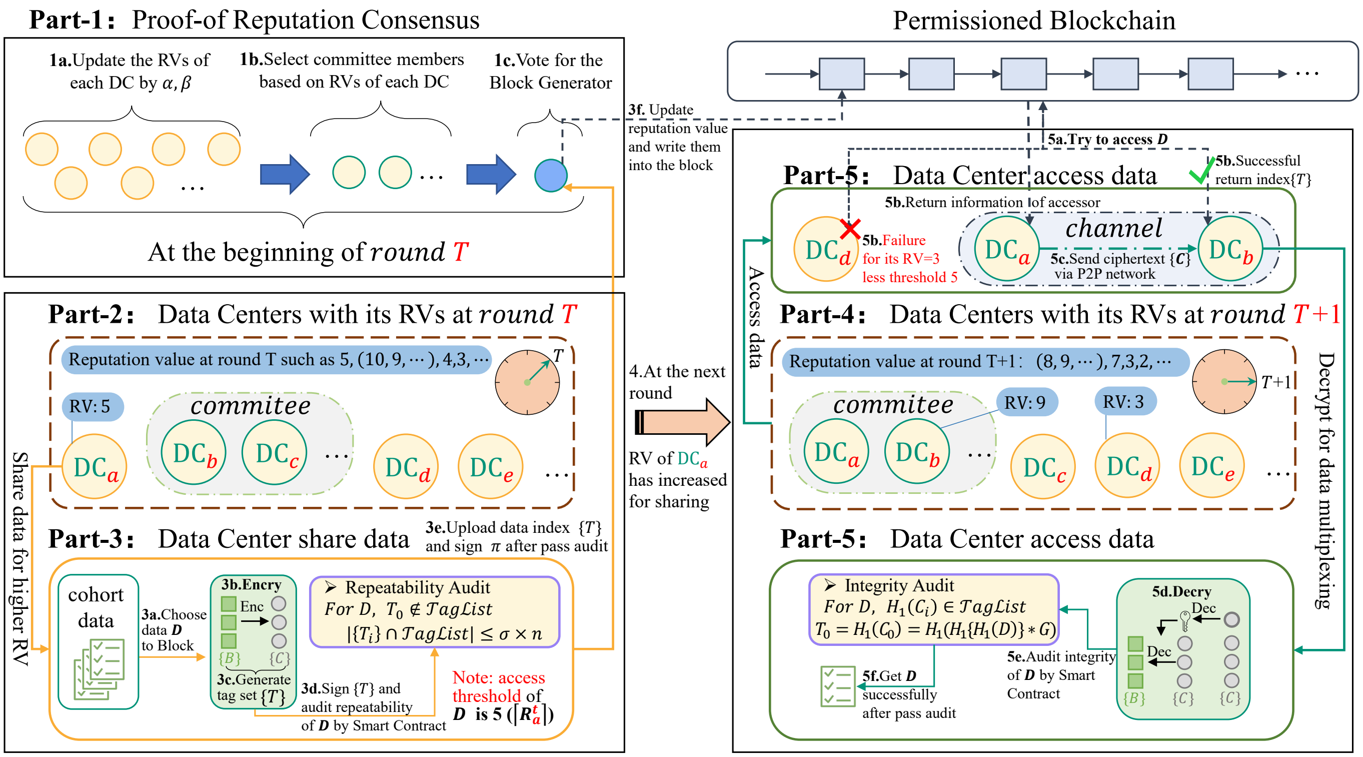
Part-1: Proof-of Reputation Consensus
1a.Update the RVs of each DC by α, β
1b.Select committee members based on RVs of each DC
1c.Vote for the Block Generator
At the beginning of round T
Permissioned Blockchain
3f. Update reputation value and write them into the block
5a.Try to access D
Part-5: Data Center access data
5b.Successful return index{T}
5b.Return information of accessor
DCd
5b.Failure for its RV=3 less threshold 5
DCa
channel
5c.Send ciphertext {C} via P2P network
DCb
Access data
Part-2: Data Centers with its RVs at round T
Reputation value at round T such as 5, (10, 9, ⋯), 4,3, ⋯
T
RV: 5
commitee
DCa
DCb
DCc
DCd
DCe
4.At the next round
RV of DCa has increased for sharing
Part-4: Data Centers with its RVs at round T+1
Reputation value at round T+1: (8, 9, ⋯), 7,3,2, ⋯
T+1
RV: 9
RV: 3
commitee
DCa
DCb
DCc
DCd
DCe
Decrypt for data multiplexing
Share data for higher RV
Part-3: Data Center share data
3e.Upload data index {T} and sign π after pass audit
cohort data
3a.Choose data D to Block
3b.Encry
Enc
{B}
{C}
3c.Generate tag set {T}
Repeatability Audit
For D, T0 ∉ TagList
|{Ti} ∩ TagList| ≤ σ × n
3d.Sign {T} and audit repeatability of D by Smart Contract
Note: access threshold of D is 5 (⌈R_a^t⌉)
Part-5: Data Center access data
Integrity Audit
For D, H1(Ci) ∈ TagList
T0 = H1(C0) = H1(H1{H1(D)} * G)
5f.Get D successfully after pass audit
5e.Audit integrity of D by Smart Contract
5d.Decry
Dec
Dec
{B}
{C}
{C}

Fig. 1. Reputation-backed auditable cohort data sharing model via permissioned blockchain. This figure illustrates that Parts 1/2/4 demonstrate the RPSM incentivizing DCs through dynamic reputation values, while Parts 3/5 implement the SASP via an enhanced ElGamal cryptographic scheme.

integrity and deduplication checks during data sharing and access phases.

As shown in Fig. 1, the proposed model consists of five parts. In Part-1, by the PoR consensus, all RVs are updated, and committee members and a block generator are elected based on RVs. In Part-2, DCs' RVs are determined for this round, reflecting contributions from previous rounds. Committee privilege can be accessed without policy restrictions, while two-factor decay prevents dominance and promotes equity of opportunity. In Part-3, SASP enables secure data sharing through an improved ElGamal algorithm, combined with tags $\{T\}$ for deduplication and integrity auditing. Upon successful auditing with the Audit SC, $\{T\}$ and *policy* are recorded on-chain, completing the sharing process and rewarding the DC with increased RV. Part-4 dynamically updates RVs based on contributions at the next round, rewarding active DCs and maintaining system fairness. In Part-5, committee members bypass policy checks if their RV exceeds the access threshold TH_D, while non-committee members must satisfy both the threshold and policy requirements. Upon validation, encrypted data $\{C\}$ is shared via an off-chain P2P network, Upon successful auditing with the Audit SC indicates completion of data access.

4.1.2 Design Goals

Our model aims to achieve the following design goals:

G1: Active Sharing. [24] The model must guarantee that the real-world participation motivation of Data Centers (DCs) seamlessly aligns with our feasibility assumptions, fostering active engagement in the practice of data sharing.

G2: Stability and Fairness. [34] The model must ensure a consistent reputation evaluation system across all DCs, and the reputation values stabilize to a constant after multiple rounds.

G3: Repetition and Integrity Auditing. [20] The model should guarantee the preservation of each shared data's uniqueness and the assurance of consistency in the accessed data.

G4: Security and Privacy Protection. [22] The model should ensure that shared data cannot be inferred from published on-chain data and allow shared data to be decrypted solely by eligible requesters through a cryptographic signature scheme.

G5: Low Performance Overhead. [44] The model should minimize the overhead of data-sharing procedures, including data encryption and decryption, storage, and communication between data owners and requesters.

G1 to **G4** will be proven in detail in the Section 6 and 8, while **G5** will be proven in detail in the Section 7.

4.2 Threat Model and Adversarial Behavior

We define three adversarial roles within our permissioned blockchain environment, where Data Centers exhibit varying trust levels:

- Fully Honest DCs : Adhere strictly to protocol specifications with no deviation from prescribed procedures. These entities maintain complete data confidentiality and execute all cryptographic operations correctly.
- Semi-Honest DCs (*Passive Adversaries*) : Adhere to protocols while attempting to infer private information from public blockchian records.

- Malicious DCs (*Active Adversaries*) : Engage in systematic adversarial behaviors across three attack surfaces aimed at undermining the confidentiality, robustness or fairness of the data sharing process:
  - **Data Layer Attacks**: such as (a).Submit duplicate/incomplete/falsified ciphertext to distort auditing results. (b).Send incomplete decryption keys to disrupt data access. (c).Conduct chosen-plaintext attacks on shared data to breach data confidentiality.
  - **Network Layer Attacks**: (a).Mount Distributed Denial of Service attacks [49] on critical protocol phases. (b).Execute Collusion attacks [50] to manipulate reputation scores. (c).Perform Replay attacks [51] on consensus messages. (d).Launch Sybil attacks [52] with forged identities.
  - **Model Honesty Attacks**: Attempt to exploit the reputation-driven proactive sharing mechanism by launching the multiple attacks described above to distort model honesty and, ultimately, compromise model fairness and stability, thereby breaking the DCs' motivation to actively participate in sharing.

Our model incorporates mechanisms to address these threats within the constraints of a permissioned blockchain environment. We *assume* that PB's CA authentication and permissioned consensus inherently limit susceptibility. While the current focus is on ensuring model security and privacy, robustness, and fairness, further exploration into susceptibility against advanced attacks is outlined as part of our future work. Security analyses in Section 6 and practical evaluations in Section 8 demonstrate the model's resilience against these identified threats in data-sharing scenarios.

4.3 Formal Security Definitions

To ensure our system meets the aforementioned design goals (**G1** to **G4**), we define five additional formal security definitions (complementing the IND-CPA security defined in Section 3), where adversaries' capabilities align with the threat model in Section 4.2 and their corresponding security proofs are provided in Section 6.

4.3.1 *Data Privacy*

Adversarial Capabilities:

- Full access to $\{T_i\}, \pi, policy, PK$ on-chain.
- Intercepts off-chain ciphertext $\{C\}$.
- Knows statistical patterns of data sharing.

Definition 2 (Data Privacy): *The model satisfies data privacy if for all PPT adversaries $\mathcal{A}$ and any shared data D,*

$$\Pr[\mathcal{A}(\{T\}, PK) = B_i] \leq \mathsf{negl}(\lambda) \quad \forall i$$

and

$$|\Pr[\mathcal{A}(PK) = \mathsf{attr}(DC)] - \Pr[\mathsf{random}]| \leq \mathsf{negl}(\lambda)$$

4.3.2 *Reputation Stability*

Definition 3 ((α, β)-Stability): *Let R_i^t denote the reputation of honest DC$_i$ at epoch t. The model is (α, β)-stable if*

$$R_{\min} \leq R_i^t \leq R_{\max} \quad \text{where} \quad \begin{cases} R_{\max} = \dfrac{DV_{\max}}{1-\beta} \\ R_{\min} = 0 \end{cases}$$

with $DV_{\max}$ as the maximum data volume per epoch.

4.3.3 *Fairness*

Definition 4 (δ-Fairness): *The model satisfies δ-fairness if for all honest DCs i, j:*

$$\left| \frac{\mathbb{E}[\Delta R_i]}{S_i} - \frac{\mathbb{E}[\Delta R_j]}{S_j} \right| \leq \delta,$$

where ΔR_i is expected reputation change, and S_i is the number of sharing attempts (successful or failed) by DC$_i$ in the epoch.

4.3.4 *Proactive Incentive*

Definition 5 (γ-Proactive Incentive): The model provides γ-proactive incentive if for all DC types:

$$\mathbb{E}[\Delta R|\text{Share}] - \mathbb{E}[\Delta R|\text{NotShare}] \geq \gamma \cdot R_{\max}$$

where $\gamma > 0$ is the minimum reputation incentive ratio, and ΔR is the reputation change per round.

4.3.5 *Honesty Incentive*

Definition 6 (η-Honesty Advantage): The model provides η-honesty advantage if:

$$\lim_{t \to \infty} \frac{\mathbb{E}[R_t^{\text{honest}}]}{\mathbb{E}[R_t^{\text{dishonest}}]} \geq 1 + \eta$$

for some $\eta > 0$, where R_t is the reputation at round t.

4.4 Model Core Components

To fulfill these security goals, we design two core components: SASP and RPSM for our model.

4.4.1 Secure Auditable Sharing Protocol

Based on the cryptosystem building on the ElGamal algorithm and Locality-Sensitive Hashing (LSH) function introduced in Section 3, our security auditable sharing protocol at the block level consists of the following seven functions (Init, KeyGen, Enc, Dec, TagGen, Sig, Ver) and three interactive actions ($Share, Audit, Access$).

- Init() → $Para$: Returns system parameters $Para = (p, q, \lambda, g, H)$.
- KeyGen$(1^\lambda, index) \to (sk, PK)$: With the input of a security parameter λ, an element $index \in \mathbb{Z}_q^*$ returns the key pair (sk, PK).
  - DC-KeyGen$(1^\lambda, index) \to (sk, PK)$: For data center DC_{id} characterized by $index$, compute: $sk \leftarrow H(index)$, $PK \equiv g^{sk}$.
  - B-KeyGen$(1^\lambda, index) \to (sk, PK)$: For data $D := B_1||B_2|| \cdots ||B_n$, the i-th data block B_i with $index_i \leftarrow$ LSH(B_i), compute: $sk \leftarrow H(index_i)$, $PK \equiv g^{sk}$.
  - D-KeyGen$(1^\lambda, index) \to (sk, PK)$: For full data $D := B_1 \parallel \cdots \parallel B_n$, compute: $index \leftarrow H(index_1 \parallel \cdots \parallel index_n)$, $sk \leftarrow H(index)$, $PK \equiv g^{sk}$.
- Enc$(PK, m) \to C$: With the input of PK and a message m and output the ciphertext C.
- Dec$(sk, C) \to m$: With the input of sk and the ciphertext C and output the message m.
- TagGen$(D, C) \to \{T\}$: With the input of the data $D := B_1||B_2|| \cdots ||B_n$ and corresponding ciphertext block $C := C_1||C_2|| \cdots ||C_n$, output the tag set $\{T_i\}$ where $i \in [0, n]$.
- Sig$(sk, m) \to \pi$: With the input of sk and a message m and output the sign π.

- $\mathsf{Ver}(m, \pi) \to 1/0$: With the input of m and the sign π and output the verification result $1/0$.
- $Share(\{T\}, \pi, policy) \to Obj/0$: With the input of tag set $\{T\}$ of data to be shared, corresponding signatures π and an access policy $policy$ by DC_{id}, the PB returns shared result Obj or 0 to DC_{id} after determining audit results, and updates the DC_{id}'s RV based on the results.
- $Audit(\{T\}, n) \to 1/0$: With the input of tag set $\{T\}$ and the size n of the set, PB performs repetition or integrity audits by SC and the on-chain records $\mathcal{TagList}$ and returns audit results $1/0$.
  - $\mathsf{Verf}_R(\{T\}, n) \to 1/0$: Input $\forall T_i \in \{T\}$ to audit repeatability, considering the probabilistic error of the Bloom filter, SC returns the 0 *iff* the number of times that $\mathsf{Ins}(T_i) = 0$ is over $\sigma \cdot n$ (σ is the repetition rate).
  - $\mathsf{Verf}_I(T'_0, 1) \to 1/0$: Input the tag T'_0 of the obtained ciphertext to audit integrity, and SC returns the 1 *iff* that $T'_0 == T_0 \in \mathcal{TagList}$.
- $Access(PK, Obj_{id}) \to \{C\}/0$: With the input of PK and the data ID Obj_{id} to be accessed, returns the corresponding ciphertext set $\{C\}$ *iff* it passes the $policy$ and access thresholds TH_{id}.

4.4.2 Reputation-driven Proactive Sharing Mechanism

Our reputation-driven proactive sharing mechanism works in rounds, and the time interval for each round is fixed.

- Init RVs: At round t, let $\{DC\}^t := \{DC_1, \cdots, DC_{N^t}\}$ denote the set of N^t data centers in the channel. Each DC_i has a reputation value R_i^t maintained by the blockchain. Newly joined DCs initialize with $R_{\text{new}}^t := 0$.
- Invoke RVs: When DC_i successfully shares object Obj_{id}, the access threshold TH_{id} is set as $TH_{id} := \lceil R_i^t \rceil$, access SC $access()$ can be invoked *iff* DC_j satisfies $R_j^t \geq TH_{id}$.
- Update RVs: The reputation value R_i^t of DC_i will be increased upon successfully sharing and being accessed, but decreased upon failed sharing and access with the penalty factor α, then fixed decay β at the start of each round.
- Form Committee: Within a specific time period, part of DCs over TH_{com}^t were selected to form the committee by our PoR consensus. Note that we require the committee to represent the vast majority; therefore, similar to [55], the committee size in each round is to be no more than $nm^t := \lceil \log_2 N^t \rceil$.
- Record RVs: Finally, a block generator (selected via PoR) writes the R^t, the TH_{com}^{t+1}, and nm^t committee members into the blockchain at the t-th round end. System state satisfies: $\underbrace{1}_{\text{generator}} + \underbrace{(nm^t - 1)}_{\text{committee}} + \underbrace{(N^t - nm^t)}_{\text{normal DCs}} \equiv N^t$.

5 Model Design

To fulfill these design goals (**G1-G5**) from Section 4.1.2, this section details the technical realization of our integrated model. Section 5.1 presents SASP's three-phase workflow derived from the five parts in Fig. 1 and illustrated in Fig. 2: Initialization (cryptographic setup), Data sharing (encryption/tag generation), and Data access (integrity verification). Section 5.2 designs RPSM's reputation mechanics with two-factor reputation updates, committee privileges, and our PoR consensus.

5.1 Our secure auditable sharing protocol

5.1.1 Parameter Initialization - Init() $\to Para$

In Step **a1** in Fig. 2, the Certificate Authority (CA) of PB chooses a large prime q of length λ and the safe prime $p = 2q + 1$, then generates a cyclic multiplicative subgroup $\mathbb{G} \subset \mathbb{Z}_p^*$ of q where g is a generator of $\mathbb{G}$, and a hash function $H()$, where $H : \{0,1\}^* \to \mathbb{Z}_q^*$. Public parameters $Para = (p, q, \lambda, g, H)$ are written to the blockchain. PB maintains two immutable on-chain records $\mathcal{TagList}, \mathcal{Policy}$, which are used to store shared data tags and corresponding access thresholds, to support repetition and integrity auditing of shared data to ensure the quality.

5.1.2 DC Initialization - KeyGen$(1^\lambda, index) \to (sk, PK)$

In Steps **a2-4**, for new data center DC_a, it is characterized by CA in three aspects: its name $name$, located address $addr$ and its coding of the unified credit identifier $code$, CA computes anti-Sybil index: $index \equiv H(name \parallel addr \parallel code)$. By randomly choosing $rv \in {Z_q}^*$, CA runs DC-KeyGen to generate DC_a's key pair (sk_a, PK_a), where $sk_a \equiv index \cdot rv \mod q$, $PK_a \equiv g^{sk_a}$, then to publish the PK_a as the id of DC_a and to initialize reputation value $R_a^t \leftarrow 0$.

5.1.3 Data Sharing and Deduplication Filtering

In Steps **b1-8**, when a data center DC_b wants to share the plaintext data which divided as $D := B_1||B_2|| \cdots ||B_n$, it performs the following:

The Step **b1** is a simple data quality validation before calling the smart contract $Share(\cdot)$, that is, for each block $B_\mathbf{i}$, the corresponding index $index_\mathsf{j} := \mathsf{LSH}(B_\mathrm{i})$ is satisfied as $\mathbf{j} == \mathbf{i}$, which avoids duplication of content in the shared data itself for **G3**.

The Steps **b2-3** generate the key pair.

- Block keys generated by B-KeyGen: $sk_{B_i} \equiv H(index_i)$, $PK_{B_i} \equiv g^{sk_{B_i}}$.
- Data keys generated by D-KeyGen: $sk_D \equiv H(index_1|| \cdots ||index_n)$, $PK_D \equiv g^{sk_D}$.

The step **b4** has three actions. The first is to encrypt block-level data for **G4**. For the i-th block B_i, according to the Enc with the randomly choosing $r_i \in {Z_q}^*$, the corresponding ciphertext C_i is computed as follows:

$$C_i \equiv (g^{r_i} \parallel B_i \cdot PK_D^{r_i}), \ where\ i \in [1, n]. \tag{1}$$

The second is to generate the set of shared data D's tags for **G3**. Considering the ciphertext storage overhead, we map it first. By TagGen, with plaintext D and corresponding ciphertexts $\{C\}$, we have

$$\begin{cases} T_i & \equiv & PK_{B_i} \\ T_0 & \equiv & H(H(C_1)|| \cdots ||H(C_n)) \end{cases}, where\ i \in [1, n]. \tag{2}$$

The final step signs the tags $\{T\}$ of shared data D's for **G4**. In order to validate data faster and more securely, and to ensure the validity of decryption, considering that a signature uniquely corresponds to a piece of data, we use the secret key sk_D of D and the secret key sk_b of DC_b, which first generate $k := H(sk_D)$, and the signature π of $\{T\}$ is generated using Sig as follows:

$$\pi \equiv \left(g^k \ , \ \left(H(\{T\}) - sk_b \cdot g^k \right) \cdot k^{-1} \mod q \right). \tag{3}$$

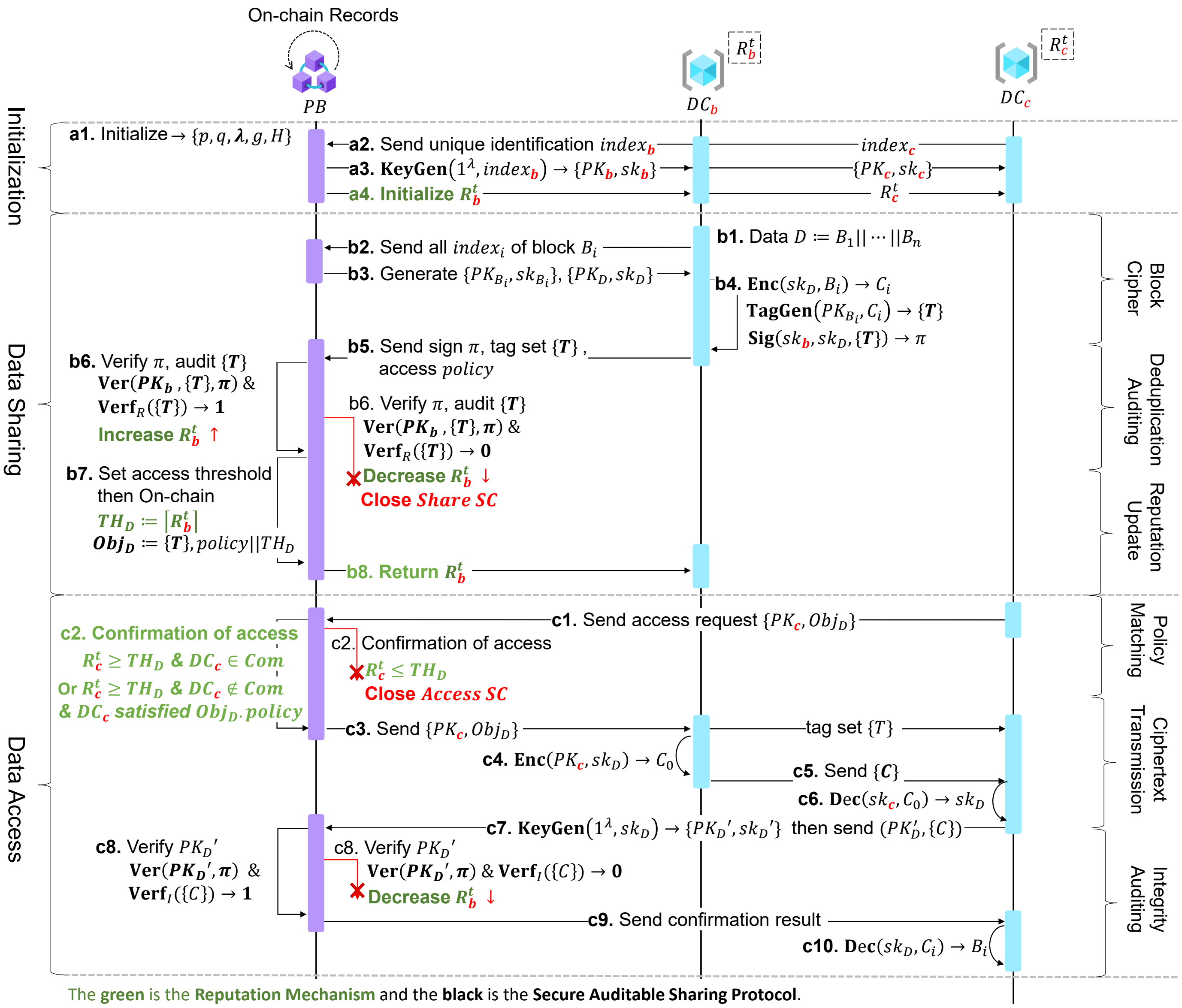
On-chain Records
PB
DC_b
R_b^t
DC_c
R_c^t
Initialization
Data Sharing
Data Access
a1. Initialize → $\{p, q, \lambda, g, H\}$
a2. Send unique identification $index_b$
$index_c$
a3. $\mathbf{KeyGen}(1^\lambda, index_b) \to \{PK_b, sk_b\}$
$\{PK_c, sk_c\}$
a4. Initialize R_b^t
R_c^t
b1. Data $D := B_1 || \cdots || B_n$
b2. Send all $index_i$ of block B_i
b3. Generate $\{PK_{B_i}, sk_{B_i}\}, \{PK_D, sk_D\}$
b4. $\mathbf{Enc}(sk_D, B_i) \to C_i$
$\mathbf{TagGen}(PK_{B_i}, C_i) \to \{\boldsymbol{T}\}$
$\mathbf{Sig}(sk_b, sk_D, \{\boldsymbol{T}\}) \to \pi$
b5. Send sign π, tag set $\{\boldsymbol{T}\}$, access $policy$
b6. Verify π, audit $\{\boldsymbol{T}\}$
$\mathbf{Ver}(\boldsymbol{PK_b}, \{\boldsymbol{T}\}, \boldsymbol{\pi})$ &
$\mathbf{Verf}_R(\{\boldsymbol{T}\}) \to \mathbf{1}$
Increase R_b^t ↑
b6. Verify π, audit $\{\boldsymbol{T}\}$
$\mathbf{Ver}(\boldsymbol{PK_b}, \{\boldsymbol{T}\}, \boldsymbol{\pi})$ &
$\mathbf{Verf}_R(\{\boldsymbol{T}\}) \to \mathbf{0}$
✘Decrease R_b^t ↓
Close $Share$ SC
b7. Set access threshold then On-chain
$TH_D := \lceil R_b^t \rceil$
$\boldsymbol{Obj_D} := \{\boldsymbol{T}\}, policy || TH_D$
b8. Return R_b^t
c1. Send access request $\{PK_c, Obj_D\}$
c2. Confirmation of access
$R_c^t \geq TH_D$ & $DC_c \in Com$
Or $R_c^t \geq TH_D$ & $DC_c \notin Com$
& DC_c satisfied $Obj_D.policy$
c2. Confirmation of access
✘$R_c^t \leq TH_D$
Close $Access$ SC
c3. Send $\{PK_c, Obj_D\}$
tag set $\{T\}$
c4. $\mathbf{Enc}(PK_c, sk_D) \to C_0$
c5. Send $\{\boldsymbol{C}\}$
c6. $\mathbf{Dec}(sk_c, C_0) \to sk_D$
c7. $\mathbf{KeyGen}(1^\lambda, sk_D) \to \{PK_D', sk_D'\}$ then send $(PK_D', \{C\})$
c8. Verify PK_D'
$\mathbf{Ver}(\boldsymbol{PK_D}', \boldsymbol{\pi})$ &
$\mathbf{Verf}_I(\{C\}) \to \mathbf{1}$
c8. Verify PK_D'
$\mathbf{Ver}(\boldsymbol{PK_D}', \boldsymbol{\pi})$ & $\mathbf{Verf}_I(\{C\}) \to \mathbf{0}$
✘Decrease R_b^t ↓
c9. Send confirmation result
c10. $\mathbf{Dec}(sk_D, C_i) \to B_i$
Block Cipher
Deduplication Auditing
Reputation Update
Policy Matching
Ciphertext Transmission
Integrity Auditing
The green is the Reputation Mechanism and the black is the Secure Auditable Sharing Protocol.

Fig. 2. Three-phase workflow of SASP integrated with the RPSM mechanism in our model. This figure demonstrates three-phase interactions (initialization, sharing, access) between DC_b/DC_c and PB: SASP ensures secure data transmission through cryptographic protection, while RPSM collaborates with audit contracts to govern data quality control.

The Steps **b5-6** are to call the smart contract $Share(\cdot)$ for sharing. A policy is the core of realizing security cross-DCs data access [56]. However, the design of the policy is not crucial in guaranteeing data circulation. So, we default to the policy that is valid. By Ver, CA verifies the correctness of the signature π and tag set $\{T\}$ using the following equation:

$$g^{(H(\{T\})} \overset{?}{\equiv} PK_b{}^{g^k} \cdot (g^k)^{\left(H(\{T\}) - sk_b \cdot g^k\right)/k}. \tag{4}$$

If the above equation holds, CA executes $\mathsf{Verf}_R(\{T\}, n)$ to complete the repetition audit. For plaintext blocks B, B', by Equations (1) and (2), different contents can get different secret keys sk_B, sk'_B in non-negligible probability by Ins, i.e. different T, T'. So the concrete repetition audit contract with time complexity of $O(n)$ is implemented as Algorithm 1, which is for **G5**. Considering the probabilistic error of the Ins, by setting a repetition factor $\sigma \in (0, 1)$, the filtering result 1 for deduplication is as follows:

$$\Pr[\mathsf{Ins}(T_i) = 0] \leq \sigma, \ \forall i \in [0, n]. \tag{5}$$

After passing the Verf_R, the Steps **b7-8** store the tag set and access threshold. As the on-chain data is stored as $key : value$, the set of these tags is stored as $Obj_D : \{T\}$ which $Obj_D := \pi$, $\mathcal{TagList} := \{T\} \cup \mathcal{TagList}$. Then, for **G1** and **G2**, with the $policy$ determined by DC_b and necessary access threshold TH_D, which is determined by R_b^t. The access policy is stored as $Obj_D : \{policy \mid\mid TH_D\}$, and $\mathcal{Policy} := \{policy \mid\mid TH_D\} \cup \mathcal{Policy}$. It means that D is shared successfully by DC_b. For each success successful share, current R_b^t increased as follows:

$$R_b^t \leftarrow R_b^t + 1. \tag{6}$$

But if DC_b does not pass the Verf_R, sharing will not be allowed again in this round to prevent possible DDoS attacks.

Algorithm 1: $AuditSC$: SC for Repetition Audit.

```
Data: Ciphertext tag set $\{T\}$
$\{\mathcal{T}ag\mathcal{L}ist\} \leftarrow PB.Obj_D.\{\mathcal{T}ag\mathcal{L}ist\}$;
begin
    // Repetition Audit at Data sharing phase
    // $DC_b$ shares $Obj_D$ with tag $\{T\}$
    if $RA(\{T\})$ then                                  // For Repetition
        $\mathcal{T}ag\mathcal{L}ist \leftarrow \mathcal{T}ag\mathcal{L}ist \cup Obj_D$;          // On-chain
        $R_b{}^t \leftarrow R_b{}^t + 1$;                      // Add the RV of $DC_b$
    else
        $R_b{}^t \leftarrow \mathbf{max}(0, R_b{}^t - \mathbf{max}(1, \alpha * R_b{}^t))$;
                                            // Reduce the RV of $DC_b$
    end
end
// RA: For Repetition Audit
function $RA(\{T\})$
    $\sigma \leftarrow 0.2$;                 // Initialize repetition rate
    if $T_0 \notin \mathcal{T}ag\mathcal{L}ist$ then
        $count \leftarrow 0$;                           // Number of repeats
        foreach $T_i \in \{T\}$ where $i \in [1, N^t]$ do
            if $T_i \in \mathcal{T}ag\mathcal{L}ist$ then
                $count \leftarrow count + 1$;
                if $count > \sigma * N^t$ then
                    return $False$
                end
            end
        end
        return $True$
    else
        return $False$
    end
```

And for each failed sharing, the current R_b^t decreased with a penalty factor α as follows:

$$R_b{}^t \leftarrow max\{0, R_b{}^t - max\left(1, \alpha * R_b{}^t\right)\}. \tag{7}$$

5.1.4 Data Access and Integrity Audit

As in Steps **c1-10**, when data center DC_c finds the shared data Obj_D by DC_b and wants to access it, to call the smart contract $Access(\cdot)$ with its PK_c for accessing in Step **c1**.

In Steps **c2-3**, after determining that DC_c is accessing Obj_D for first-time and satisfies $policy$ and $R_c^t \geq TH_D$, for **G5**, by the channel, PB sends the public key PK_c of DC_c and Obj_D to DC_b, and return the tag set $\{T\}$ to DC_c. Note that once DC_c fails due to multiple accesses or an unsatisfied threshold, it will not be able to access any other shared data in this round to prevent possible DDoS attacks.

In Step **c4**, for **G4** and **G5**, by Enc, with the randomly chosen $v \in \mathbb{Z}_q^*$ the secret key sk_D of D is encrypted as C_0 is as follows:

$$C_0 \equiv \; (g^v \mid\mid sk_D \cdot PK_c^v). \tag{8}$$

Then, in Step **c5**, DC_b sends the ciphertext set $\{C_i\}$ to DC_c, where $i \in [0, n]$.

In Steps **c6-10**, for **G3, G4** and **G5**, DC_b sends ciphertexts $\{C_i\}_{i=0}^n$ via P2P to DC_c, then DC_c calculates the $T_0' \equiv H(H(C_1')||\cdots||H(C_n'))$ and sends T_0' to CA to execute the $\mathsf{Verf}_I(T_0', 1)$ for the result of integrity audit. Considering the existence of malicious DCs deliberately submitting inconsistent T_0', DC_b will apply for a recalculation of T_0' by the PB after the audit results to eliminate this situation. For all the ciphertext in set $\{C\}$, by Equation (2), changing any $C_i \in \{C\}$ will result in a completely different tag T_0, so the concrete integrity audit contract with the time complexity of $O(n)$ is implemented as Algorithm 2. After passing the Verf, according to the Dec with sk_c and C_0, the corresponding data secret key sk_D is decrypted as follows.

Algorithm 2: $AuditSC$: SC for Integrity Audit.

```
Data: Ciphertext tag set $\{T\}$
$\{\mathcal{T}ag\mathcal{L}ist\} \leftarrow PB.Obj_D.\{\mathcal{T}ag\mathcal{L}ist\}$;
begin
    /* Integrity Audit at Data sharing phase  */
    // Data auditor $DC_c$ has obtained the
       ciphertext $\{C'\}$ from data sharer $DC_b$
    $T_0' \leftarrow null$;
    foreach $C_i{}' \in \{C'\}$ do                        // $i \in [1, n]$
        $T_0' \leftarrow T_0' \mid\mid C_i'$;
    end
    $T_0{}' \leftarrow H(T_0')$;                       // Generate tag set
    if $IA(T_0{}')$ then                          // For Integrity Audit
        $R_b{}^t \leftarrow R_b{}^t + 1$;                      // Shared completion
    else
        $R_b{}^t \leftarrow \mathbf{max}(0, R_b{}^t - \mathbf{max}(1, \alpha * R_b{}^t))$
    end
end
/* IA: For Integrity Audit                          */
function $IA(T_0')$
    if $T_0' \in \mathcal{T}ag\mathcal{L}ist$ then      // Unique identification
        if $T_0' \neq T_0$ then
            return $False$
        end
        return $True$
    else
        return $False$
    end
```

$$sk_D \equiv \; (sk_D \cdot PK_c^v) \cdot (g^v)^{-sk_c}. \tag{9}$$

Then for **G4**, the decrypted sk_D as the $index$ is inputted into the $\mathsf{D\text{-}KeyGen}$ to get a new key pair (sk_D', PK_D'), where $sk_D' \equiv H(sk_D)$ and $PK_D' \equiv g^{sk_D'}$. Then PK_D' is sent to CA, compared with the π, the authenticity of the secret key sk_D is verified as follows:

$$PK_D' \overset{?}{\equiv} g^k. \tag{10}$$

If the above equation holds, in Step **c10**, for each $C_i \equiv (g^{r_i}, c_i)$, the ciphertext data is decrypted by verifying sk_D as follows:

$$B_i \equiv \; c_i \cdot (g^{r_i})^{-sk_D}, where \; i \in [1, n]. \tag{11}$$

Finally the data $D := B_1||B_2||\cdots||B_n$ is successfully sent from DC_b to DC_c while its privacy is always protected. Actually, if the execution result from CA is 0, the R_b^t decreased via Equation (7).

5.2 Our reputation-driven proactive sharing mechanism

Towards reputation-driven proactive sharing, the core is to associate the data access threshold with the reputation value for **G1**. There is an access threshold TH_i for any shared data Obj_i, and any DC DC_a with $R_a^t \leq TH_i$ have no access to Obj_i. Meanwhile, DC_a shares the data Obj_D, which its $TH_D := \lceil R_b^t \rceil$. In order to achieve this purpose, we designed the protocol as follows:

5.2.1 Two-factor to ensure stability and viability

For **G2**, we designate α as a penalty factor, β as a decay factor, where $\alpha, \beta \in (0, 1)$, to ensure the reputation mechanism is stable and available, where the reputation value is bounded up and down, and the reputation value can be used as a suitable quantitative indicator. At the beginning of each round, each DC's reputation value R_{id} decays as follows:

$$R_{id}^{t+1} \leftarrow R_{id}^{t} * \beta. \tag{12}$$

5.2.2 Committee to incentivize sharing

For **G1**, we designed a special committee Com whose members can ignore the *policy* of shared data Obj_D and only need to satisfy that its Reputation Values (RVs) are greater than TH_D for successful access, which realizes motivation among DCs. We refer to this property as *committee's right*. It means that for $DC_a \in Com$ and $DC_b \notin Com$, for successful accessing the Obj_D, $R_a^t, R_b^t \geq TH_D$ and DC_b needs to additionally satisfy *policy* but DC_a does not.

5.2.3 Consensus to achieve fairness

We propose a Proof-of-Reputation (PoR) consensus as a novel variant of Proof-of-Stake [36], where **reputation value (RV)** replaces stake value as the election weight. This design achieves fair committee elections through three steps:

Step 1 - *Committee member selection.* For N^t DCs, initial RVs are derived from the $(t-1)$-th block in Equation (12). To handle RV heterogeneity, we apply Z-Score normalization [57]:

$$\begin{cases} \widehat{\mu} & := \frac{1}{N^t}\sum_{i=1}^{N^t} R_i{}^t \\ \widehat{\sigma} & := \left(\frac{1}{N^t}\sum_{i=1}^{N^t}\left(R_i{}^t - \widehat{\mu}\right)^2\right)^{\frac{1}{2}} \end{cases} \tag{13}$$

$$\begin{cases} Z_{R_i^t} & := (R_i^t - \widehat{\mu})/\widehat{\sigma} \\ Z_{TH_{com}^t} & := (TH_{com}^t - \widehat{\mu})/\widehat{\sigma} \end{cases} \tag{14}$$

Candidates satisfy $Z_{R^t} \geq \min(2, Z_{TH_{com}^t})$, with top-$nm^t$ ($nm^t := \lceil \log_2 N^t \rceil$) by Z_{R^t} becoming committee members.

Step 2 - *Determine the block generator.* Non-committee $(N^t - nm^t)$ DCs independently vote for generators among members within a certain period of time to avoid the effect of network delay. Similar to [58], the DC with the most votes is the block generator. The rule for the block generator to be selected from the nm^t members is as follows:

$$\arg\max_j \left(Vote_j \cdot \frac{Z_{R_j^t}}{\sum_{k=1}^{nm^t} Z_{R_k^t}} \right), \forall j \in [1, nm^t] \tag{15}$$

where the $Vote_j$ is the vote count for DC_j. Similar to [43], [59], by PoR, only members with higher RVs can become generators with a higher probability, which is a positive way to incentivize DCs to participate in data sharing.

Step 3 - *Block finalization.* The elected generator DC_j sets the next-round threshold $TH_{com}^{t+1} := \lceil R_j^t \rceil$ and writes all RVs, committee members, and TH_{com}^{t+1} into the block.

Our Proof-of-Reputation (PoR) consensus fundamentally differs from existing reputation-based mechanisms in three critical aspects: Unlike Proof-of-Authority (PoA)[2], our Reputation Value (RV) functions as a participation qualification metric rather than a currency, specifically governing block generator selection and threshold determination. Distinct from [43] and [59], we intentionally disregard historical RV through our decay factor β in Equation (12), which exclusively incentivizes sustained data sharing participation. While sharing probabilistic election principles with Proof-of-Stake [36], our consensus activates solely at round initialization for block generation and explicitly binds generator selection to current RV values.

6 Security Analysis

This Section provides formal proofs for all security definitions from Section 4.3 and analyzes attack mitigations referencing threat model (Section 4.2).

6.1 Confidentiality and Privacy of the data sharing

The one-way Hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ we used is proven to satisfy collision resistant [7].

We prove firstly that any data of the data centers satisfy *confidentiality protection*, which means that for any Probabilistic Polynomial Time (PPT), the encryption of two arbitrary data cannot be distinguished in our model; that is, indistinguishability under chosen-plaintext attack (IND-CPA) is satisfied.

Theorem 1. *Our model satisfies IND-CPA security (Definition.1) if the DDH assumption (Sec.3.1.1) holds.*

Proof. Let $\mathcal{A}$ have advantage ϵ. Construct $\mathcal{B}$ against DDH: 1. $\mathcal{B}$ receives (g, g^a, g^b, Z) where $Z = g^{ab}$ or random; 2. Simulate Init to send $Para = (p, q, \lambda, g, H)$; 3. When $\mathcal{A}$ chooses DC_{id}^*, simulate $\mathsf{DC-KeyGen}$ to set $PK^* \leftarrow g^a$; 4. When $\mathcal{A}$ submits m_0, m_1, pick $b \xleftarrow{\$} \{0,1\}$ then compute $C^* \leftarrow (g^b, Z \cdot m_b)$; 5. For $\mathcal{A}$'s oracle queries, $\mathsf{DC-KeyGen}(index)$: $sk \leftarrow H(index)$, $PK \leftarrow g^{sk}$, return (sk, PK), or $\mathsf{Enc}(PK, m)$, pick $r \xleftarrow{\$} \mathbb{Z}_q$, return $(g^r, m \cdot PK^r)$; 6. $\mathcal{A}$ outputs b'; $\mathcal{B}$ outputs 1 if $b' = b$, else 0. If $Z = g^{ab}$, $C^* = (g^b, g^{ab} \cdot m_b) = (g^b, (g^a)^b \cdot m_b)$, so Enc with $PK^* = g^a$, $r = b$, $\mathcal{A}$ sees valid ciphertexts such as

$$\Pr[\mathcal{B} \rightarrow 1] = \Pr[b' = b] = \frac{1}{2} + \epsilon$$

Else $Z \xleftarrow{\$} \mathbb{G}$, $C^* = (g^b, R \cdot m_b)$ is uniformly random, $R \cdot m_b$ uniform in $\mathbb{G}$, no m_b information, then

$$\Pr[\mathcal{B} \rightarrow 1] = \Pr[b' = b] = \frac{1}{2}$$

Thus $\mathcal{B}$'s DDH advantage is:

$$\left| (\frac{1}{2} + \epsilon) - \frac{1}{2} \right| = \epsilon$$

Contradicts DDH assumption if ϵ non-negligible. □

Then we prove that any data of the data centers satisfies *data privacy*, which means that for any PPT adversary, it is impossible to infer the corresponding real information from the information published on-chain.

Theorem 2. *Our model satisfies data privacy (Definition.2) if the $H()$ satisfies collision resistance.*

Proof. **Part 1: Plaintext from tag**. By construction: $T_i = H(\mathsf{LSH}(B_i))$ Since LSH is locality-sensitive and H is collision-resistant:

$$\Pr[\mathcal{A}(T_i) = B_i] \leq \Pr[\mathsf{collision}] \leq \mathsf{negl}(\lambda)$$

[2] https://github.com/paritytech/parity/wiki/Proof-of-Authority-Chains

Part 2: Identity from PK. $PK = g^{H(index)}$ where $index$ combines DCs' attributes (name/addr/code) and random nonce from registration. The DDH assumption implies:

$$\Pr[\mathcal{A}(g^{H(index)}, g^r)] \leq \mathsf{negl}(\lambda) \quad r \xleftarrow{\$} \mathbb{Z}_q$$

Thus $\mathcal{A}$ cannot distinguish PK from random element.
Part 3: Attributes from access patterns. Access policy enforcement is reputation-based on $\mathsf{Access}(PK, Obj) \to \{C\}$ iff $R_{id} \geq TH$. Since TH is set proportional to owner's RV and RVs are updated via public SC, access patterns leak only reputation relations, not raw attributes. □

6.2 Fairness and Stability of the data sharing

We first prove that for any data center DC, its reputation value RV maintains stable bounds under the effect of penalty (α) and decay (β) factors, a property termed *stability*.

Theorem 3. *Our model satisfies (α, β)-Stability (Definition 3).*

Proof. **Part 1: Upper bound ($R_{\max}$).** For DC$_i$ sharing maximum data $DV_{\max}$ every epoch, reputation evolves as:

$$R^t = \beta R^{t-1} + DV_{\max}$$

The closed-form solution and limit yield:

$$R^t = \beta^t R^0 + DV_{\max}\tfrac{1-\beta^t}{1-\beta} \xrightarrow{t\to\infty} \tfrac{DV_{\max}}{1-\beta} = R_{\max}$$

Since $\beta \in (0,1)$, $R^t < R_{\max}$ $\forall t$, establishing strict upper bound.
Part 2: Lower bound (0). By Equations (7),

$$R_b^t \leftarrow \max\left\{0, R_b^t - \max(1, \alpha R_b^t)\right\} \geq 0$$

Bound 0 is reached when: (i) $R^t < 1$ (directly set to 0), (ii) $R^t \in [1, 1/\alpha)$ (linear decrease to 0), or (iii) $R^t \geq 1/\alpha$ (multiplicative decrease then case ii).
Tightness: $R_{\max}$ is asymptotically achieved under continuous maximum sharing, while 0 is reached after consecutive failures. Bounds $[0, R_{\max}]$ are strict for all DCs and epochs. □

In our context, *fairness* means that all operations and decisions depend solely on the DC's current reputation value. We prove that the expectation of reputational change from sharing is similar between different DCs.

Theorem 4. *Under DCTR assumption (Sec.3.1.3), our model satisfies δ-fairness (Definition 4) with $\delta = \alpha R_{\max}$ ($R_{\max} = \frac{DV_{\max}}{1-\beta}$).*

Proof. **Part 1. Expectation.** For DC$_i$ with S_i attempts:

$$\mathbb{E}[\Delta R_i] = \underbrace{p_{\text{succ}} \cdot S_i}_{\text{successes}} - \underbrace{(1-p_{\text{succ}})\mathbb{E}[P_i]}_{\text{failures}}$$

where penalty $P_i \leq \alpha R_{\max}$ by Equations (7) and Theorem 3.
Part 2. Normalized yield. The expected yield per sharing attempt:

$$\frac{\mathbb{E}[\Delta R_i]}{S_i} = p_{\text{succ}} - (1-p_{\text{succ}})\frac{\mathbb{E}[P_i]}{S_i}$$

Part 3. Bounded difference. For any two honest DCs i, j:

$$\left|\frac{\mathbb{E}[\Delta R_i]}{S_i} - \frac{\mathbb{E}[\Delta R_j]}{S_j}\right| \leq (1-p_{\text{succ}})\alpha R_{\max} \leq \alpha R_{\max}$$

since Theorem 3 stability bounds. □

6.3 Proactive and Honesty Incentives of the data sharing

We prove firstly that our model can incentivize the DCs of varying data volumes to proactively participate in sharing if the DCTR assumption holds (Section 8.1 for case proofs).

Theorem 5 (Reputation-Based Proactive Incentive)**.** *Under DCTR assumption (Sec.3.1.3), the model provides γ-proactive incentive (Definition 5) with $\gamma = \min(\gamma_1, \gamma_2, \gamma_3)$ for:*

$$\gamma_1 = \beta(1-\alpha) \qquad \textit{(Committee Member)}$$

$$\gamma_2 = \beta\left(1 - \frac{TH_{com}}{R_{\max}}\right) \qquad \textit{(Large Non-Committee)}$$

$$\gamma_3 = \beta \cdot \frac{\overline{DV}_{accessed}}{DV_i} \qquad \textit{(Small Non-Committee)}$$

Proof. **For Committee Member ($R_i \geq TH_{\text{com}}$):**

$\mathbb{E}[\Delta R|\text{Share}] = \beta(1)$ (successful sharing)
$\mathbb{E}[\Delta R|\text{NotShare}] = \beta(-\alpha R_i)$ (decay + penalty)
$\Delta_{\text{incentive}} = \beta(1-(-\alpha R_i)) \geq \beta(1-\alpha) \quad (\text{since } R_i \leq R_{\max})$

For Large Non-Committee ($R_i \approx TH_{\text{com}}$):

$$\mathbb{E}[\Delta R|\text{Share}] = \beta\left(1 + \underbrace{\frac{R_{\max} - R_i}{R_{\max}}}_{\text{committee access gain}}\right)$$

$$\mathbb{E}[\Delta R|\text{NotShare}] = \beta(-\alpha R_i)$$

$$\Delta_{\text{incentive}} \geq \beta\left(1 + 1 - \frac{TH_{\text{com}}}{R_{\max}}\right) > \beta\left(1 - \frac{TH_{\text{com}}}{R_{\max}}\right)$$

For Small Non-Committee ($R_i \ll TH_{\text{com}}$):

$$\mathbb{E}[\Delta R|\text{Share}] = \beta\left(1 + \underbrace{\frac{\overline{DV}_{\text{accessed}}}{DV_i}}_{\text{DCTR efficiency}}\right)$$

$$\mathbb{E}[\Delta R|\text{NotShare}] = 0 \quad (\text{negligible penalty})$$

$$\Delta_{\text{incentive}} = \beta\left(1 + \frac{\overline{DV}_{\text{accessed}}}{DV_i}\right) > \beta \cdot \frac{\overline{DV}_{\text{accessed}}}{DV_i}$$

Taking minimal γ across DC types ensures universal incentive. □

Then we prove that compared with honest DCs, the potential benefits for dishonest DCs are significantly diminished by the associated risks. Real case proofs will be given in Section 8.2.

Theorem 6. *Under the reputation update rules with decay β and penalty $\alpha > 0$, with detection probability $p_d > 0$, our model provides η-honesty advantage (Definition 6):*

$$\lim_{t\to\infty} \frac{\mathbb{E}[R_t^{honest}]}{\mathbb{E}[R_t^{dishonest}]} = \frac{1}{1-p_d}\left(1 + \frac{\beta p_d \alpha}{(1-\beta)(1-p_d)}\right) > 1$$

Proof. **For Honest DC,** reputation follows:

$$R_t^H = \beta(R_{t-1}^H + 1) \Rightarrow \lim_{t\to\infty} R_t^H = \frac{\beta}{1-\beta}$$

For Dishonest DC, reputation follows stochastic process:

$$\mathbb{E}[R_t^D] = \beta\left[(1-p_d)(\mathbb{E}[R_{t-1}^D] + 1) + p_d(1-\alpha)\mathbb{E}[R_{t-1}^D]\right]$$

At steady state ($\mathbb{E}[R_t^D] = \mathbb{E}[R_{t-1}^D] = R_\infty^D$):

$$R_\infty^D = \beta\left[(1-p_d)(R_\infty^D+1) + p_d(1-\alpha)R_\infty^D\right]$$

Solving:

$$R_\infty^D = \frac{\beta(1-p_d)}{1-\beta(1-p_d\alpha)}$$

So we can get the ratio:

$$\frac{R_\infty^H}{R_\infty^D} = \frac{\beta/(1-\beta)}{\beta(1-p_d)/(1-\beta(1-p_d\alpha))} = \frac{1}{1-p_d} + \frac{\beta p_d \alpha}{(1-\beta)(1-p_d)}$$

Both terms > 0 when $\alpha > 0, p_d > 0$, so ratio > 1. □

6.4 Defense Against Specific Attacks in the network

This section discusses our model's defenses against known attacks in the network.

DDoS attacks. In a P2P network, a DDoS attack can consume a large amount of bandwidth and impede the communication of normal DCs. Our model sets the DC to only invoke $Share$ and $Access$ requests to the PB. If access or sharing fails once, the DC will not be allowed to access or share again in this round. This means that the frequency of $Share$ and $Access$ between DC and PB is controllable. Thus it is difficult for a malicious DC to initiate a large number of malicious requests to consume bandwidth. So, our model can defend against DDoS attacks.

Sybil attack. In blockchain, a Sybil attack can spoof multiple virtual users to reach a consensus for malicious message writing. Our model makes it more difficult to fake a virtual DC by using multi-dimensional information at the time of initialization. When reaching consensus, our PoR consensus values its own RVs more than the number of DCs, thus reducing the role of virtual DCs in the consensus. So our model can defend against Sybil attacks.

Collusion Attack. In reputation-driven mechanisms, a collusion attack can achieve rapid RV growth of internal members by multiple DCs conspiring to form a malicious organization. For a malicious DC with a low RV, the only possible attack is to form a malicious group and increase the RV of one of them to achieve access to high threshold data. The increase of reputation value in our model is only realized by successful sharing, and the audit contract ensures the quality of shared data, the penalty and decay factors again limit the rapid growth of the reputation value of malicious nodes, which makes it robust against low RV DCs' attacks. We consider the worst case as all the committee members are malicious, the shared data cannot be used due to policy restrictions. In our model, our PoR ensures that only the DCs' number N^t over 375 can eliminate the fixing of the committee nm^t. Because Z_{R^t} approximates $(0-1)$ normal distribution and about 2.4% ($2\hat{\sigma}$) can be candidates by Equation (14). Therefore, just $2.4\% \cdot N^t \geq \lceil \log_2 N^t \rceil$ can replace committee members with $Votes$, this undermines the complicity of malicious groups.

Replay Attack. A Replay attack happens when a malicious DC can share similar data multiple times to increase its RV. Although our model allows sharing plaintext and the corresponding tags, it uses LSH that can map similar content to the same $index$ with high probability and eventually generate the same tags. Thus, the presence of repetitive auditing SC makes the sharing of similar data fail and leads to a decrease in the reputation value. Therefore, considering that the cost of counterfeiting is much higher than the increasing RV, which is successfully shared, our model can defend against Replay attacks.

7 Performance Analysis

Since our blockchain-based reputation-backed auditable model is dominated by SASP & RPSM, our model is most influenced by SASP in terms of performance evaluation. However, as analyzed in the previous section, the existing repetitive auditing lacks a mature scheme oriented to distributed architectures, so the auditing schemes we compare are still oriented to cloud storage using bilinear mapping for implementation. In this section, we conduct a theoretical comparison with existing auditing schemes [14], [18], [20] along with a comprehensive performance test of SASP, and verify the feasibility of the model driven by RPSM with a real-world case study in Section 8.

For the sake of comparison, we assume that the cyclic group $\mathbb{G}$ be a subgroup of $\mathbb{Z}_p^*$ with $|\mathbb{G}| = \lambda$[3]. We also ignore the multiple hash operation distinctions in existing schemes and unify all hash operations (including LSH and Ins in our model) as generic H computations. For clarity, we define some necessary operating descriptions in Table **??**, which E denotes exponential computation, H denotes hash computation, and P denotes bilinear mapping operation. Other computations such as the $\mod p$ operation are ignored due to their low overhead. Table 3 compares computational/storage/communication costs between existing schemes and our model, where our construction eliminates pairing operations (P) through careful algebraic design and storage overhead is reduced through tag aggregation.

7.1 Performance Evaluation

Via a peer node running Ubuntu 20.04 LTS on Intel Core i7-11700 @2.50GHz, 4GB RAM, we conducted microbenchmarks using Go's `crypto` and `math` libraries[4] with the following cryptographic primitives:

- Bilinear pairings: BN254 curve (196.41ns/op)
- Hash function: SHA-256 (13.92ns/4KB[5])
- Modular exponentiation: 11.73ns/op (λ=1024-bit)

Note that bilinear mapping in the test takes about 31.064% of the CPU, while all other operations have negligible CPU usage. The results of a more intuitive performance comparison based on the theoretical analysis in Table 3 are shown in Table 4. We categorize the operations into local-based and on-chain operations, and we can see that we achieves 75.44% lower on-chain overhead per block vs. [20], which considers duplicity and integrity auditing in Table. 4, where [14] and [18] only implement a portion of the data auditing. In summary, we can both achieve faster ciphertext integrity and repeatability auditing in a distributed architecture and provide restricted access control and reputation incentives.

[3] In practical pairing-based systems, $\mathbb{G}$ would *not* be implemented over $\mathbb{Z}_p^*$ due to subgroup confinement requirements for pairing compatibility.

[4] https://github.com/golang/go/tree/release-branch.go1.18

[5] Following Ng *et al.*'s research [60], a block size of 4KB was utilized in this study as it maximizes the space-saving efficiency of repetition.

TABLE 3
Comprehensive theoretical analysis in computing, storage and communication costs.

Computation Costs									
Schemes	**Initialization**		**Encryption**		**Sharing**	**Verify**	**Access**	**Auditing**	
	DC	**Data**	**Data**	**Tag**	**Sig**	**Ver**	**DC**	**Repetition**	**Integrity**
[14]	-	1H+1E	(2n+1)H	nH	2nE	3nE+1P	(n+1)H	1H	-
[18]	1H+2E	-	-	nH+2nE	-	2H+1E+1P	-	-	2nH+(n+1)E+1P
[20]	(n+2)H	2H+nE	(2n+1)H	nH+2nE	2nE	nE+1P	(2n+3)H+nE	9E+1P	4nH+4nE+1P
Ours	1H+1E	(2n+1)H+(n+1)E	2nE	(n+1)H	2H+1E	3E	(n+1)H+(n+3)E	nH	(n+2)H+1E
Storage and Communication Costs									
			Storage costs					**Communication costs**	
Schemes	**Initialization**		**Encryption**		**Sharing**	**Verify**	**Access**	**Auditing**	
	DC	**Data**	**Data**	**Tag**	**Sig**	**Ver**	**DC**	**Repetition**	**Integrity**
[14]	-	$2\lvert\mathbb{G}\rvert$	$1\lvert D\rvert$	(n+1)$*\lvert\mathbb{G}\rvert$	n$*\lvert\mathbb{G}\rvert$	-	$1\lvert D\rvert$ + n$*\lvert\mathbb{G}\rvert$	(2n+1)$*\lvert\mathbb{G}\rvert$	(2n+2)$*\lvert\mathbb{G}\rvert$
[18]	$2\lvert\mathbb{G}\rvert$	-	-	$2n*\lvert\mathbb{G}\rvert$	-	-	-	(3n+2)$*\lvert\mathbb{G}\rvert + \lvert D\rvert$	-
[20]	(n+2)$*\lvert\mathbb{G}\rvert$	(n+2)$*\lvert\mathbb{G}\rvert$	$1\lvert D\rvert$	(n+3)$*\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert$	-	$1\lvert D\rvert$+n$*\lvert\mathbb{G}\rvert$	(n+5)$*\lvert\mathbb{G}\rvert$	(n+3)$*\lvert\mathbb{G}\rvert$
Ours	$2\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert$	$2\lvert D\rvert$	(n+1)$*\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert$	-	$2\lvert D\rvert$+2$*\lvert\mathbb{G}\rvert$	(n+3)$*\lvert\mathbb{G}\rvert$	(n+3)$*\lvert\mathbb{G}\rvert$

Note : n is the number of blocks of D, and no consideration of access policy overhead. $|D|$ and $|\mathbb{G}|$ denote the length of shared data D and element in $\mathbb{G}$. H,E,P denote the evaluation of hash function, exponent operation, and bilinear mapping operation

TABLE 4
More intuitive performance comparison results based on the theoretical analysis in Table 3.

Computation Costs (ns).				
Schemes	**Time consumption on DCs**		**Time consumption on-chain**	
[14]	(4n+3)H+(2n+1)E	$\approx 79n + 53$	1H+3nE+1P	$\approx 35n + 210$
[18]	(n+1)H+(2n+2)E	$\approx 37n + 37$	(2n+2)H+(n+2)E+2P	$\approx 40n + 444$
[20]	(6n+8)H+6nE	$\approx 154n + 111$	4nH+(5n+9)E+3P	$\approx 114n + 695$
Ours	(4n+5)H+(4n+6)E	$\approx 103n + 140$	(2n+2)H+4E	≈ **28n+75**
Storage Costs.				
Schemes	**Storage overhead on DCs**		**Storage overhead on-chain**	
[14]	$2\lvert\mathbb{G}\rvert+1\lvert D\rvert$	≈ $n*$4KB+128B	(2n+1)$*\lvert\mathbb{G}\rvert$	≈ $n*$256B+128B
[18]	$2\lvert\mathbb{G}\rvert$	≈ 256B	(3n+2)$*\lvert\mathbb{G}\rvert + \lvert D\rvert$	≈ $n*$(4KB+384B)+256B
[20]	(2n+4)$\lvert\mathbb{G}\rvert+1\lvert D\rvert$	≈ $n*$(4KB+256B)+512B	(n+5)$*\lvert\mathbb{G}\rvert$	≈ $n*$128B+640B
Ours	$4\lvert\mathbb{G}\rvert+2\lvert D\rvert$	≈ $n*$8KB+512B	(n+3)$*\lvert\mathbb{G}\rvert$	≈ $n*$**128B+384B**
Communication Costs.				
Schemes	**Communication overhead on DCs**		**Communication overhead on-chain**	
[14]	n$\lvert\mathbb{G}\rvert+1\lvert D\rvert$	≈ $n*$ (4KB+128B)	(2n+2)$*\lvert\mathbb{G}\rvert$	≈ $n*$256B+256B
[18]	-	-	-	-
[20]	n$\lvert\mathbb{G}\rvert+1\lvert D\rvert$	≈ $n*$(4KB+128B)	(n+3)$*\lvert\mathbb{G}\rvert$	≈ $n*$128B+384B
Ours	$2\lvert\mathbb{G}\rvert+2\lvert D\rvert$	≈ $n*$8KB+256B	(n+3)$*\lvert\mathbb{G}\rvert$	≈ $n*$**128B+384B**

Note : the size of each Block=4KB, $|\mathbb{G}|$=1024bit, the time consumption of 1H≈23.93ns, 1E≈11.73ns, 1P≈194.41ns.
[14] only has duplicate audit for ciphertext, [18] only has plaintext integrity audit, [20] and Ours have integrity and duplicity audit with ciphertext.

7.1.1 Computational Complexity

For data D partitioned into $n\times$ 4KB blocks:

In **Sharing Phase**, the cost in our model is $\underbrace{n(3\mathsf{H}+3\mathsf{E})}_{\text{Block processing}} + \underbrace{4\mathsf{H}+3\mathsf{E}}_{\text{Data-level ops}}$;

In **Access Phase**, the cost in our model is $\underbrace{n(\mathsf{H}+\mathsf{E})}_{\text{Decryption}} + \underbrace{3\mathsf{E}+\mathsf{H}}_{\text{Key verification}}$;

In **On-chain Auditing**, the cost in our model is $\underbrace{n\mathsf{H}}_{\text{Repetition check}} + \underbrace{\mathsf{E}+(n+2)\mathsf{H}}_{\text{Integrity verification}}$.

7.1.2 Storage & Communication Costs

In **Storage Overhead**, the cost in our model is $\underbrace{n \times 8\text{KB} + 4|\mathbb{G}|}_{\text{DC}} + \underbrace{(n+3)|\mathbb{G}|}_{\text{PB}}$;

In **Communication Cost**, the cost in our model is $\underbrace{(n+3)|\mathbb{G}|}_{\text{Metadata}} + \underbrace{n \times 8\text{KB} + 2|\mathbb{G}|}_{\text{Ciphertexts}}$.

7.2 Experimental Results

On DCs, the data initialization part is divided into loading and chunking of data (Load), generation of corresponding keys (PKsk), encryption of data and corresponding decryption (EncDec), generation of tag set (GenTag) and finally signature and verification (SigVer).

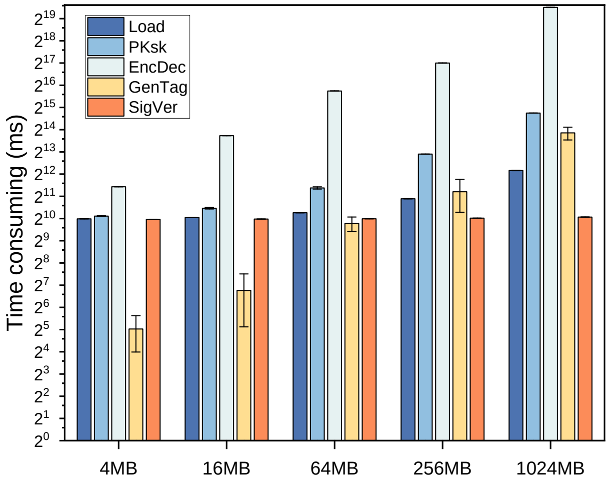

Fig. 3. Time consumption for operations with different data sizes.

Fig. 3 presents the time consumption for various operations with different data sizes. The time consumed for operations other than signature verification increases linearly with the number of blocks, indicating that the experimental results align with the theoretical analysis. On average, the most expensive encryption and decryption take about "2.84ms±0.174" per block with the Elgamal. Interestingly, in practice, due to data reads and repetitive auditing, tag generation using a hash function takes up to "0.997ms±0.136" per block. As a result, the elapsed time of the other operations remains largely stable across the different data sharing operations and is acceptable in terms of the overall elapsed time of the model itself. The experimental results indicate that operations completed within "1min" are acceptable when the data is within 64MB in terms of time consumption.

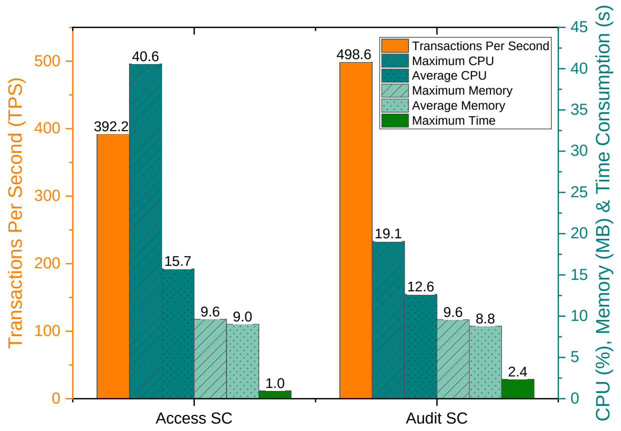

Fig. 4. Comprehensive performance testing of access and audit SC.

While involving the on-chain operations, in order to better evaluate the on-chain performance of our model, Hyperledger Fabric V2.2[6] is used to develop Access and Audit SC based on Go 1.18.8 and test based on Caliper[7]. We test 1000 times by caliper to call the Access SC and Audit SC within 60s, the results are shown in Fig. 4. The results show that the Access SC has higher resource consumption (CPU, Memory) than the Audit SC, while the Audit SC has higher time consumption due to the addition of repetitive auditing, but its throughput is higher than the Access SC due to the concurrency mechanism. We will further show in Section 8 that the experiment-based performance of our SASP is sufficient in the real world.

8 A Case Study

In this section, we demonstrate the feasibility of our model in real-world scenarios through a practical case study. We showcase the changes in each round using three metrics: DV, SV, and RV. DV represents the current cohort data owned by the DC; SV represents the cohort data shared by the DC in each round; and RV represents the initial reputation value of the DC in each round. Each cohort data size is 16KB, and the DV of each DC increases by 10% per round. To highlight the details, we abstract the data itself and represent changes using the data volume, which is a reasonable simplification.

In this case study, there are six hospitals[8] as DCs to validate that controlled flow of cohort data between hospitals can be achieved based on our model. These DCs are divided into BDC (initially owning 200 cohort data) and SDC (initially owning 2 cohort data) based on the actual number of cohorts they can provide. Each successful data sharing updates the DC's RV via Equation (6). We set the penalty factor $\alpha = 0.1$, decay factor $\beta = 0.9$ and update it for each round by Equation (12). The number of committee members is set to 3 according to $\lceil \log_2 N^t \rceil$. The block generator is determined by Equation (15).

8.1 First Phase

In the first phase, each cohort data provided is unique and complete. Given the scarcity of cohort data in medical scenarios, we implement restricted sharing, meaning that the shared data must meet certain access policies to be successfully accessed. This background leads to a question: What happens if a selfish DC sets an access policy that is never satisfied? We designate BDC3 as this selfish DC, while other DCs share data openly.

Under the existing schemes [21], [23], [35], other DCs fall into a passive state as they cannot access BDC3's shared data. Despite the presence of a decay coefficient, other DCs counteract the decay by sharing a fixed amount of data each round to protest the "unfairness". As shown in Fig. 5(a), there is a clear stagnation, where SDC only reaches 200 cohort data (BDC level) by the 33rd round.

To address this, we introduce the committee mechanism. Even with the selfish BDC3, committee members ignore access policies, forcing BDC3 to improve its RV to raise the access threshold for its shared data. The absence of committee members motivates other DCs to actively participate in sharing, ultimately benefiting all DCs. With BDC's SVs at 1% of their DV each round, the final DVs for SDCs reached "890", and BDC3's DVs reached "2,245" in Fig. 5(b).1, substantially surpassing the BDC's DVs achieved in normal scene that reached "862" in Fig. 5(a).1.

As analyzed in Section 6.2, the committee mechanism and access thresholds achieve proactive incentives. Therefore, we demonstrate in the first phase that our RPSM can break down the data silos and promote the DCs to participate in data sharing proactively.

8.2 Second Phase

In the second phase, we introduce scenarios with duplicate or incomplete cohort data. We conduct experiments in three scenarios (perfect, normal, and malicious) to evaluate the effectiveness and accuracy of the Audit SC. The first phase, with perfect data quality, is referred to as the perfect scenario. In the normal and malicious scenarios, the selfish BDC3 unintentionally or maliciously shares some duplicate or incorrect data (with a probability of 10%). The duplication rate is set to $\sigma = 20\%$.

In the normal scenario, BDC3 shares abnormal data with a 10% probability each round. The Audit SC reduces BDC3's

[6] https://github.com/hyperledger/fabric.git
[7] https://github.com/hyperledger/caliper.git

[8] There are 3 DCs in Beijing (with 200+, 1000+ and 2000+ beds), 1 in Tianjin (with 2000+ beds), 1 in Shenyang (with 300+ beds), 1 in Jinan (with 4000+ beds)

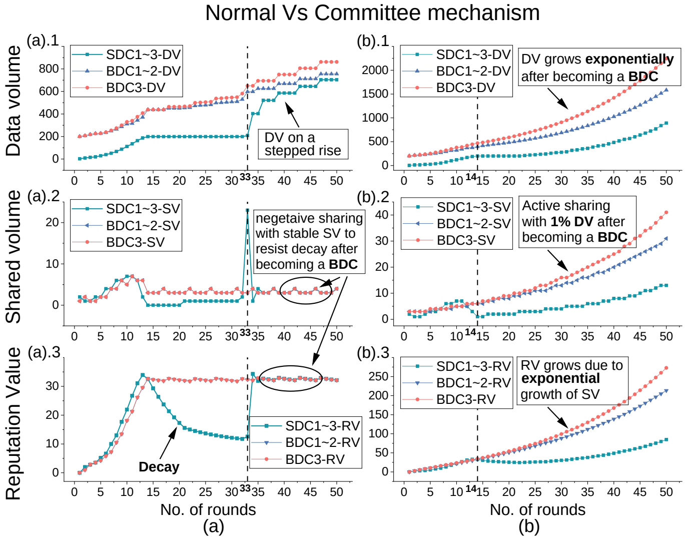

Fig. 5. Comparison of normal and our RPSM. (a) at left is the normal scene with negative sharing. (b) at right is the positive sharing with 1% DV scene with committee incentives.

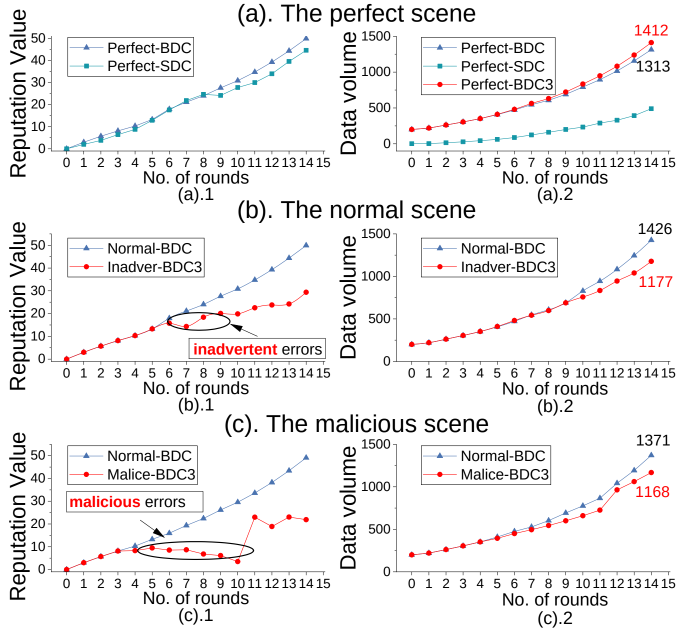

Fig. 6. Changes in reputation values and data volume in three scenes. (a) is the perfect scene. (b) is the normal scene with a 10% probability of inadvertent error. (c) is the malicious scene with 1 malicious behavior after 9 normal.

RV when duplicate or incomplete data is detected through Equation (7). The changes in DCs' RV and DV are illustrated in Fig. 6(b). Compared to the perfect scene in Fig. 6(a), BDC3's audit results do not affect other DCs.

In the malicious scene, BDC3 actively shares anomalous data after every 9 normal data shares (equal 10% probability). The audited changes in RV and DV are displayed in Fig. 6(c). Compared with Fig. 6(b).2, the DV of BDCs decreases because of the increasing proportion of useless data being shared. However, when compared with the perfect scene, the DV of BDC1-2 is higher. In the perfect scene, the selfish BDC3 is always the committee member, restricting BDC1-2 from accessing its shared data. However, in a malicious scene, each BDC3 shares malicious data, its RV decreases with the execution of Audit SC, and the data shared by BDC3 contributes to the increase of other BDCs' DVs.

Therefore, comparing the three scenes, it can be seen that in perfect and normal scenes, the sharing of each DC will not be affected. BDCs with malicious behavior will be maximally restrained by the Audit SC to reduce the impact on other DCs. For the malicious DC, the only way to increase its RV after being unable to bypass the Audit SC to achieve the malicious behavior is to create the perfect scene through increased SVs. In conclusion, our Audit SC guarantees the flow of data.

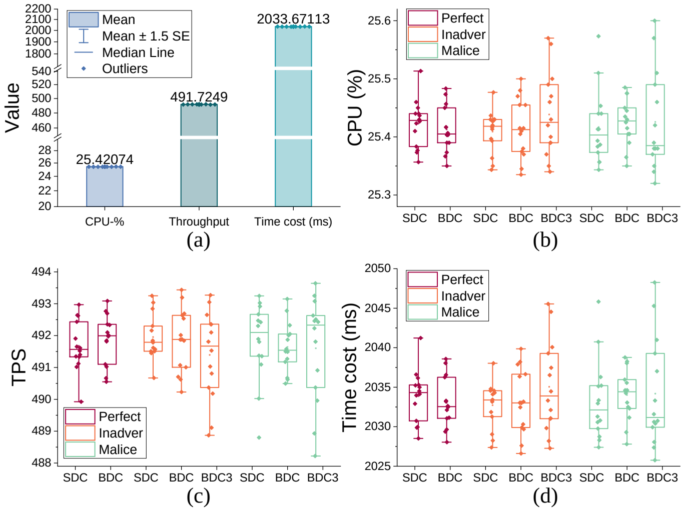

Fig. 7. Performance of Audit SC in three scenes. (a) is the total performance of Audit SC. (b) is the average CPU consumption. (c) is transaction throughput. (d) is the transaction latency of Audit SC.

In the second phase, due to the fact that our Audit SC is invoked in each round, independent of whether the scenario is normal or malicious, and maintains stability in terms of throughput and other aspects, it exhibits no significant fluctuations. As depicted in Fig. 7(a), our model can achieve approximately "1000" audits of data sharing in "2s" with a CPU usage of "25.42%". It demonstrates that our model is effective when auditing in terms of quality, stability, and efficiency in terms of computational performance. In specific scenes, it can be observed that the resource consumption of BDC3 fluctuates more than that of other DCs, as shown in Fig. 7(b, c, d). Meanwhile, unintentional errors in orange exhibit lower fluctuations than malicious operations in green. This is understandable, as the Audit SC needs to make more records on BDC3 in the malicious scenario. Interestingly, when comparing the perfect scenes, implementing repetition and integrity auditing based on the Audit SC does not significantly degrade performance, such as throughput, which actually increases due to BDC3 continuously sharing data and calling the contract to increase its chances of becoming a committee member. However, other factors increase accordingly. By comparing Fig. 7(b, d), we observe consistent characterization results, indicating a strong correlation between the SC's time consumption and CPU usage, which aligns with the algorithmic description.

9 Discussion

This section delves into a discussion of the proposed model. Initially, the nature of cohort data requires certain supervision in data flow. This limitation restricts its usability for large-scale use in public chain scenarios and, consequently, prevents an in-depth analysis of scalability and robustness involving multiple nodes.

Moreover, our approach involves multiple hashing operations to achieve efficient and secure data protection. This design choice avoids resource-intensive encryption and decryption operations, such as re-encryption. However, it poses challenges in finding suitable references for a comprehensive comparison, as existing mature algorithms are utilized without a thorough exploration of more advanced alternatives like bilinear maps.

In terms of the reputation mechanism, our case in Section 8 demonstrates that our RPSM is probabilistic and thus not entirely precise. However, the core focus of RPSM is incentivization rather than reputation evaluation. Therefore, our concern is the effectiveness of data-sharing incentives rather than the accuracy of reputation scoring. The case study also proves that our reputation-based incentive mechanism achieves the desired outcomes.

Finally, in data encryption, efficient symmetric encryption schemes (e.g., AES) hinder the direct mapping of ciphertext information to plaintext, making it difficult to distinguish the generated ciphertext from its repetitive counterpart, and existing research has addressed this problem through bilinear maps [14], [15], [19], [20] while similarly imposing a high performance overhead. Therefore, we adopt an asymmetric encryption scheme to construct the shared model.

While our work demonstrates notable advancements, certain shortcomings persist. Accurate auditing of the repetition of shared data remains challenging, and a more secure approach to authenticity verification of data private keys is yet to be discovered. While we have provided formal security proofs for all key properties (Theorems 1-6), the probabilistic nature of reputation updates (as shown in Fig. 6) introduces $\mathcal{O}(1/\sqrt{N})$ variance in fairness metrics. Future work will explore deterministic reputation mechanisms with similar incentive properties. Additionally, this study does not extensively evaluate network-layer robustness against attacks, as this focus lies outside the scope of the data-sharing layer. However, we recognize its importance and outline it as a potential direction for future work.

10 Future Work

In the forthcoming research, our primary objective is to identify auditing algorithms that efficiently reconcile repetition and integrity auditing while preserving the privacy of plaintext data. This entails seeking a delicate balance that ensures the robustness of our model without compromising the confidentiality of sensitive information. Another crucial aspect of our future investigations involves delving into the realm of secure proof mechanisms, particularly the application of advanced methods like the zero-knowledge proof. Our aim is to leverage these mechanisms to enhance the correctness verification process for transmitted file private keys. This avenue promises to provide a higher level of assurance and reliability in data security. Equally important is an in-depth study of game theory to model the behavior of DCs and derive the change in value gains among DCs under incentive mechanisms through Nash equilibrium or Markov decision processes.

11 Conclusion

In conclusion, our work presents a novel reputation-backed auditable cohort data-sharing model effectively addresses the challenges of data silos. In the integrated model, RPSM, incentivizes DCs to engage in active data sharing, and SASP ensures efficient and secure data flow with a focus on preserving data privacy. The integration is facilitated through smart contracts, and a comprehensive evaluation involving security and performance analysis substantiates the model's effectiveness. Case studies of real-world scenarios show that the model can be applied to a wide range of high-value, high-privacy data sharing scenarios and can provide novel solutions to the data quality and data quantity dilemmas in sharing.

References

[1] B. H. Thuesen, C. Cerqueira, M. Aadahl, J. F. Ebstrup, U. Toft, J. P. Thyssen, R. V. Fenger, L.-G. Hersoug, J. Elberling, O. Pedersen, T. Hansen, J. D. Johansen, T. Jørgensen, and A. Linneberg, "Cohort Profile: The Health2006 cohort, Research Centre for Prevention and Health," *International Journal of Epidemiology*, vol. 43, no. 2, pp. 568–575, Apr 2013.

[2] C.-N. Members and Partners, "Database Resources of the National Genomics Data Center, China National Center for Bioinformation in 2023," *Nucleic Acids Research*, vol. 51, no. D1, pp. D18–D28, Nov 2022.

[3] M. Kou, Y. Yang, and K. Chen, "Financial technology research: Past and future trajectories," *International Review of Economics & Finance*, vol. 93, pp. 162–181, 2024.

[4] J. Fang, L. Zhao, and S. Li, "Exploring open government data ecosystems across data, information, and business," *Government Information Quarterly*, vol. 41, no. 2, p. 101934, 2024.

[5] J. Kim, H. Ha, B.-G. Chun, S. Yoon, and S. K. Cha, "Collaborative analytics for data silos," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, 2016, pp. 743–754.

[6] L. Kapsner, J. Mang, S. Mate, S. Seuchter, A. Vengadeswaran, F. Bathelt, N. Deppenwiese, D. Kadioglu, D. Kraska, and H.-U. Prokosch, "Linking a consortium-wide data quality assessment tool with the miracum metadata repository," *Applied Clinical Informatics*, vol. 12, no. 04, pp. 826–835, Aug 2021.

[7] J. Jayabalan and N. Jeyanthi, "Scalable blockchain model using off-chain ipfs storage for healthcare data security and privacy," *Journal of Parallel and Distributed Computing*, vol. 164, pp. 152–167, June 2022.

[8] S. A. Kraft, M. K. Cho, K. Gillespie, M. Halley, N. Varsava, K. E. Ormond, H. S. Luft, B. S. Wilfond, and S. S.-J. Lee, "Beyond consent: building trusting relationships with diverse populations in precision medicine research," *The American Journal of Bioethics*, vol. 18, no. 4, pp. 3–20, Apr 2018.

[9] M. M. Lumpkin, M. A. Hamburg, W. B. Schultz, and J. M. Sharfstein, "Transparency practices at the fda: A barrier to global health," *Science*, vol. 377, no. 6606, pp. 572–574, Aug 2022.

[10] J. Chi, Y. Li, J. Huang, J. Liu, Y. Jin, C. Chen, and T. Qiu, "A secure and efficient data sharing scheme based on blockchain in industrial internet of things," *Journal of Network and Computer Applications*, vol. 167, p. 102710, Oct 2020.

[11] Z. Chen, W. Xu, B. Wang, and H. Yu, "A blockchain-based preserving and sharing system for medical data privacy," *Future Generation Computer Systems*, vol. 124, pp. 338–350, Nov 2021.

[12] A. Manzoor, A. Braeken, S. S. Kanhere, M. Ylianttila, and M. Liyanage, "Proxy re-encryption enabled secure and anonymous iot data sharing platform based on blockchain," *Journal of Network and Computer Applications*, vol. 176, p. 102917, Feb 2021.

[13] L. Xue, D. Liu, C. Huang, X. Shen, W. Zhuang, R. Sun, and B. Ying, "Blockchain-based data sharing with key update for future networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3437–3451, Dec 2022.

[14] G. Tian, Y. Hu, J. Wei, Z. Liu, X. Huang, X. Chen, and W. Susilo, "Blockchain-based secure deduplication and shared auditing in decentralized storage," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 6, pp. 3941–3954, Nov 2022.

[15] C. Li, M. Dong, X. Xin, J. Li, X.-B. Chen, and K. Ota, "Efficient privacy preserving in iomt with blockchain and lightweight secret sharing," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 22 051–22 064, Dec 2023.

[16] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Advances in Cryptology – EUROCRYPT 2013*, T. Johansson and P. Q. Nguyen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 296–312.

[17] H. Yuan, X. Chen, J. Wang, J. Yuan, H. Yan, and W. Susilo, "Blockchain-based public auditing and secure deduplication with fair arbitration," *Information Sciences*, vol. 541, pp. 409–425, 2020.

[18] X. Yang, X. Pei, M. Wang, T. Li, and C. Wang, "Multi-replica and multi-cloud data public audit scheme based on blockchain," *IEEE Access*, vol. 8, pp. 144 809–144 822, 2020.

[19] S. Li, C. Xu, Y. Zhang, Y. Du, and K. Chen, "Blockchain-based transparent integrity auditing and encrypted deduplication for cloud storage," *IEEE Transactions on Services Computing*, vol. 16, no. 1, pp. 134–146, Jan 2023.

[20] M. Song, Z. Hua, Y. Zheng, H. Huang, and X. Jia, "Blockchain-based deduplication and integrity auditing over encrypted cloud storage," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 6, pp. 4928–4945, Nov 2023.

[21] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, June 2019.

[22] R. Zou, X. Lv, and J. Zhao, "Spchain: Blockchain-based medical data sharing and privacy-preserving ehealth system," *Information Processing & Management*, vol. 58, no. 4, p. 102604, July 2021.

[23] J. Cui, F. Ouyang, Z. Ying, L. Wei, and H. Zhong, "Secure and efficient data sharing among vehicles based on consortium blockchain," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8857–8867, July 2022.

[24] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, Mar 2019.

[25] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Satoshi Nakamoto*, 2008. [Online]. Available: https://static.upbitcare.com/931b8bfc-f0e0-4588-be6e-b98a27991df1.pdf

[26] D. Deuber, B. Magri, and S. A. K. Thyagarajan, "Redactable blockchain in the permissionless setting," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 124–138.

[27] G. Ramseyer, A. Goel, and D. Mazières, "SPEEDEX: A scalable, parallelizable, and economically efficient decentralized EXchange," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. Boston, MA: USENIX Association, Apr 2023, pp. 849–875.

[28] Z. Liu, Y. Xiang, J. Shi, P. Gao, H. Wang, X. Xiao, B. Wen, and Y.-C. Hu, "Hyperservice: Interoperability and programmability across heterogeneous blockchains," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 549–566.

[29] G. Panwar, R. Vishwanathan, S. Misra, and A. Bos, "Sampl: Scalable auditability of monitoring processes using public ledgers," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2249–2266.

[30] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18. New York, NY, USA: Association for Computing Machinery, Apr 2018.

[31] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, July 1985.

[32] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, ser. SCG '04. New York, NY, USA: Association for Computing Machinery, June 2004, p. 253–262.

[33] H. Jin and J. Xiao, "Towards trustworthy blockchain systems in the era of "internet of value": development, challenges, and future trends," *Science China Information Sciences*, vol. 65, pp. 1–11, 2022.

[34] J. Yu, D. Kozhaya, J. Decouchant, and P. Esteves-Verissimo, "Repucoin: Your reputation is your power," *IEEE Transactions on Computers*, vol. 68, no. 8, pp. 1225–1237, Aug 2019.

[35] J. Zhang, Y. Cheng, X. Deng, B. Wang, J. Xie, Y. Yang, and M. Zhang, "A reputation-based mechanism for transaction processing in blockchain systems," *IEEE Transactions on Computers*, vol. 71, no. 10, pp. 2423–2434, Oct 2022.

[36] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, J. Garcia-Alfaro, G. Navarro-Arribas, H. Hartenstein, and J. Herrera-Joancomartí, Eds. Cham: Springer International Publishing, Sep 2017, pp. 297–315.

[37] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, Jan. 2019.

[38] G. S. Ramachandran, R. Radhakrishnan, and B. Krishnamachari, "Towards a decentralized data marketplace for smart cities," in *2018 IEEE International Smart Cities Conference (ISC2)*, 2018, pp. 1–8.

[39] Z. Guo, G. Wang, Y. Li, J. Ni, and G. Zhang, "Attribute-based data sharing scheme using blockchain for 6g-enabled vanets," *IEEE Transactions on Mobile Computing*, vol. 23, no. 4, pp. 3343–3360, 2024.

[40] K. O.-B. O. Agyekum, Q. Xia, E. B. Sifah, C. N. A. Cobblah, H. Xia, and J. Gao, "A proxy re-encryption approach to secure data sharing in the internet of things based on blockchain," *IEEE Systems Journal*, vol. 16, no. 1, pp. 1685–1696, 2022.

[41] X. Zhang, W. Xia, Q. Cui, X. Tao, and R. P. Liu, "Efficient and trusted data sharing in a sharding-enabled vehicular blockchain," *IEEE Network*, vol. 37, no. 2, pp. 230–237, 2023.

[42] X. Xu, K. Meng, H. Xiang, G. Cui, X. Xia, and W. Dou, "Blockchain-enabled secure, fair and scalable data sharing in zero-trust edge-end environment," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2025.

[43] W. Tong, X. Dong, Y. Shen, Y. Zhang, X. Jiang, and W. Tian, "Chchain: Secure and parallel crowdsourcing driven by hybrid blockchain," *Future Generation Computer Systems*, vol. 131, pp. 279–291, June 2022.

[44] C. Zhang, C. Xu, H. Wang, J. Xu, and B. Choi, "Authenticated keyword search in scalable hybrid-storage blockchains," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, April 2021, pp. 996–1007.

[45] L. Han, G. Hu, X. Li, F. Xia, S. Wang, and L. You, "A novel lattice-based blockchain infrastructure and its application on trusted data management," *IEEE Transactions on Network Science and Engineering*, pp. 1–13, 2025.

[46] I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, "Privysharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities," *Computers & Security*, vol. 88, p. 101653, Jan 2020.

[47] D. Reijsbergen, A. Maw, T. T. A. Dinh, W.-T. Li, and C. Yuen, "Securing smart grids through an incentive mechanism for blockchain-based data sharing," in *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '22. New York, NY, USA: Association for Computing Machinery, Apr 2022, p. 191–202.

[48] N. Truong, G. M. Lee, K. Sun, F. Guitton, and Y. Guo, "A blockchain-based trust system for decentralised applications: When trustless needs trust," *Future Generation Computer Systems*, vol. 124, pp. 68–79, Nov 2021.

[49] S. Purohit, R. Neupane, N. R. Bhamidipati, V. Vakkavanthula, S. Wang, M. Rockey, and P. Calyam, "Cyber threat intelligence sharing for co-operative defense in multi-domain entities," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 5, pp. 4273–4290, Sep. 2023.

[50] M. Rezvani, A. Ignjatovic, E. Bertino, and S. Jha, "Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 98–110, Jan 2015.

[51] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2009, pp. 911–918.

[52] J. R. Douceur, "The sybil attack," in *Peer-to-Peer Systems*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, Oct 2002, pp. 251–260.

[53] D. Boneh, "The decision diffie-hellman problem," in *Algorithmic Number Theory*, J. P. Buhler, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, May 1998, pp. 48–63.

[54] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, p. 422–426, July 1970.

[55] M. Zhang, J. Li, Z. Chen, H. Chen, and X. Deng, "Cycledger: A scalable and secure parallel protocol for distributed ledger via sharding," in *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2020, pp. 358–367.

[56] X. Hao, W. Ren, Y. Fei, T. Zhu, and K.-K. R. Choo, "A blockchain-based cross-domain and autonomous access control scheme for internet of things," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 773–786, Mar 2023.

[57] Z. Guan, X. Lu, W. Yang, L. Wu, N. Wang, and Z. Zhang, "Achieving efficient and privacy-preserving energy trading based on blockchain and abe in smart grid," *Journal of Parallel and Distributed Computing*, vol. 147, pp. 34–45, Jan 2021.

[58] M. Zhang, J. Li, Z. Chen, H. Chen, and X. Deng, "An efficient and robust committee structure for sharding blockchain," *IEEE Transactions on Cloud Computing*, vol. 11, no. 3, pp. 2562–2574, July 2023.

[59] F. Gai, B. Wang, W. Deng, and W. Peng, "Proof of reputation: A reputation-based consensus protocol for peer-to-peer network," in *Database Systems for Advanced Applications*, J. Pei, Y. Manolopoulos, S. Sadiq, and J. Li, Eds. Cham: Springer International Publishing, May 2018, pp. 666–681.

[60] C.-H. Ng and P. P. C. Lee, "Revdedup: a reverse deduplication storage system optimized for reads to latest backups," in *Proceedings of the 4th Asia-Pacific Workshop on Systems*, ser. APSys '13. New York, NY, USA: Association for Computing Machinery, July 2013.

Biography Section

Jie Zhang (Student Member, IEEE) received the M.S. degree in computer application technology from the Yunnan Normal University in 2022. He is currently working toward the Ph.D. degree in computer science and technology with the Tianjin University. His research interests include efficient and secure data sharing within blockchain systems.

Xiaohong Li (Member, IEEE) received the Ph.D. degree in computer application technology from Tianjin University in 2005. She is currently a Full Tenured Professor with the Department of Cyber Security, College of Intelligence and Computing, Tianjin University. Her research interests include knowledge engineering, trusted computing, and security software engineering.

Ruitao Feng is a Lecturer at Southern Cross University, Australia. He received the Ph.D. degree from the Nanyang Technological University. His research centers on security and quality assurance in software-enabled systems, particularly AI4Sec & SE. This encompasses learning-based intrusion/anomaly detection, malicious behavior recognition for malware, and code vulnerability detection.

Shanshan Xu received the M.S. degree in computer application technology from the Yunnan Normal University in 2022. She is currently working toward the Ph.D. degree in physical geography with the School of Geographic Sciences, East China Normal University. Her research interests include Atmospheric vapor, machine learning and model building.

Zhé Hóu is a Senior Lecturer at Griffith University, Australia. He obtained his Ph.D. degree from the Australian National University in 2015. His research mainly focuses on automated reasoning, formal methods, AI, quantum computing and blockchain.

Hanwei Wu is a position of Associate Professor of Information Security at Hainan University. He received the B.S. from Peking University and he is a Ph.D. candidate in the Computer Science at Tianjin University. His research interests include applied cryptography, security protocol analysis, security penetration and defense.

Guangdong Bai (Member, IEEE) received the B.S. and M.S. degrees in computing science from Peking University in 2008 and 2011, respectively, and the Ph.D. degree in computing science from the National University of Singapore in 2015. He is currently a Senior Lecturer with The University of Queensland. His research interests include cyber security, software engineering, and machine learning.