electronics

MDPI

Article

Data Secure Storage Mechanism for Trustworthy Data Space

Xinyi Yang [1], Qicheng Luo [1], Jiang Xu [1,2,*] and Qinghong Cao [3]

[1] School of Computer Science, School of Cyber Science and Engineering, Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing 210044, China; 202312490814@nuist.edu.cn (X.Y.)
[2] Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin 541004, China
[3] System Operation Headquarters, Shanghai Stock Exchange Technology Co., Ltd., Shanghai 200120, China
* Correspondence: 001136@nuist.edu.cn

Abstract

In today's rapidly evolving data environment, secure and efficient storage solutions are fundamental to supporting the robust development of the data economy. Trustworthy data space serves as an innovative technological framework for addressing critical challenges in data circulation. It is specifically designed to facilitate the secure exchange in data elements and overcome trust barriers in cross-organizational data sharing. However, current decentralized storage architectures still have significant implementation gaps. Practical deployment and system integration remain substantial challenges for existing technological solutions. To address these issues, this paper first conducts a systematic analysis of existing trusted data storage methods. On this basis, it proposes a data-secure storage mechanism based on polynomial commitment. This mechanism uses polynomial commitment to implement data storage and verification, thereby ensuring data integrity and consistency. Meanwhile, it integrates homomorphic signature technology to guarantee the authenticity of data sources without disclosing original data. Additionally, a data modification recording function is introduced to ensure the traceability of all operations. Experimental results show that the proposed scheme achieves superior performance in three key aspects: communication overhead, storage efficiency, and data update costs.

Keywords: homomorphic signature; on-chain and off-chain hybrid storage; polynomial commitment; trustworthy data space

Academic Editor: Hung-Yu Chien

Received: 13 September 2025
Revised: 2 November 2025
Accepted: 4 November 2025
Published: 6 November 2025

Citation: Yang, X.; Luo, Q.; Xu, J.; Cao, Q. Data Secure Storage Mechanism for Trustworthy Data Space. *Electronics* **2025**, *14*, 4348. https://doi.org/10.3390/electronics14214348

1. Introduction

The accelerating digital transformation has elevated secure, efficient, and sovereign-compliant data sharing to a critical global priority. European initiatives are pioneering innovative approaches to open data implementation through customized technical components and adaptive standards frameworks. The European Open Science Cloud (EOSC) [1] establishes a federated digital research infrastructure. It combines distributed computing resources with identity federation protocols and FAIR principles to create a unified scientific data commons. Complementing this ecosystem, OpenAIRE [2] operates as a knowledge integration hub. It employs semantic technologies to enhance the discoverability and contextual value of multidisciplinary research outputs through systematic aggregation and enrichment processes.

Data spaces embody a distributed framework for data exchange. They transcend traditional centralized or federated models by implementing uniform protocols. These

protocols safeguard information sharing among diverse entities, systems, and end-users. The International Data Space (IDS) reference architecture [3] serves as the core technical standard for this ecosystem. It provides fundamental support for decentralized data environments. In 2021, this architecture was systematically integrated through the Data Space Business Alliance (DSBA). DSBA was established by IDS Alliance, FIWARE, and other organizations. This integration enabled DSBA to become the leading framework for building and connecting data space. DSBA uses a decentralized structure and standardized interfaces. It aims to create a secure and trustworthy data-sharing environment. It facilitates data exchange across different organizations, industries, and countries through these unified standards. This ensures both security and transparency in data transactions [4].

The trustworthy data space [5] was proposed by the National Data Standardization Technical Committee (SAC/TC609) of China. It is a data circulation infrastructure based on consensus rules. It enables secure data sharing and value co-creation through digital contracts and usage control technologies. Trustworthy data space ensures the safe and reliable storage, exchange, and sharing of data among multiple parties. It achieves this through technical means and institutional design. The framework has three core objectives: ensuring data security, preserving privacy, maintaining integrity, and facilitating optimal data value utilization [6]. As an infrastructure based on consensus rules, a trustworthy data space connects multiple entities. It realizes the sharing and use of data resources and promotes the circulation and use of data [7]. Nevertheless, secure data storage—its foundational capability—still faces core challenges.

Traditional cloud storage has become one of the mainstream choices for current data storage, thanks to its flexible storage space allocation and convenient remote access capabilities [8]. However, its centralized architecture inherently has an insurmountable trust bottleneck at the technical level: the security protection, privacy preservation, and integrity verification of data ultimately depend entirely on the credibility endorsement of cloud service providers and the stability of centralized systems [9]. This single-point trust model not only exposes data to risks such as malicious abuse by internal personnel of the service provider and large-scale leakage caused by system vulnerabilities but also easily becomes a target for concentrated attacks by external attackers. Even if optimized through technical means such as encryption algorithms and access control, these measures are merely "patchwork" fixes and cannot fundamentally resolve the contradiction between the centralized architecture and the demand for "multi-entity mutual trust". Especially in cross-organizational data sharing scenarios, it is difficult for different institutions to establish trust solely based on the commitments of cloud service providers, and the compliance and traceability of data circulation are even less guaranteed.

To address this dilemma, blockchain has emerged for its distributed ledger and tamper-proof features. It enables joint data record maintenance via a multi-node consensus mechanism (without centralized authorities), achieving "data immutability" and "full traceability of operations" [10]. Its hash-chain structure requires reconstructing subsequent block hashes for any data modification; with multi-node backups, attack costs outweigh benefits, natively protecting data integrity. Additionally, blockchain's decentralization resolves the "lack of trust intermediaries" in cross-organizational scenarios—instutions can verify data directly via consensus rules, no third-party endorsement needed [11].

It should be noted, however, that blockchain itself is not a perfect solution. In terms of data protection and privacy, the transparency of blockchain directly conflicts with the demand for privacy preservation. If sensitive data is directly stored on the chain, even if user addresses use anonymous identifiers, attackers can still associate user identities, trace behavioral patterns, and even obtain sensitive information in batches through public metadata such as transaction flows and timestamps. On the other hand, the immutability

of blockchain means that once sensitive data is mistakenly stored on the chain, it cannot be deleted. Even if subsequent withdrawal is required to comply with regulatory requirements, the full-node backup feature will cause the data to remain permanently, creating long-term privacy risks. Furthermore, in terms of storage and performance, the "full-node backup" mechanism of blockchain causes data storage costs to rise sharply as the volume of on-chain data increases. At the same time, data processing speed is limited by the consensus mechanism and block capacity; in scenarios involving large-scale data writing or high-frequency interactions, problems such as increased latency and insufficient throughput are likely to occur [12].

To address the core contradiction between blockchain's inherent storage and privacy performance, the "on-chain and off-chain hybrid storage" model has emerged. It divides responsibilities through "functional layering and data separation": the on-chain layer stores trust anchors such as data hashes and verification credentials to ensure credibility, while the off-chain layer hosts massive volumes of original and sensitive data to resolve privacy risks and performance bottlenecks. This model also optimizes the storage efficiency of trusted data spaces from the dimensions of "trust-efficiency-privacy", yet it faces challenges including the credibility of off-chain data, real-time synchronization between on-chain and off-chain layers, and the scalability of verification mechanisms for large-scale operations [13].

To tackle this, this paper proposes a novel secure storage mechanism specifically designed for trusted data environments, addressing the storage challenges in trusted data spaces caused by the increasing number of blockchain nodes and transactions, while ensuring storage efficiency and security compliance. By combining polynomial commitment and homomorphic signature, this mechanism guarantees data integrity, authenticity, and consistency and optimizes communication and verification overhead. Compared with traditional solutions, it not only improves the storage and verification efficiency in trusted data environments but also enhances the security of data storage components.

The contributions of this paper are as follows:

- This paper proposes a blockchain-based data security storage framework for trustworthy data space. In this framework, metadata is stored on-chain and business data is stored off-chain. Furthermore, data integrity and consistency are verified by querying one or more positional indexes stored in polynomial commitment on-chain.
- A data modification function is designed to record the update status of data, which achieves traceability in the data storage mechanism.
- To ensure data authenticity in the trustworthy data space, the paper verifies the identity of data senders. Combined with homomorphic signature technology, the authenticity of data sources can be verified without exposing original data.

The paper is organized as follows: Section 2 introduces data storage technologies and reviews previous research on homomorphic signatures. Section 3 discusses the security challenges in trustworthy data space storage systems and presents the specific solutions proposed in this work. Section 4 details the system model, formal definitions, and the construction of the polynomial commitment-based storage security guarantee mechanism for trustworthy data space. It also includes correctness proofs and security analysis. Section 5 presents a performance evaluation of the proposed scheme. Section 6 concludes the paper.

2. Related Work

2.1. Secure Data Storage for Trustworthy Data Space

In a trustworthy data space, data needs to be stored efficiently and securely, and trustworthy data properties (which include data integrity, authenticity, and consistency between on-chain and off-chain) must be ensured. Therefore, a reliable, trustworthy data storage

security assurance mechanism needs to be designed. Napione et al. [14] investigated the use of privacy-preserving technologies in data storage systems, introducing cryptographic and decentralized storage methods to mitigate risks of data breaches. However, storage performance may be affected by encryption and distributed storage, especially in the case of large amounts of data, which may lead to slower access speeds and consistency issues. Bonura et al. [15] explored the privacy protection of federated learning in data storage and reduced the risk of leakage by distributing data storage. Jadhav et al. [16] proposed a security-enhanced IPFS framework integrated with hybrid encryption and constructed a platform for secure file sharing. Nevertheless, federated learning requires high bandwidth for model updates. Moreover, data consistency and trust issues in distributed environments have not been fully addressed. Although these two solutions can effectively protect data privacy, they face efficiency and consistency issues when processing large-scale data.

To address storage scalability for trustworthy data space, Esposito et al. [17] studied healthcare cloud storage. They found a key issue: even when conventional encryption technologies are used, cloud service providers' "super administrator privileges" can still be a vulnerability. This vulnerability may lead to privacy leakage. Specifically, service providers can access unauthorized healthcare data. They do this through technical backdoors. What is more, such operations are hard to supervise from the outside. Even if service providers promise to comply with rules, there is no third-party verifiable credibility. Wang et al. [18] looked at access control and data traceability. They further pointed out a problem with cloud storage: access permission management depends completely on centralized nodes. Therefore, cloud providers themselves may not be fully trustworthy, potentially leading to data leaks, unauthorized modifications, or abuse.

Given these persistent challenges, blockchain technology is particularly suitable for trustworthy data storage due to its inherent properties: decentralization, immutability, and auditability. Yang et al. [19] developed an access control framework utilizing blockchain technology for cloud storage environments, implementing both authorization and deauthorization operations. Nevertheless, this methodology continues to impose substantial storage demands on the blockchain infrastructure, rendering it impractical for large-scale data applications. Kuznetsov et al. [20] proposed a scalable zero-knowledge proof scheme based on the Plonky2 framework, which addresses the issues of high computational overhead and poor scalability in large-scale data encryption hash verification within blockchain systems. However, this scheme only focuses on single-point technical optimization for general hash verification scenarios. It lacks the full-process and multi-dimensional guarantee capabilities required by a secure storage system and fails to establish a complete secure storage architecture. Liang et al. [21] proposed an industrial blockchain data storage and recovery scheme based on local regenerative codes and blockchain consensus mechanisms. This scheme addresses the issues of low efficiency and high storage overhead in multi-node fault repair. However, it lacks a blockchain-native identity authentication mechanism, making it unable to perform identity confirmation and legitimacy verification for nodes or entities participating in operations. Meanwhile, it fails to record data interaction behaviors and operation trajectories among multiple nodes on the blockchain, resulting in difficulties in achieving behavior traceability and trusted endorsement for multi-entity collaboration. In a separate approach, Sun et al. [22] employed Hyperledger Fabric to construct a medical blockchain-enabled system for record storage and sharing, wherein the blockchain provides tamper-evident record preservation, smart contracts regulate access permissions, and cryptographic methods ensure data and signature security. Zhang et al. [23] introduced a blockchain-based mobile edge computing framework, which provides low-latency response, anonymous authentication, and data integrity protection.

2.2. Data Storage Based on Blockchain

In the data storage of trustworthy data space, blockchain has emerged as a fundamental technical strategy to ensure data reliability. Scholars have conducted extensive research on secure data storage in trustworthy data space, gradually expanding from early data security verification to performance optimization and privacy protection. Liu et al. [24] introduced a solution that leverages a trustworthy execution environment (TEE). By expanding the trust scope of the blockchain, it solves the risk of off-chain data tampering during the collection and storage process and enhances the credibility of on-chain and off-chain collaboration, but it relies on hardware security. Liu et al. [25] proposed FACOS, a system enhancing off-chain storage privacy via fine-grained access control integrated with on-chain permissions, though frequent dynamic updates reveal significant performance limitations. In the file sharing scenario, Ding et al. [26] introduced a hybrid storage architecture that integrates on-chain file metadata with off-chain file content to enable secure file sharing and non-repudiation, although further optimization is still required for the efficiency of off-chain storage.

In response to the vulnerability detection and scalability issues in on-chain and off-chain hybrid storage, Zhang et al. [27] developed the ĐArcher framework, which detects on-chain and off-chain synchronization vulnerabilities through automated testing, effectively reducing the risk of data inconsistency, but its detection relies on predefined patterns and is difficult to fully cover abnormal situations. In response to the interoperability issues in cross-chain and multi-chain environments, Cao et al. [28] proposed a decentralized cross-chain solution, MAP, which optimizes the security and scalability of cross-chain operation, but its compatibility in complex cross-chain scenarios still needs further research.

In the field of privacy protection, the research results combining blockchain and attribute encryption (ABE) have made significant progress. For example, Pandey et al. [29] proposed a hybrid storage solution based on document classification, which combines on-chain storage metadata with off-chain storage of actual documents, thereby enhancing the document privacy protection capability, but still faces the challenge of high computational overhead in high-frequency access scenarios. The blockchain-based distributed storage solution [30] improves overall storage efficiency and guarantees data integrity by managing large-scale data off-chain while utilizing on-chain storage for metadata. Miyachi et al. [31] proposed a modular privacy-preserving framework based on an on-chain and off-chain hybrid architecture. While ensuring data auditability, this framework fundamentally prevents the leakage of raw data containing sensitive information. However, its data integrity verification relies on third parties, lacking on-chain native verification capabilities. Additionally, there is no corresponding on-chain synchronization and consistency guarantee strategy for data update operations.

Eberhardt and Tai [32] proposed a strategy to alleviate on-chain load through selective off-chain computation and storage in specific use cases. However, this approach demonstrates significant limitations in real-time data validation efficiency. Ahmadjee and Bahsoon [33] proposed a classification system for blockchain security technical debt. Their work assesses structural vulnerabilities inherent in hybrid on/off-chain storage architectures and highlights how formally measuring such technical debt influences the sustainable security of blockchain infrastructures. To enhance consensus mechanisms in blockchain, Kiayias et al. [34] developed the Ouroboros protocol—a prominent Proof-of-Stake framework that employs state channels and off-chain processing to lessen on-chain storage requirements while maintaining transaction consistency and security. Nonetheless, synchronizing state channels continues to present challenges that warrant further optimization. Meanwhile, Guo et al. [35] proposed a cloud data storage mechanism leveraging blockchain technology, achieving security by on-chain recording of transaction information

and off-chain retention of data content. It is suitable for data integrity verification and identity authentication in cloud environments, but there is still room for improvement in the off-chain data synchronization efficiency. Goint et al. [36] introduced a consent management mechanism utilizing data encryption to tackle the issue of off-chain data access control, which effectively improves the privacy of off-chain storage. However, there is still room for improvement in permission synchronization and verification efficiency. In response to the storage needs of large-scale spatiotemporal data, Ren et al. [37] proposed an on-chain and off-chain hybrid storage mechanism called BSMD, which uses an updateable subvector commitment technology to build a secure authentication protocol, thereby alleviating the problem of insufficient blockchain storage capacity and ensuring data consistency. However, within large-scale data storage environments, the associated proof size grows substantially as data volumes expand, thereby impairing the efficiency of both on-chain storage and verification processes. Additionally, there remains a need to enhance the computational handling of intricate off-chain data relationships.

Although these studies have progressively addressed the issues of data security, consistency verification, and privacy safeguarding in hybrid on-chain and off-chain storage and promoted the application of blockchain technology in cloud storage, distributed storage and multi-chain environments, the processing efficiency, dynamic update performance and multi-chain compatibility of large-scale data still need further in-depth research. To address these issues, this paper introduces the polynomial commitment technique: the committer commits to a polynomial using a short string, and the verifier uses this short string to verify the evaluation result of the committed polynomial. Both its size and initialization overhead are fixed, and the communication overhead remains stable even when a large number of verifications are required. Shen et al. [38] introduced a publicly verifiable and secure computational scheme for cloud environments utilizing polynomial commitment, demonstrating the efficiency of this approach in reducing computational overhead.

Existing approaches fail to simultaneously ensure the trustworthy data properties. Therefore, we use the homomorphic properties of polynomials to introduce homomorphic signature technology [39] into the proposed secure storage mechanism. Dan Boneh et al. [40] demonstrated that for constant-degree polynomials, the size of the resulting signature grows merely logarithmically with the dataset's dimensions. Therefore, adding a polynomial homomorphic signature to the polynomial commitment scheme of constant size significantly guarantees the security of the blockchain data storage system.

3. Secure Requirement of Data Storage for Trustworthy Data Space

As blockchain technology rapidly evolves, the on-chain and off-chain hybrid storage architecture has steadily emerged as an efficient solution to overcome the storage limitations of blockchain. However, as the demand for trustworthy data environments grows, securing decentralized data storage has become a crucial issue. Such storage must guarantee three essential attributes: tamper resistance, privacy preservation, and managed access control—each introducing considerable implementation difficulties. Although the hybrid storage framework mitigates on-chain load by shifting data storage off-chain and using blockchain for verification, it also introduces a number of security concerns.

Confidentiality. Since data is stored on off-chain servers, data owners cannot fully control the access rights to their data. This creates a risk that unauthorized third parties may access and steal the data.

Integrity. Off-chain stored data is vulnerable to tampering by attackers during transmission and storage. It is thus difficult to ensure that the data remains free from malicious modifications during use.

Authenticity. This refers to ensuring the credibility of the source of data, messages, or communication content, as well as preventing tampering or forgery during transmission. Its core purpose is to verify the identity of the message sender and ensure the message remains intact during transmission, without malicious modification by any third party.

Consistency. The data proof recorded on-chain may not reflect the latest status of off-chain data in a timely manner, leading to data inconsistency.

To tackle the issues mentioned above, this work proposes a hybrid storage architecture mechanism that utilizes blockchain and cryptographic techniques. Multiple technical strategies are adopted to ensure trustworthy data properties. The corresponding implementation methods for each security property are illustrated in Figure 1.

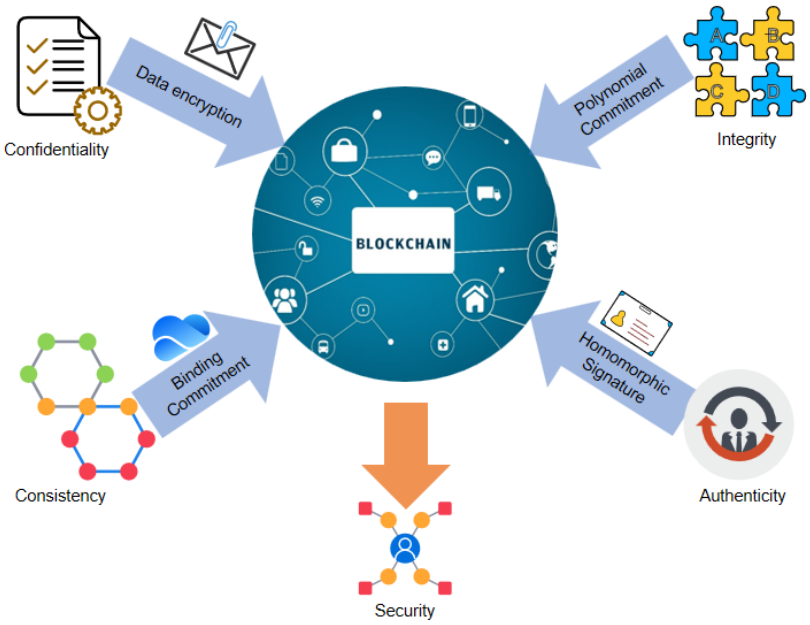

Figure 1. Security property for trustworthy data space.

Confidentiality. The encryption method adopted in this paper works as follows: the smart contract randomly selects a polynomial to encrypt the plaintext message uploaded by the data owner. This ensures that unauthorized users cannot access the data during transmission, storage, and processing. It also guarantees that only authorized users can decrypt and access the off-chain stored data.

Integrity. Data verifiers can verify the integrity of off-chain stored messages by combining smart contracts with the mathematical properties of polynomials. Specifically, data verifiers provide the database record index i and execute the Verify algorithm to check whether the message is complete. As a result, the on-chain commitment value can record modifications to off-chain messages.

Authenticity. When sending messages, the data owner uses a polynomial function to generate a homomorphic signature. This allows the data verifier to confirm the signer's identity while keeping the message confidential. Additionally, the homomorphic signature can aggregate polynomial signatures of multiple messages.

Consistency. Any on-chain stored data can be verified off-chain. Whenever off-chain data is updated, the corresponding on-chain commitment will be updated, and a record will be added under the relevant index. Moreover, only verified data can affect on-chain data, thus ensuring consistency.

To enhance the security of the storage architecture, this study leverages polynomial functions and their homomorphic characteristics to design a hybrid storage model integrated with smart contracts. By incorporating encryption and verification techniques, the

model preserves the confidentiality and integrity of off-chain data. Meanwhile, homomorphic signatures are used to authenticate the data source. Furthermore, updates to off-chain data are linked to the corresponding on-chain commitment values, and each modification triggers a synchronization process with on-chain records. This approach effectively addresses consistency challenges, thereby enhancing the overall security of the system.

4. Storage Security Guarantee Mechanism Based on Polynomial Commitment for Trustworthy Data Space

4.1. Framework of Data Security Storage for Trustworthy Data Space

The core goal of the solution proposed in this paper is to build a distributed data storage and verification infrastructure that aligns with the concept of trustworthy data space. This system architecture is specifically designed to meet the core requirements of data sovereignty, trustworthy circulation, and cross-organizational collaboration in data space. As shown in Figure 2, this blockchain system includes five key entities, which together form the technical foundation for supporting the operation of trustworthy data space.

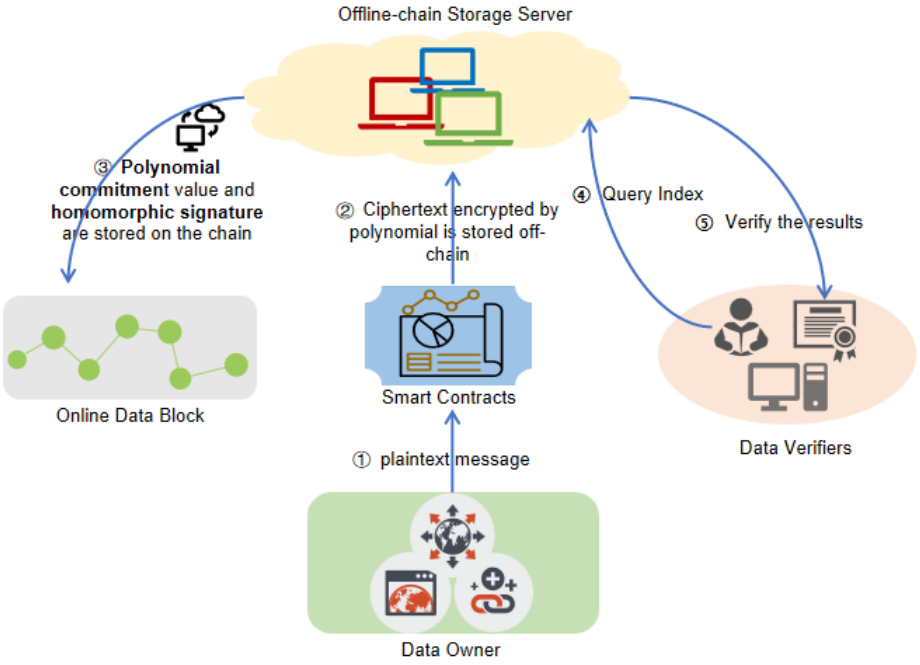

Figure 2. Data security storage mechanism for trustworthy data space.

Data owner: The data owner serves as the entity responsible for transmitting data to the blockchain infrastructure. During system initialization, this party generates the necessary cryptographic key pair (PK, SK) and subsequently creates homomorphic signatures through polynomial operations specified by the smart contract in the setup phase. The data owner is responsible for maintaining data sovereignty and in charge of preparing and uploading original records to trigger the storage process. This authorized entity retains full operational capabilities—including data persistence and modification operations—while generating corresponding cryptographic signatures. Following verification by blockchain nodes, the system establishes an irrevocable binding between commitment values and their associated signatures, then uploads the indexed records to the distributed ledger. Furthermore, the data owner retains continuous access privileges to off-chain repositories, enabling regular authentication of both data integrity preservation and confidentiality safeguards throughout the storage lifecycle.

Data verifier: The data verifier plays a significant role in maintaining blockchain integrity by engaging in both the consensus mechanism and data validation procedures.

Their primary responsibilities include (1) retrieving data from off-chain storage servers and corresponding blockchain records; (2) executing predefined verification algorithms to authenticate data integrity; and (3) processing multiple concurrent query requests (ranging from 1 to n) during off-chain data validation. The storage server responds to these queries by returning the requested information along with cryptographically generated proofs at designated locations. Upon receipt, verifiers analyze this evidentiary material to complete the authentication process, thereby ensuring secure and valid data transactions throughout the system. If the verification is successful, a confirmation message is returned; otherwise, an error is reported if verification fails.

Online data block: The process begins with the data owner storing all generated metadata in specialized on-chain data blocks. This operational model guarantees that any changes in the off-chain database environment trigger the associated server to automatically record cryptographically authenticated signatures onto the blockchain ledger, in strict compliance with established blockchain storage standards. By binding and storing polynomial commitments on the blockchain, the system ensures both security and traceability.

Off-chain data storage server: The off-chain storage server refers to an entity or service provider responsible for managing off-chain data—including encrypted raw data and polynomial coefficients. It provides efficient storage and retrieval capabilities and is responsible for processing users' access requests.

Smart contract executor: It executes smart contracts on the blockchain, randomly selects polynomial functions, generates polynomial commitment values for corresponding messages, and stores these values in the on-chain server. Additionally, it coordinates data interaction and verification between data owners and off-chain storage providers. It implements functions such as data upload, polynomial commitment generation, and data verification to ensure trustworthy data properties.

4.2. Secure Storage Solution Based on Polynomial Commitment for Trustworthy Data Space

Formal Definition. The secure storage model based on polynomial commitment consists of four algorithms (*Setup*, *Commit*, *Query*, *Verify*), as follows:

- Setup $(1^a, t)$: The data owner needs to store a string of messages, and the smart contract executor runs the algorithm. After the execution is completed, the smart contract outputs PK, which is used to generate commitment and query messages on the blockchain.
- Commit (PK, DB): The data owner uploads the message, the smart contract executor encrypts the message according to the selected polynomial and stores the encrypted value off-chain according to the index, and the generated commitment value is stored on the chain.
- Query (PK, k_i): The data verifier provides the index i to be queried, and the smart contract generates auxiliary parameters and auxiliary polynomials based on the previously selected polynomial and off-chain data.
- Verify (C, PK): Substitute the generated parameters to determine whether the commitment value has been tampered with.

Specific Solution Construction: We specifically describe the secure storage scheme based on polynomial commitment below. A concise table of the input and output data for the store-verify algorithms is shown in Table 1.

Setup $(1^a, t)$: In the initial setup phase, the smart contract selects security parameters μ, polynomial order t, key s selected from the number group and generator g, p, creates the system public key PK, and saves it to the blockchain. The smart contract outputs $PK = \{G,\ G_T,\ g,\ g^s,\ p,\ p^s\}$ at this stage.

Commit (PK, DB): In the storage commitment phase, the smart contract generates a commitment C according to the polynomial formula. $DB = \{I, K, V\}$, I is the stored index value, the encrypted value k_i is stored off-chain, the commitment value v_i is stored on-chain, the polynomial $\varphi_1(x)$ and polynomial $\varphi_2(x)$ are selected, the commitment value $C = g^{\varphi_1(s)} \cdot p^{\varphi_2(s)}$, and the commitment value C are stored on-chain. For $\varphi_1(x) = \sum_{j=0}^{deg(\varphi_1)} \varphi_{1j} x^j$ and $\varphi_2(x) = \sum_{j=0}^{deg(\varphi_2)} \varphi_{2j} x^j$, it outputs $C = \prod_{j=0}^{\deg(\varphi_1)} \left(g^{s^j}\right)^{\varphi_{1j}} \cdot \prod_{j=0}^{\deg(\varphi_2)} \left(p^{s^j}\right)^{\varphi_{2j}}$ as a commitment to the polynomial.

Query (PK, k_i): In the query phase, the smart contract generates auxiliary polynomials ϕ and auxiliary parameters θ according to the established formula. The detailed steps are outlined below: provide the index i to be queried to the smart contract, generate auxiliary polynomials φ_1 and auxiliary polynomials φ_2, auxiliary polynomials $\varphi_{1i}(x) = (\varphi_1(x) - \varphi_1(i))/(x - i)$, auxiliary polynomials $\varphi_{2i}(x) = (\varphi_2(x) - \varphi_2(i))/(x - i)$ and auxiliary parameters $\theta_i = g^{\varphi_{1i}(s)} \cdot p^{\varphi_{2i}(s)}$.

Verify (C, PK): In the verification phase, determine whether the equation $e(C, g) = e\left(\theta_i, g^s / g^i\right) e\left(g^{\varphi_1(i)} p^{\varphi_2(i)}, g\right)$ is established. If it holds, return 1 to the data verifier; otherwise, return 0.

Table 1. Concise table of store–verify algorithms.

Phase	Input Parameters	Output Parameters
Setup	μ, t, s, g, p	$PK = \{G, G_T, g, g^s, p, p^s\}$
Commit	$DB = \{I, K, V\}, \varphi_1(x), \varphi_2(x)$	$C = g^{\varphi_1(s)} \cdot p^{\varphi_2(s)}$
Query	i	$\varphi_1, \varphi_2, \theta_i$
Verify	$e(C, g) \stackrel{?}{=} e(\theta_i, g^s / g^i) e(g^{\varphi_1(i)} p^{\varphi_2(i)}, g)$	1/0

Database Dynamic Operations: Our scheme enables dynamic operations by storing data, along with its associated index values and keys, off-chain. If the data owner modifies the off-chain database, the corresponding index values and keys must be updated accordingly. To support these operations, index hash tables are used in conjunction with digital signature technology. A detailed explanation of dynamic operations is provided below:

Update: To update the i-th off-chain data block, the data owner first encrypts the new block m_i' locally into k_i', then transmits k_i' to the off-chain database. The off-chain database subsequently replaces the specified i-th block with this encrypted value. After that, the authority computes updated verification metadata and stores it in the on-chain index table (without overwriting existing entries). To verify the update status of the i-th off-chain data block, the data owner queries the smart contract with index i and checks the timestamp T in the on-chain record.

Delete: When the data owner needs to remove the i-th data block, the data is not directly erased but marked as deleted. Specifically, set the data record value v_i, corresponding to k_i, and counter T on the blockchain to 0, and add a deletion mark (deleted = 1). To confirm deletion, the data owner provides index i to the smart contract and checks the record T_i to verify the mark.

Insert: When removing the i-th data block from the blockchain, the data owner does not erase it physically; instead, the block is logically marked as deleted. Specifically, the values of the data record v_i, its corresponding k_i, and the counter T are reset to 0 in the blockchain, while a deletion flag (deleted = 1) is set to reflect the invalidation of the data block. To verify the deletion status, the data owner submits the index i to the smart contract, which then checks the corresponding entry T_i to confirm whether the deletion marker has been set.

4.3. Homomorphic Signature Based on Polynomial Function

Formal Definition of Homomorphic Signature of Polynomial Function. A polynomial homomorphic signature scheme consists of the following four functions:

- Setup $(1^{\mu}, k)$: The data owner generates pk and sk. The public key defines the message space M, the signature space Σ and the valid functions set $F : M^k \to M$.
- Sign (sk, m, τ, i): The data owner utilizes the key sk, the label $\tau \in \{0,1\}$, n, the message $m \in M$ and the index $i \in \{1, \ldots, k\}$, and generates the signature $\chi \in \Sigma$.
- Verify (pk, m, τ, χ, f): The data verifier employs the public key pk, the label $\tau \in \{0,1\}$, n, the message $m \in M$, the signature $\chi \in \Sigma$ and the function $f \in F$ to authenticate the message sender's identity, then outputs 0 (reject) or 1 (accept).
- Evaluate $(pk, \tau, f, \vec{\chi})$: The data owner can obtain the public key pk, the label $\tau \in \{0,1\}$, n, the function $f \in F$ and the signature tuple $\vec{\chi} \in \Sigma^k$, aggregate the signatures of different messages, and output the signature $\chi' \in \Sigma$.

Specific Description of Polynomial Function Homomorphic Signature. The following content will elaborate on the homomorphic signature based on polynomial function, and a concise table of the input and output for this process is shown in Table 2.

Table 2. Concise table of authentication.

Phase	Input Parameters	Output Parameters
Setup	μ, k	$pk = (F, j, l, u, v, d, s, H, y), sk = T$
Sign	$sk, \tau \in \{0,1\}^n, m, i$	signature: χ
Verify	$pk, \tau \in \{0,1\}^n, m, \chi, f$	1/0
Evaluate	$pk, \tau, f, \vec{\chi}$	evaluation: $\hat{f}(\chi_1, \ldots, \chi_k)$

1. Setup $(1^{\mu}, k)$. Provide the μ (security parameter) and the k (maximum dataset size) as inputs, and then execute the subsequent actions:
   i. Select n as an irreducible polynomial of degree, $F(x) \in \mathbb{Z}[x]$, $\gamma_p = p(n)$. Let $K := \mathbb{Q}[x]/(F(x))$ be embedded into Q^n by coefficient embedding. Let $R = Z_n$ be the lattice associated with $\mathbb{Z}[x]/(F(x)) \subseteq \mathcal{O}_K$.
   ii. Run the PrincGen algorithm twice with inputs F and n to generate two distinct principal degree prime ideals $u = (u, x - j)$ and $v = (v, x - l)$ of R, along with their respective generators g_u and g_v.
   iii. Let T be the basis of u, v, $\{g_u g_v, g_u g_v x \ldots, g_u g_v x^{n-1}\}$.
   iv. The parameter $s := {Y_P}^2 \cdot \mu^3 \cdot \log \mu$ is defined, with integers $y = poly(n)$ and $d = O(1))$ chosen.
   v. A hash function $H : \{0,1\}* \to F_q$ is employed, treated as a random oracle.
   vi. The keys are generated as $pk = (F, j, l, u, v, d, s, H, y)$ for the public key and $sk = T$ for the secret key.

   The public key pk specifies the following system parameters:

   - Messages are elements of F_p, while signatures take the form of compact vectors from R.
   - The permitted function class F includes all polynomials from $F_p[x_1, \ldots, x_k]$ satisfying coefficients ranging from $-y$ to y, a maximum degree of d, and no constant term (Note: The parameter y is employed solely within the verification process).
   - Consider the combinatorial parameter $l = \binom{k+d}{d} - 1$. Let $\{Y_j\}_{j=1}^{l}$ enumerate all non-constant monomials ${x_1}^{e_1} \cdots {x_k}^{e_k}$ with total degree $\sum e_i \leq d$, ordered lexicographically. Each polynomial function $f \in F$ admits representation:

$f(\overrightarrow{m}) = \sum_{j=1}^{l} t_j Y_j(\overrightarrow{m})$, where coefficients $t_j \in F_p$ are interpreted as integers in $[-y, y]$. The canonical encoding is $\langle f \rangle = (t_1, \ldots, t_l) \in \mathbb{Z}^l$.

- For hash function ω_τ operating on encoded polynomial $\langle f \rangle = (t_1, \ldots, t_l) \in \mathbb{Z}^l$:
  (a) Generate k field elements $\forall i \in \{1, \ldots, k\}$, compute $\alpha_i \leftarrow H(\tau||i)$.
  (b) Evaluate $\omega_\tau(\langle f \rangle) := \sum_{j=1}^{l} t_j Y_j(\alpha_1, \ldots, \alpha_k) \in F_q$.

2. Sign (sk, m, τ, i). Input the secret key sk, n-bit tag $\tau \in \{0,1\}^n$, a message in prime field $m \in F_p$ and an index i, and perform the following operations:
   i. Derive $\alpha_i := H(\tau||i) \in F_q$ through hash computation.
   ii. Construct polynomial $h = h(x) \in R$ satisfying $h(j) \bmod u = m, h(l) \bmod v = \alpha_i$.
   iii. Generate signature $\chi \leftarrow SamplePre\ (u \cdot v,\ T, h, s) \in (u \cdot v) + h$.
3. Verify (pk, m, τ, χ, f). For inputs pk, $\tau \in \{0,1\}^n$, $m \in F_p$, signature $\chi = \chi(x) \in R$, and function $f \in F$, accept (output 1) if all conditions hold, otherwise reject (output 0):
   (a) Norm constraint $\|\chi\| \leq \ell \cdot y \cdot \Upsilon_p^{d-1} \cdot (s\sqrt{n})^d$.
   (b) Message check $\chi(j) \bmod u = m$.
   (c) Authentication check $\chi(l) mod v = \omega_\tau(\langle f \rangle)$.
4. Evaluate $(pk, \tau, f, \vec{\chi})$. Processing inputs pk, $\tau \in \{0,1\}^n$, $f \in F$ encoded as $\langle f \rangle = (t_1, \ldots t_l) \in Z^l$ and a signature $\chi_1, \ldots, \chi_k \in Z^n$, and perform the following:
   i. Elevate $\hat{f} \in F_p(x_1, \ldots, x_k)$ to $Z(x_1, \ldots, x_k)$ via $f := \sum_{j=1}^{l} t_j Y_j(x_1, \ldots, x_k)$.
   ii. Return the evaluation $\hat{f}(\chi_1, \ldots, \chi_k)$.

4.4. Security Analysis

The correctness and security of the scheme are analyzed under three key conditions: maintained process consistency, verified sender authenticity, and network operation at peak performance levels. Special attention is paid to security evaluation in scenarios where no failure conditions are present.

Theorem 1 (correctness)**.** *If the data owner and the off-chain storage server faithfully carry out the algorithm and aggregate proof protocol of our proposed scheme, the data verifier is assured to receive a valid verification outcome.*

Proof. In the proposed storage scheme, the commitment C is generated by a function Commit (PK, DB), where pk represents the public key generated by Setup (a, b). For a given query Query(PK, k_i), its output is verified by the verification function Verify (C, PK). If the verification equation $e\left(\theta_i, g^s / g^i\right) e\left(g^{\varphi_1(i)} p^{\varphi_2(i)}, g\right) = e(C, g)$ is satisfied, it ensures the integrity and verifiability of the database record, thereby confirming both the immutability of the data and the correctness of the verification process.

Let $PK \leftarrow \text{Setup}(a, b)$ and $C \leftarrow \text{Commit}(PK, DB)$. For the commitment C output by Commit(PK, DB) and all $\varphi(x) \in \mathbb{Z}_p[x]$, the output of Query$(PK, k_i)$ has been successfully verified by Verify(C, PK).

As outlined in Section 4.2, a valid verification equation enables the data verifier to confirm the correctness of the scheme's verification procedure. The left-hand component of this equation is computed as follows:

$$e(\theta_i, g^s / g^i)\ e(g^{\varphi_1(i)} p^{\varphi_2(i)}, g) = e(g^{\varphi_{1i}(s)} g^{\lambda \cdot \varphi_{2i}(s)}, g^{s-i}) e(g^{\varphi_1(i)} g^{\lambda \cdot \varphi_2(i)}, g)$$

Note that it has been defined $\varphi_i(x) = (\varphi(x) - \varphi(i))/(x - i)$ in the scheme Query(PK, k_i), and substituting it into the formula, the above formula

$$
\begin{aligned}
&= e\left(g^{[(\varphi_1(s)-\varphi_1(i))/(s-i)]}g^{\lambda\cdot[(\varphi_2(s)-\varphi_2(i))/(s-i)]},\ g^{s-i}\right)e\left(g^{\varphi_1(i)}g^{\lambda\cdot\varphi_2(i)},\ g\right)\\
&= e\left(g^{[\varphi_1(s)-\varphi_1(i)]}g^{\lambda\cdot[\varphi_2(s)-\varphi_2(i)]},\ g\right)e\left(g^{\varphi_1(i)+\lambda\cdot\varphi_2(i)},\ g\right)\\
&= e\left(g^{\varphi_1(s)+\lambda\cdot\varphi_2(s)},\ g\right)\\
&= e\left(g^{\varphi_1(s)}p^{\varphi_2(s)},\ g\right)\\
&= e(C,\ g).
\end{aligned}
$$

The right-hand side of the verification equation is given by $e(\theta_i, g^s/g^i)e(g^{\varphi_1(i)}p^{\varphi_2(i)}, g) = e(C, g)$. When this equation is satisfied, the validity of the verification process is confirmed. Therefore, the reliability of the verification procedure proposed in this work is formally established. □

Theorem 2 (Security)**.** *If the scheme meets both the binding and hiding properties of the commitment scheme, as well as the zero-knowledge property of the zero-knowledge proof, then the commitment value and associated evidence are unforgeable. This ensures that the proposed scheme is resistant to forgery attacks, making the proposed polynomial commitment scheme secure.*

Proof. The scheme can resist forgery attacks. Specifically, the probability that forged data received by the verifier passes the verification algorithm is negligible. Forged data can appear in two situations: the prover forges the data, or the verifier forges the data.

If a dishonest prover forges data to deceive the verifier, the polynomial $\varphi(x)$ as well as generators g and p used by the prover will be untrue and may be replaced by $\varphi'(x)$. According to the hiding property of commitments, no party can forge $\varphi(x) \neq \varphi'(x)$ to satisfy commit$[\varphi(x)]$ = commit$[\varphi'(x)]$. Thus, the probability that a dishonest prover computes a commitment value commit$[\varphi'(x)]$ equal to commit$[\varphi(x)]$ is negligible. In this case, the hiding property of polynomial commitment is satisfied. The security of data storage in the commitment scheme is guaranteed on the condition that for every non-uniform polynomial-time interactive adversary $\mathcal{A}$, there exists a negligible function $\varepsilon(k)$ such that the advantage of $\mathcal{A}$ is bounded by $\varepsilon(k)$.

$$
\Pr\left(\begin{array}{c}\text{VerifyPloy(Params, } C, f(x)) = 1\\ \text{VerifyPloy(Params, } C, f'(x)) = 1\\ f(x) \neq f'(x)\end{array}\ \middle|\ \begin{array}{c}\text{Params} \leftarrow \text{Setup}\left(1^k, d\right)\\ (C, f(x), f'(x)) \leftarrow \mathcal{A}(\text{Params})\end{array}\ \middle|\ \leq \varepsilon(k)\right) \quad (1)
$$

If there is a dishonest verifier who deceives the prover, after Alice calculates the response algorithm, Bob needs to forge a random number z'' so that the verification evidence $w' = w$ at z''. Due to the binding property of the commitment scheme, if the function is considered a negligible function $\varepsilon(k)$, then Bob's probability of successfully completing the verification is also negligible.

$$
\Pr\left(\begin{array}{c}\text{Params} \leftarrow \text{Setup}(k),\ \langle C, z', f(z'), w\rangle,\\ \langle z'', f'(z''), w'\rangle :\\ \text{verify(Params, } C, z'', f'(z''), w') = 1 \wedge\\ \text{verify(Params, } C, z'', f'(z''), w') = 1\ \wedge\ f(z'') = f'(z'')\end{array}\right) = \varepsilon(k) \quad (2)
$$

At this stage, the scheme fulfills the binding characteristic of polynomial commitment. If there exists an insignificant function $\varepsilon(k)$ for all interactive adversaries $\mathcal{A}$ with non-uniform polynomial time, the commitment scheme satisfies the binding characteristic and can guarantee the security of the transaction value.

$$\Pr\left(\begin{array}{c}\text{Verify}(\text{Params}, C, i, z_1^i, w_1^i) = 1 \\ \text{Verify}(\text{Params}, C, i, z_2^i, w_2^i) = 1 \\ z_1^i \neq z_2^i\end{array}\right) \quad \begin{array}{l}\text{Params} \leftarrow \text{Setup}(1^k, d) \\ (C, \langle i, z_1^i, w_1^i\rangle, \langle i, z_2^i, w_2^i\rangle) \leftarrow \mathcal{A}(\text{Params})\end{array} \tag{3}$$

The zero-knowledge property ensures that no information about stored data is revealed to verifiers during proof verification. If Bob obtains any knowledge of the data, this will violate the hiding property of polynomial commitment and contradict Equation (1). This proves that the scheme maintains complete data confidentiality.

In conclusion, the proposed approach effectively meets the binding and concealment requirements of the commitment mechanism. It also complies with the zero-knowledge property of the cryptographic proof system. As a result, the probability of an adversary falsifying transaction information remains computationally negligible. This demonstrates the overall reliability of the proposed solution. □

Theorem 3. *The proposed scheme supports public verifiability of calculation results. All data verifiers are able to use the public key PK and commitment C to publicly verify the correctness of the query data through the verification function* $\text{Verify}(C, PK)$ *to ensure the validity of the calculation results.*

Proof. In the calculation result verification phase, the data verifier uses the *PK*, index i, and commitment C to confirm the correctness of the calculation result C_i. This indicates that any entity can use the *PK* and rely on the *TA* to validate the accuracy of the calculation result. Thus, the proposed scheme supports public verification.

The verification process requires three inputs, namely the *PK*, a database record proof t, and a commitment value C. Among these inputs, t acts as the query response. It can be accessed by any party holding the corresponding database record key. The commitment C is publicly distributed to all participants in the system. This design allows any client to validate database contents using the *PK*, which ensures the VDB scheme achieves public verifiability. □

The following lemma are already in the article [41] and are used here without further proof.

Lemma 1. *There is an algorithm PrincGen that takes as input a n-dimensional irreducible polynomial* $f(x) \in \mathbb{Z}[x]$ *and a parameter* δ*, and outputs a principal first-degree prime ideal* $\mathfrak{u} = (u; x - j)$ *in* $K := \mathbb{Q}[x]/(f(x))$ *and a generator* g *of* $\mathfrak{p}$ *that satisfies* $\|g\| \leq \delta\sqrt{n}$.

Lemma 2. *Assume* Λ *is an m-dimensional lattice. There exists a deterministic polynomial-time algorithm that, given an arbitrary basis of* Λ *and a full-rank set* $S = \{s_1; \ldots; s_m\}$ *in* Λ*, returns a basis* T *of* Λ *that satisfies* $\|T\| \leq \|S\|$ *and* $\|T\| \leq \|S\|\sqrt{m}/2$.

Lemma 3. *Let* Λ *be an n-dimensional lattice, let* T *be a basis of* Λ*, and suppose* $\chi \geq \|\widetilde{T}\| \cdot \omega(\sqrt{\log n})$*. Then for any* $\boldsymbol{c} \in \mathbb{R}^n$*, we have* $\Pr\left[\|\boldsymbol{x} - \boldsymbol{t}\| > \chi\sqrt{n} : \boldsymbol{x} \xleftarrow{R} \mathcal{D}_{\Lambda,\sigma,\boldsymbol{c}}\right] \leq negl(n)$.

Theorem 4. *A valid signer can produce a verification outcome that aligns with the message m and the signature* χ*, ensuring that the signature remains valid once it passes the verification process. Furthermore, no adversary can counterfeit a legitimate signature for this case.*

Proof. Let $\tau \in \{0,1\}^n$, $m \in \mathbb{F}_p$ and $i \in \{1, \ldots, k\}$, and define $\chi = Sign(sk, \tau, m, i)$. We verify the three conditions related to the projection function π_i, where π_i is a monomial x_i.

Lemma 1 establishes that the generators g_u, g_v satisfy at most $n^{1.5}$. Consequently, for all indices $i = 0, \ldots, n-1$, the polynomial products obey the bound $\|g_u g_v x^i\| \leq Y_P^2 \cdot n^3$. Applying Lemma 2 to basis T yields $\|T\| \leq Y_P^2 n^3$, which implies the lower bound $s \geq \|\widetilde{T}\| \cdot \omega(\sqrt{\log n})$. Furthermore, Lemma 3 guarantees that the inequality $\|x\| \leq s\sqrt{n}$ holds with overwhelming probability.

The SamplePre algorithm guarantees $\chi \in u \cdot v + h(x)$, which implies $\chi(j) \bmod u = h(j) \bmod u$. From the construction of $h(x)$, it follows that $h(j) \equiv m \bmod u$. Similarly, since χ maintains the same algebraic structure, we derive $\chi(l) \bmod v = h(l) \bmod v = \chi_i$.

Consider $\tau \in \{0,1\}^n$, message vector $\vec{m} = (m_1, \ldots, m_k) \in \mathbb{F}_p^k$ and $f \in \mathcal{F}$ be encoded as $\langle f \rangle = (t_1, \ldots, t_\ell) \in \mathbb{Z}^\ell$. The signature vector $\vec{\chi} = (\chi_1, \ldots, \chi_k)$ is generated through $\chi_i \leftarrow \mathrm{Sign}(\mathrm{sk}, \tau, m_i, i)$. With each coefficient t_j interpreted as an integer in $[-y, y]$. The evaluation operation yields: $\chi' := \mathrm{Evaluate}(pk, \tau, f, \vec{\chi}) = \sum_j t_j Y_j(\vec{\chi})$.

Each monomial evaluation satisfies $\|Y_j(\vec{\chi})\| \leq Y_P^{d-1} \cdot (\max\{\|\chi_i\| : \chi_i \in \vec{\chi}\})^d$ due to at most d multiplications. Validity of f ensures for all j, we have $|t_j| \leq y$. Combining these gives: $\|\chi'\| \leq \ell \cdot y \cdot Y_P^{d-1} \cdot (\max\{\|\chi_i\| : \chi_i \in \vec{\chi}\})^d \leq \ell \cdot y \cdot Y_P^{d-1} \cdot (s\sqrt{n})^d$ where the final inequality follows from Lemma 3.

The mapping $\widehat{h(x)} \mapsto h(l) mod v$ defines a ring homomorphism from R to $\mathbb{F}_p$, which preserves monomial structure: $\big(Y_j(\chi_1, \ldots, \chi_k)\big)(j) \bmod u = Y_j(\chi_1(j), \ldots, \chi_k(j)) \bmod u$.

By signature correctness, for each $i \in \{1, \ldots, k\}$, $\chi_i(j) \bmod u = m_i$. This induces the evaluation: $\chi'(j) \bmod u = \sum_j t_j \big(Y_j(\chi_1, \ldots, \chi_k)\big)(j) \bmod u = \sum_j t_j Y_j(\chi_1(j), \ldots, \chi_k(j)) \bmod u = \sum_j t_j Y_j(m_1, \ldots, m_k) = f(\vec{m})$.

For valid signatures, we have for each $i \in \{1, \ldots, k\}$, $\chi_i(l) \bmod v = \alpha_i$. Through the homomorphism $h(x) \mapsto h(l) \bmod v$, we derive: $\chi'(l) \bmod v = \sum_j t_j Y_j(\alpha_1, \ldots, \alpha_k) \bmod v = \omega_\tau(\langle f \rangle)$. □

5. Performance of Proposed Scheme

The core concept of a trustworthy data space is to establish a secure and efficient data circulation ecosystem among participants. The realization of this vision highly depends on the performance and cost scalability of the underlying storage infrastructure in cross-organizational interaction scenarios.

Therefore, in this experiment, a test environment is built based on a Hyperledger Fabric consortium blockchain network. The network is close to production scenarios. It aims to verify the performance of the secure storage mechanism for trustworthy data space. The core of the network consists of four Docker container nodes. These include one ordering node, two peer nodes, and one certificate authority (CA). The ordering node is responsible for transaction ordering and block generation. The two peer nodes belong to different organizations. They are used to simulate cross-organizational collaboration. The CA manages digital certificates of entities. The network runs on the Ubuntu 20.04.5 operating system. Its hardware configuration includes an Intel Xeon E5-2650 v2 @ 2.60 GHz CPU (Intel Corporation, Santa Clara, CA, USA). and 16 GB RAM. This ensures computing resources meet the test requirements. The experiment adopts an "on-chain and off-chain hybrid architecture" for data storage and processing: On the chain, Java chaincode integrated with the JPBC cryptographic library is deployed. It executes core logic. Such logic includes metadata management and polynomial commitment generation and verification. Off the chain, a MongoDB database cluster is used to store business data. The secure chaincode external interface is employed. It realizes trusted interaction between on-chain smart contracts and off-chain databases. It also realizes data consistency verification. In the performance evaluation phase, the Hyperledger Caliper benchmarking framework is used. It comprehensively measures the performance of the proposed scheme in cross-organizational data collaboration scenarios.

5.1. Communication Overhead

During the initial setup phase, the data owner deploys initialization parameters—including polynomial commitment parameters and homomorphic signature keys—to the off-chain storage and blockchain. This scheme, together with the schemes proposed by Zhang et al. [23] and Cao et al. [28], only needs to generate verification parameters once and store them on-chain. Subsequent operations do not require repeated submission of these parameters, so the communication overhead remains constant and independent of data volume. In contrast, the scheme by Guo et al. [35] requires separate calculation and transmission of verification metadata for each batch of cloud-stored data. As a result, its communication cost increases linearly with data volume. When the data volume exceeds 20 MB, the overhead surges due to cloud shard uploads.

Figure 3 show that compared with other schemes, the proposed method achieves significantly lower communication overhead during large-volume data transmission. This greatly improves the overall communication efficiency of the system.

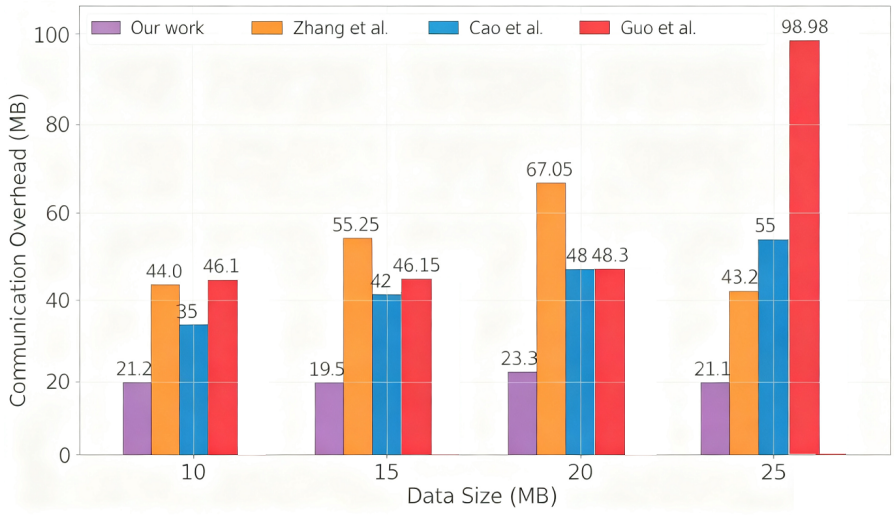

Figure 3. Communication overhead comparison [23,28,35].

5.2. Storage Overhead Evaluation

To comprehensively verify the advantages of the proposed scheme in storage efficiency, this section conducts experiments across different data scales n from four dimensions: on-chain storage, off-chain storage, signature size, and commitment size. The original business data is uniformly 20 MB of structured data.

As can be seen from the data in Table 3, the proposed scheme exhibits significant advantages in storage overhead: on-chain storage shows a linear low-growth trend, off-chain redundancy rate is controllable, signature size grows logarithmically and remains lightweight, and commitment size is fixed. These characteristics enable it to efficiently adapt to the "large-scale and dynamic" data storage requirements in trustworthy data spaces, while ensuring data security and verifiability and significantly reducing storage costs.

Table 3. Comparison of multi-dimensional storage overhead.

Data Size (n)	On-Chain Storage (KB)	Off-Chain Storage (MB)	Off-Chain Redundancy Rate (%)	Average Signature Size (Bytes)	Commitment Size (Bytes)
100	4.1	21.4	7.0	126	32
500	8.3	21.6	8.0	154	32
1000	12.5	21.8	9.0	181	32
2000	20.8	22.1	10.5	203	32
3000	29.1	22.4	12.0	222	32

The on-chain storage content includes polynomial commitment values and data index metadata. Experimental results show that on-chain storage increases linearly with the data scale n, but the growth slope is extremely low. This characteristic stems from the fact that the proposed scheme only needs to store fixed-size polynomial commitments (32 Bytes) and lightweight data indexes (each data block index is approximately 10 Bytes), which avoids the "storage bloat" problem of traditional blockchain schemes caused by storing full hash values or ciphertexts. Thus, it can maintain the efficiency of on-chain storage even in large-scale data scenarios.

Off-chain storage includes encrypted business data and polynomial coefficients. In the experiment, for 20 MB of original data, when $(n = 100)$, the off-chain storage is 21.4 MB with a redundancy rate of only 7.0%; when $(n = 3000)$, the off-chain storage is 22.4 MB with a redundancy rate of 12.0%. This indicates that the off-chain redundancy rate of the proposed scheme is always controlled at a low level and grows slowly with the increase in data scale. The core reason is that polynomial encryption does not require storing redundant shards, and only a small amount of space is needed to store polynomial coefficients. While ensuring data security, it significantly reduces off-chain storage costs.

The homomorphic signature used in the scheme grows logarithmically with the data scale n, and the growth rate gradually decreases, demonstrating the lightweight advantage of the scheme in large-scale data scenarios. Moreover, the signature size is always smaller than that of the traditional RSA signature (fixed at 256 Bytes). In contrast, the polynomial commitment size is fixed at 32 Bytes and has no correlation with the data scale n. This design is the key to the proposed scheme achieving "no on-chain storage bloat"—only a single group element needs to be stored to complete the binding of data and polynomials, fundamentally ensuring the efficiency of on-chain storage.

5.3. Off-Chain Time Cost

This subsection evaluates the off-chain storage performance of the proposed scheme. The scheme proposed by Zhang et al. [23] relies on a Trusted Execution Environment (TEE) and distributed storage. While it ensures data security, TEE introduces additional encryption and verification overhead. Meanwhile, distributed consensus increases the complexity of node interaction, resulting in relatively high storage latency and time overhead. The method by Goint et al. [36] adopts a stacked encrypted file system. Multi-layer encryption and I/O amplification issues lead to significant performance degradation, making it exhibit the highest storage latency and time overhead among all compared schemes. The approach by Guo et al. [35] employs a blockchain-based multi-cloud storage architecture. Although it achieves data redundancy, cross-system interactions between cloud nodes and the blockchain layer increase operational complexity and time consumption.

In contrast, the polynomial commitment used in this study (our work) enables efficient data integrity verification without requiring extensive cryptographic operations. Additionally, the parallel processing of data writing and commitment generation optimizes the overall time overhead. Therefore, this scheme achieves the optimal balance between performance and security, with significantly lower storage latency and time overhead compared to the other three schemes.

Figure 4 presents a comparison of the time overhead among the four schemes under the following conditions: consistent data format and original content before encryption, a fixed test data size of 20 MB, 500 independent storage operations executed for each scheme, and the exclusion of the top 5% and bottom 5% of extreme values when calculating the average overhead.

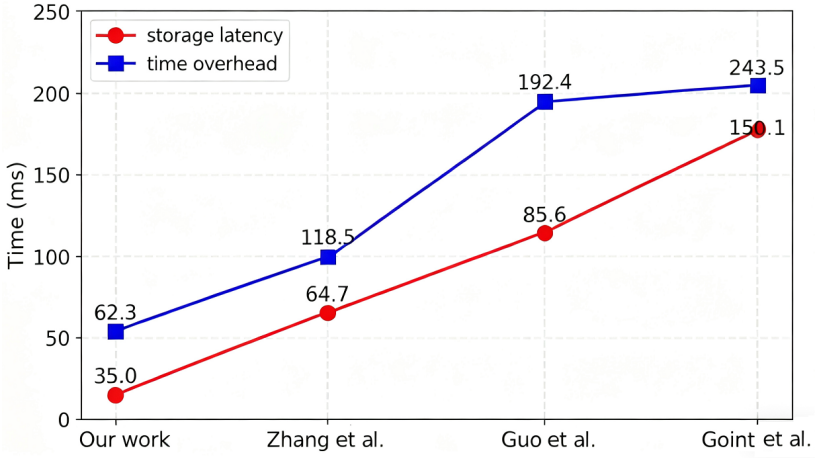

Figure 4. Comparison of off-chain storage time cost [23,35,36].

5.4. On-Chain Time Cost

In the Hyperledger Fabric consortium blockchain environment, on-chain operations focus on the execution efficiency of Java chaincode integrated with the JPBC cryptographic library. These operations cover two core types: polynomial commitment verification and homomorphic signature verification. The operations are executed on two peer nodes from different organizations, with a "2-out-of-2" endorsement policy adopted to simulate cross-organizational collaboration scenarios. Performance data is collected through 500 independent tests using the Hyperledger Caliper benchmarking framework. The specific time overhead is presented in Table 4 below.

Table 4. The latency and throughput of the two operations.

Operation	Sending Rate (TPS)	Max Latency (ms)	Min Latency (ms)	Avg Latency (ms)	Throughput (TPS)
Commitment Verification 1	298.36	52.17	22.35	46.89	9.82
Commitment Verification 2	295.41	51.82	21.98	46.53	9.91
Commitment Verification 3	297.15	51.59	22.11	46.72	9.78
Signature Verification 1	362.85	28.43	12.57	14.89	15.93
Signature Verification 2	365.27	28.16	12.73	14.72	15.98
Signature Verification 3	368.51	28.32	12.64	14.58	15.89

The experimental results show that the average latency of polynomial commitment verification (46.71 ms on average) is significantly higher than that of homomorphic signature verification (14.73 ms on average) by approximately 2.17 times. The core reason lies in the significant difference in computational complexity between the two operations. For polynomial commitment verification, it is necessary to generate auxiliary polynomials and auxiliary parameters in real time. It is also necessary to perform bilinear pairing operations to complete equation verification. Such operations involve complex group element multiplication and pairing operations. These operations consume considerable computational resources of the cryptographic library (JPBC). In contrast, homomorphic signature verification only requires checking norm constraints and message consistency. It does not need to generate additional auxiliary parameters. Its logic is simpler. Its dependence on computational resources is lower. This results in significantly lower latency.

Figure 5 illustrates the comparison of time overhead between homomorphic signature generation and polynomial commitment generation as the data scale n increases. This study tests the characteristics of signature and commitment generation time. It also tests their sizes under varying data scale n. The study aims to quantify the scalability of the scheme's storage efficiency and performance in practical scenarios. It also verifies the

scheme's advantage: the signature size does not increase linearly with n. Meanwhile, it clarifies the controllable range of time overhead. This is to meet the efficiency requirements of trustworthy data space for on-chain operations of large-scale data.

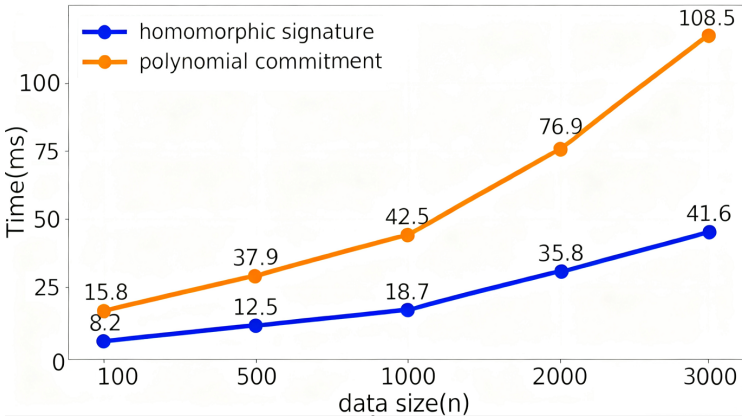

Figure 5. Time cost comparison of two operations.

5.5. Overhead of Data Update

In the context of experimental analysis, we classify update operations into three types: modification, insertion, and deletion. Figure 6 presents a comparison chart of the costs for these three update operations.

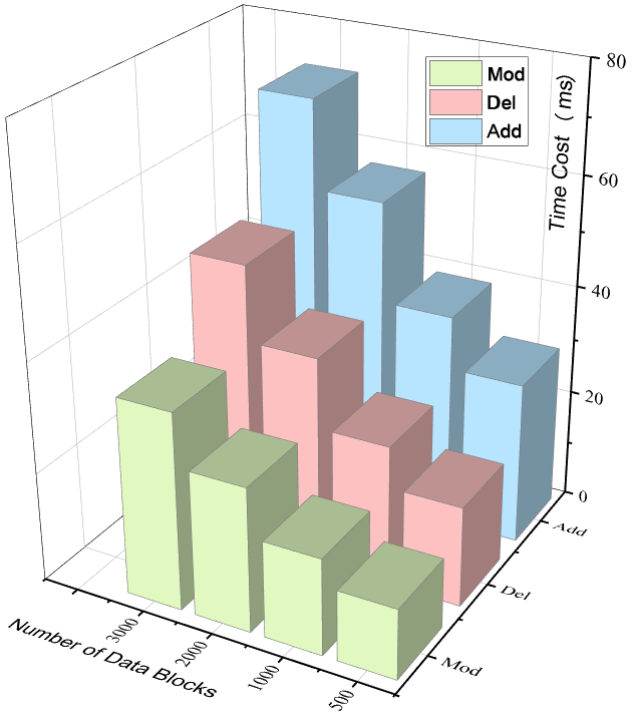

Figure 6. Data update operation time cost of our work.

Modification operation: Modifying a data block only updates business data content. It does not change the total number of blocks. The metadata index remains stable. Homomorphic signatures need not be regenerated. Only the polynomial commitment is updated via smart contracts to match modified data. Overhead is concentrated on partial adjustment of commitment coefficients. This makes it the least resource-intensive operation.

Delete operation: Deleted data blocks are not physically removed. They are only marked in on-chain metadata. The polynomial commitment must be recalculated for

remaining valid data. Index table entries are updated synchronously. Homomorphic signatures remain valid for the dataset. This results in slightly higher overhead than modification.

Insertion operation: Adding a new data block requires updating the metadata index. It also requires recalculating the polynomial commitment for the full dataset. Additionally, it needs to generate and aggregate a homomorphic signature for the new block. These steps make it the most resource-intensive dynamic management operation.

The update overhead of the scheme in this paper is significantly lower than that of the schemes in Pandey et al. [29] and Ren et al. [37], as shown in Figure 7.

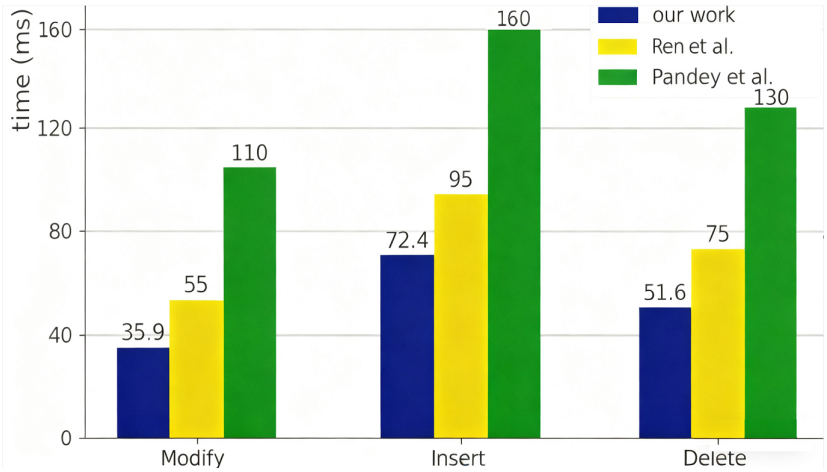

Figure 7. Data update operation time cost comparison chart [29,37].

Firstly, this scheme adopts a partial update mechanism for polynomial commitments. Only the corresponding coefficients need to be adjusted to maintain consistency after data modification or deletion. This avoids two issues. One is "sub-vector reconstruction required by sub-vector commitments to maintain spatiotemporal data correlation" in Ren et al. [37]. The other is "full ciphertext recalculation caused by the strong coupling between permissions and content in attribute-based encryption (ABE)" in Pandey et al. [29]. This fundamentally reduces computational complexity.

Secondly, the decoupled design of homomorphic signatures and metadata eliminates the need to regenerate signatures when data is updated. Only the polynomial commitments are used to synchronize state changes. This reduces additional overhead. The overhead comes from "spatiotemporal index adjustment" in Ren et al. [37] and "full update of hash linked lists" in Pandey et al. [29].

Finally, the on-chain and off-chain hybrid architecture enables separate storage of business data and metadata. During cross-organizational collaboration, only metadata changes need to be synchronized. This avoids the synchronization cost of uploading full-volume data to the chain. Thus, the scheme is more suitable for meeting the efficiency requirements of high-frequency dynamic update scenarios in trustworthy data space.

5.6. Fault Tolerance Testing

In cross-organizational collaboration within a trustworthy data space, peer node failure of a single organization is a typical risk, which directly affects the continuity of core services such as cross-organizational data query and sharing. This section verifies the fault tolerance of the proposed scheme in such scenarios through experiments, focusing on testing service availability, data consistency, and recovery efficiency during failures to demonstrate the advantages of distributed verification using polynomial commitments and homomorphic signatures.

The experiment first conducts a benchmark test under normal conditions to establish a reference. The test includes 100 cross-organizational operations. These operations are as follows: 60 times of "Organization A queries authorized data from Organization B", 40 times of "internal data updates within Organization A", and 50 times of "Organization A shares new data with Organization B". The results show the following: The average response time for queries is 65 ms, 90 ms for internal updates, and 130 ms for cross-organizational sharing; the success rate of all operations is 100%; the matching rate between on-chain polynomial commitments and off-chain data is 100%. This verifies the stability under normal conditions.

Subsequently, the offline failure of Organization B's peer node is simulated. The 100 operations mentioned above are repeated after stopping Organization B's node. The results show the following: the success rate of internal updates in Organization A remains 100%, with a response time fluctuation of $\leq$10 ms; the success rate of Organization A's query for data from Organization B reaches 92%; all cross-organizational sharing requests are cached in the client queue, with a temporary storage rate of 100% and no data loss; no on-chain-off-chain data inconsistency occurs during the failure.

In the failure recovery phase, Organization B's node is restarted. After that, the cached cross-organizational sharing requests are automatically submitted, and all are successful, with an average processing time of 142 ms; the consistency matching rate of randomly sampled data is 100%; the full service recovery time is 23 s. This verifies efficient self-healing capability.

The experimental results are shown in Table 5. Internal updates of Organization A are hardly affected. The success rate of cross-organizational queries remains above 90%. Sharing requests achieves zero loss through the temporary storage mechanism. This indicates that the proposed scheme can still ensure service continuity when a single organization's node fails. The incremental verification mechanism during the recovery phase significantly reduces the overhead of consistency maintenance.

Table 5. Fault tolerance performance in single peer node failure scenario.

Phase	Query	Update	Consistency
Benchmark	Average response time 65 ms, success rate 100%	Average response time 90 ms, success rate 100%	Matching rate 100%
Fault	Success rate 92%	Success rate 100%, response time fluctuation $\leq$10 ms	No inconsistency
After Recovery	-	-	Matching rate 100%
Recovery Time	-	-	Full business recovery in 23 s

6. Conclusions

This paper presents a data security storage mechanism for trustworthy data spaces, addressing their core challenge: balancing data sovereignty and verifiable sharing. Integrating polynomial commitment and homomorphic signatures, it verifies data integrity and ensures consistency after dynamic operations. Security analysis confirms its correctness and safety; performance tests demonstrate efficiency superior to existing solutions, meeting large-scale storage needs for security and speed. Notably, the current test environment differs from real large-scale production systems in network scale and node distribution, potentially challenging boundary performance under extreme concurrency or wide-area distribution. Future work will focus on two directions: exploring adaptive parameter configuration in dynamic networks and developing lightweight cross-chain protocols to reduce synchronization overhead between heterogeneous blockchains while maintaining security, fostering efficient trusted data circulation.

Author Contributions: Conceptualization, X.Y. and Q.L.; methodology, X.Y. and Q.L.; validation, X.Y., Q.L. and J.X.; formal analysis, X.Y.; writing—original draft preparation, X.Y. and Q.L.; writing—review and editing, Q.C.; supervision, Q.C.; project administration, J.X.; funding acquisition, J.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research Fund of Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education (EBME25-F-08).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Qinghong Cao was employed by the Shanghai Stock Exchange Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. EOSC. EOSC Future Results. 2024. Available online: https://eoscfuture.eu/results/ (accessed on 30 June 2024).
2. OpenAIRE. Openaire Guidelines. 2024. Available online: https://guidelines.openaire.eu/en/latest (accessed on 31 July 2024).
3. Otto, B.; Hompel, M.T.; Wrobel, S. International Data space: Reference architecture for the digitization of industries. In *Digital Transformation*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 109–128. [CrossRef]
4. Data Space Business Alliance. Technical Convergence. 2022. Available online: https://data-spaces-business-alliance.eu/wp-content/uploads/dlm_uploads/DSBA-Technical-Convergence.pdf (accessed on 26 September 2022).
5. Trustworthy Data Space—Technology Architecture. 2025. Available online: http://jkzgnews.com/d/file/2025-05-12/1747035305429248.pdf (accessed on 30 April 2025).
6. Huber, M.; Wessel, S.; Brost, G.; Menz, N. Building Trust in data space. In *Designing Data Space*; Otto, B., ten Hompel, M., Wrobel, S., Eds.; Springer: Cham, Switzerland, 2022; pp. 147–164.
7. Hackel, S.; Makohl, M.E.; Petrac, S. Trustworthy Data Exchange: Leveraging Linked Data for Enhanced IDS Certification. In Proceedings of the INFORMATIK 2024: 9th IACS Workshop, Wiesbaden, Germany, 24–26 September 2024; Gesellschaft für Informatik e.V.: Bonn, Germany, 2024; pp. 1977–1987.
8. Ren, Y.; Leng, Y.; Qi, J.; Sharma, P.K.; Wang, J.; Almakhadmeh, Z.; Tolba, A. Multiple Cloud Storage Mechanism Based on Blockchain in Smart Homes. *Future Gener. Comput. Syst.* **2021**, *115*, 304–313. [CrossRef]
9. Lou, J.T.; Bhat, S.A.; Huang, N.F. Blockchain-based privacy-preserving data-sharing framework using proxy re-encryption scheme and interplanetary file system. *Peer-to-Peer Netw. Appl.* **2023**, *16*, 2415–2437. [CrossRef]
10. Gai, K.; Guo, Y.; Yu, J.; Chan, W.; Zhu, L.; Zhang, Y. CAPE: Commitment-Based Privacy-Preserving Payment Channel Scheme in Blockchain. *IEEE Trans. Dependable Secur. Comput.* **2025**, *22*, 3977–3992. [CrossRef]
11. Wang, J.; Chen, J.; Ren, Y.; Sharma, P.; Alfarraj, O.; Tolba, A. Data Security Storage Mechanism Based on Blockchain Industrial Internet of Things. *Comput. Ind. Eng.* **2022**, *164*, 107903. [CrossRef]
12. Wang, K.; Yan, Y.; Guo, S.; Wei, X.; Shao, S. On-Chain and Off-Chain Collaborative Management System Based on Consortium Blockchain. In *Advances in Artificial Intelligence and Security*; Sun, X., Zhang, X., Xia, Z., Bertino, E., Eds.; Springer: Cham, Switzerland, 2021; pp. 197–210.
13. Ren, Y.; Lv, Z.; Xiong, N.N.; Wang, J. HCNCT: A Cross-Chain Interaction Scheme for the Blockchain-Based Metaverse. *ACM Trans. Multimed. Comput. Commun. Appl.* **2024**, *20*, 188. [CrossRef]
14. Bonura, S.; Carbonare, D.D.; Díaz-Morales, R.; Fernández-Díaz, M.; Morabito, L.; Muñoz-González, L.; Napione C.; Navia-Vázquez, A.; Purcell M. Privacy-Preserving Technologies for Trusted data space. In *Technologies and Applications for Big Data Value*; Curry, E., Auer, S., Berre, A.J., Metzger, A., Perez, M.S., Zillner, S., Eds.; Springer: Cham, Switzerland, 2022; pp. 111–134.
15. Deepa, N.; Pham, Q.V.; Nguyen, D.C.; Bhattacharya, S.; Prabadevi, B.; Gadekallu, T.R.; Maddikunta, P.K.R.; Fang, F.; Pathirana P.N. A Survey on Blockchain for Big Data: Approaches, Opportunities, and Future Directions. *Future Gener. Comput. Syst.* **2022**, *131*, 209–226. [CrossRef]
16. Jadhav, S.; Choudhari, G.; Bhavik, M.; Bura, R.; Bhosale, V. A Decentralized Document Storage Platform using IPFS with Enhanced Security. In Proceedings of the 2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, 23–24 August 2024; pp. 1–11.
17. Esposito, C.; De Santis, A.; Tortora, G.; Chang, H.; Choo, K.R. Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy? *IEEE Cloud Comput.* **2018**, *5*, 31–37. [CrossRef]
18. Wang, S.; Wang, X.; Zhang, Y. A Secure Cloud Storage Framework with Access Control Based on Blockchain. *IEEE Access* **2019**, *7*, 112713–112725. [CrossRef]

19. Yang, C.; Tan, L.; Shi, N.; Xu, B.; Cao, Y.; Yu, K. AuthPrivacyChain: A Blockchain-Based Access Control Framework with Privacy Protection in Cloud. *IEEE Access* **2020**, *8*, 70604–70615. [CrossRef]
20. Kuznetsov, O.; Yezhov, A.; Yusiuk, V.; Kuznetsova, K. Scalable Zero-Knowledge Proofs for Verifying Cryptographic Hashing in Blockchain Applications. *arXiv* **2024**, arXiv:2407.03511. [CrossRef]
21. Liang, W.; Fan, Y.; Li, K.C.; Zhang, D.; Gaudiot, J.L. Secure Data Storage and Recovery in Industrial Blockchain Network Environments. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6543–6552. [CrossRef]
22. Sun, Z.; Han, D.; Li, D.; Weng, T.; Li, K.; Mei, X. MedRSS: A Blockchain-Based Scheme for Secure Storage and Sharing of Medical Records. *Comput. Ind. Eng.* **2023**, *183*, 109521. [CrossRef]
23. Zhang, Y.; Xiong, L.; Li, F.; Niu, X.; Wu, H. A Blockchain-Based Privacy-Preserving Auditable Authentication Scheme with Hierarchical Access Control for Mobile Cloud Computing. *J. Syst. Archit.* **2023**, *142*, 102949. [CrossRef]
24. Liu, C.; Guo, H.; Xu, M.; Wang, S.; Yu, D.; Yu, J. Extending On-Chain Trust to Off-Chain—Trustworthy Blockchain Data Collection Using Trusted Execution Environment (TEE). *IEEE Trans. Comput.* **2022**, *71*, 3268–3280. [CrossRef]
25. Liu, C.; Hou, C.; Jiang, T.; Ning, J.; Qiao, H.; Wu, Y. FACOS: Enabling Privacy Protection Through Fine-Grained Access Control with On-Chain and Off-Chain System. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 7067–7074. [CrossRef]
26. Ding, Y.; Wu, Z.; Miao, Y.; Xie, L.; Ding, M. Genuine On-Chain and Off-Chain Collaboration: Achieving Secure and Non-Repudiable File Sharing in Blockchain Applications. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 1802–1816. [CrossRef]
27. Zhang, W.; Wei, L.; Li, S.; Liu, Y.; Cheung, S.C. ÐArcher: Detecting On-Chain-Off-Chain Synchronization Bugs in Decentralized Applications. In Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis, Virtual, 11–17 July 2021; pp. 553–565.
28. Cao, Y.; Cao, J.; Bai, D.; Wen, L.; Liu, Y.; Li, R. MAP the Blockchain World: A Trustless and Scalable Blockchain Interoperability Protocol for Cross-Chain Applications. In Proceedings of the ACM Web Conference, Sydney, Australia, 28 April–2 May 2025; pp. 717–726.
29. Pandey, S.; Rishiwal, V.; Jat, D.S.; Yadav, P.; Yadav, M.; Jain, A. Towards Securing the Digital Document Using Blockchain Technology with Off-Chain Attribute Based Encryption Framework. In Proceedings of the International Conference on Emerging Trends in Networks and Computer Communications, Windhoek, Namibia, 23–25 July 2024; pp. 857–864.
30. Javed, M.U.; Rehman, M.; Javaid, N.; Aldegheishem, A.; Alrajeh, N.; Tahir, M. Blockchain-Based Secure Data Storage for Distributed Vehicular Networks. *Appl. Sci.* **2020**, *10*, 2011. [CrossRef]
31. Miyachi, K.; Mackey, T.K. hOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design. *Inf. Process. Manag.* **2021**, *58*, 102535. [CrossRef]
32. Eberhardt, J.; Tai, S. On or Off the Blockchain? Insights on Off-Chaining Computation and Data. In *Service-Oriented and Cloud Computing*; De Paoli, F., Schulte, S., Broch Johnsen, E., Eds.; Springer: Cham, Switzerland, 2017; pp. 3–15.
33. Ahmadjee, S.; Bahsoon, R. A Taxonomy for Understanding the Security Technical Debts in Blockchain Based Systems. *arXiv* **2019**, arXiv:1903.03323. [CrossRef]
34. Kiayias, A.; Russell, A.; David, B.; Oliynykov, R. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. In *Advances in Cryptology—CRYPTO 2017*; Katz, J., Shacham, H., Eds.; Springer: Cham, Switzerland, 2017; pp. 357–388.
35. Guo, X.; Xiong, Z.; Chen, J.; Chen, D. A Secure, Blockchain-Based Data Storage Scheme for Cloud Environments. In Proceedings of the International Conference on Computer, Artificial Intelligence, and Control Engineering (CAICE 2023), Hangzhou, China, 17–19 February 2023; p. 1264524.
36. Goint, M.; Bertelle, C.; Duvallet, C. Secure Access Control to Data in Off-Chain Storage in Blockchain-Based Consent Systems. *Mathematics* **2023**, *11*, 1592. [CrossRef]
37. Ren, Y.; Huang, D.; Wang, W.; Yu, X. BSMD: A Blockchain-Based Secure Storage Mechanism for Big Spatio-Temporal Data. *Future Gener. Comput. Syst.* **2023**, *138*, 328–338. [CrossRef]
38. Shen, J.; Liu, D.; Chen, X.; Huang, X.; Chen, J.; Zhang, M. Secure Publicly Verifiable Computation with Polynomial Commitment in Cloud Computing. In *Information Security and Privacy*; Susilo, W., Yang, G., Eds.; Springer: Cham, Switzerland, 2018; pp. 417–430.
39. Johnson, R.; Molnar, D.; Song, D.; Wagner, D. Homomorphic Signature Schemes. In *Topics in Cryptology—CT-RSA 2002*; Preneel, B., Ed.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 244–262.
40. Boneh, D.; Drake, J.; Fisch, B.; Gabizon, A. Halo Infinite: Recursive zk-SNARKs from Any Additive Polynomial Commitment Scheme. *Cryptol. ePrint Arch.* **2020**. Available online: https://eprint.iacr.org/2020/1536 (accessed on 3 November 2025).
41. Boneh, D.; Freeman, D.M. homomorphic signature for Polynomial Functions. In *Advances in Cryptology—EUROCRYPT 2011*; Paterson, K.G., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 149–168.