electronics

MDPI

Article

PBTMS: A Blockchain-Based Privacy-Preserving System for Reliable and Efficient E-Commerce

Ruochi Zhang [1], Yi Li [1,*] and Li Fang [2]

[1] Key Laboratory of Universal Wireless Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China; zhangruochi99@bupt.edu.cn
[2] Beijing Laboratory of Advanced Information Networks, Beijing University of Posts and Telecommunications, Beijing 100876, China; fangli@bupt.edu.cn
* Correspondence: liyi@bupt.edu.cn; Tel.: +86-13401014667

Abstract: With the development of communication infrastructure and the popularity of smart devices, e-commerce is presenting in more diverse forms and attracting the attention of more and more users. Since e-commerce transactions usually involve sensitive information of a large number of users, privacy and security have become increasingly important issues. Despite certain advantages (e.g., trading security), the privacy protection capability and efficiency of blockchain is still limited by some key factors, especially of its architecture. In this paper, we propose a blockchain-based privacy protection system named PBTMS that integrates zero-knowledge proofs, hybrid encryption, and Pedersen commitments as foundational mechanisms to ensure robust privacy protection for transaction data and user information. To achieve secure, reliable, and efficient e-commerce transactions, the PBTMS employs blockchain technology and consensus mechanisms to enable distributed storage, thereby mitigating single points of failure and addressing the risks posed by malicious nodes. Moreover, by integrating on-chain storage with off-chain computation, the system substantially reduces blockchain-related overheads, including processing time, gas consumption, and storage costs. This design establishes the PBTMS as a highly adaptable and efficient system for the evolving requirements of secure and privacy-preserving e-commerce platforms. Theoretical analysis and experimental validation demonstrate that PBTMS reduces decryption and authentication times by 79.2% and 52.6%, respectively, while cutting encrypted data size by 52.5% and overall gas consumption by 55.4%, outperforming state-of-the-art solutions. These results indicate that PBTMS is a reliable and efficient system for secure e-commerce transaction platforms and provides a novel approach to enhancing privacy protection in e-commerce.

Keywords: blockchain; smart contracts; zero knowledge; data privacy; e-commerce trading

check for updates

Academic Editor: Subir Halder

Received: 25 January 2025
Revised: 5 March 2025
Accepted: 11 March 2025
Published: 17 March 2025

Citation: Zhang, R.; Li, Y.; Fang, L. PBTMS: A Blockchain-Based Privacy-Preserving System for Reliable and Efficient E-Commerce. *Electronics* **2025**, *14*, 1177. https://doi.org/10.3390/electronics14061177

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the development of communication infrastructure and the popularity of smart devices, e-commerce is presenting in more diverse forms, such as live streaming [1,2] and cross-border transactions [3,4]. These e-commerce activities have some specific characteristics, such as real-time interaction and personal recommendation, and thus greatly change people's shopping habits. Correspondingly, e-commerce is attracting the attention of more and more users, so the scale of transactions and the number of users have significantly expanded. The frequency of users' online transactions and the use of personal information have increased substantially [5].

Since e-commerce transactions usually involve sensitive information of a large number of users, privacy and security have become increasingly important issues in today's rapidly evolving e-commerce landscape [6–9]. The information provided by users in online trading is at risk of being maliciously accessed by unauthorized third parties, which may lead to serious privacy leaks [10–12]. Notably, Albshaier et al. [13] noted that due to the lack of transparency and adequate security measures for e-commerce platforms, trading data and user information stored on remote servers are highly vulnerable to cyber-attacks and unauthorized access, thereby increasing the risk of data theft or leakage. Kim et al. [14] pointed out that fraudulent trading by malicious users, such as fake purchases or refund scams, further exacerbates the security concerns of online trading. In particular, numerous data breach incidents have occurred in recent years, which have aroused great concern among people. For example, Equifax experienced a major data breach during which hackers managed to pilfer sensitive information belonging to more than 146.6 million consumers in 2017 [15]. Another latest example is that Change Healthcare, a data processing firm owned by the UnitedHealth Group, was the target of a ransomware cyberattack in 2024 and thus suffered losses of more than USD 78 million [16].

These security issues indicate the need for more comprehensive security measures to protect users' personal information and trading data in e-commerce. Fortunately, blockchain has been demonstrated by prior studies as an effective security technique to improve the security of users' information and trading data in e-commerce [12,17]. Compared to traditional privacy protection mechanisms that offer anonymity by scrambling the direct connection between the sender and receiver, the blockchain can not only reduce the dependence on single points of failure through distributed nodes but also protect user identity information by allowing authorized accounts to access the data [13,18]. Because of these advantages, blockchain has been successfully applied to multiple scenes in e-commerce, such as trusted trading platforms [19,20], reputation systems [21–24], secure search [25], identity authentication [26], price auditing [27], and sales mode selection [28]. Particularly, Dahal [29] pointed out that since trading data in e-commerce are special due to immutable and time-stamped characteristics, a blockchain-based guarantee of integrity, anti-tampering, and traceability is necessary and challenging.

Therefore, this paper integrates blockchain technology for data storage and proof records to enhance system security. Blockchain's decentralization, immutability, and smart contracts reduce the risk of data tampering, ensure integrity, and provide transparent transaction tracking, which complements traditional security strategies for e-commerce systems.

Despite certain advantages, the privacy protection capability of blockchain is still limited by some key factors, especially its architecture. For example, Taherdoost and Madanchian [30] emphasized that the distributed ledger and decentralized nature of blockchain can threaten user anonymity, which becomes a major limiting factor for blockchain in e-commerce applications. Ghesmati et al. [31] argued that blockchain may be cracked by on-chain analysis and thus increases the computation costs and compliance risks. To this end, research has gradually shifted to more advanced privacy-preserving technologies like ring signatures, stealth addresses, confidential trading, and zero-knowledge proofs. These technologies protect the privacy of identities and trading amounts while guaranteeing the verifiability of trading. For example, Zerocash is the first blockchain-based security trading platform that enables completely anonymous trading through the zk-SNARKs protocol [32]. Monero [33] implements anonymous cryptocurrency with enhanced accountability for a high level of privacy protection based on the CryptoNote protocol and M-LSAG. Zether [34] combines smart contracts and non-interactive zero-knowledge proofs to provide anonymous trading support for Ethereum users. Furthermore, BlockMaze [35] protects user privacy through a dual balance model based on zk-SNARKs.

In this paper, our main objective is to design a reliable and efficient blockchain-based e-commerce trading system that ensures privacy protection for transactions and provides data security for users' information. However, the practical implementation of such a system faces several significant challenges that must be addressed to fully realize the potential of blockchain technology in e-commerce. Specifically, the system is confronted with the following three key challenges:

- Ensuring node credibility: Blockchain's reliance on multiple nodes for transaction validation adds complexity, with reward-and-punishment mechanisms and limited interoperability between platforms, which can undermine node reliability and system credibility.
- Node abuse in data access: Blockchain's transparency can lead to vulnerabilities, where nodes with access to sensitive data may exploit this privilege, potentially altering transaction history, breaching data integrity, and threatening user privacy.
- Balancing privacy and performance: Privacy-preserving techniques like zero-knowledge proofs protect sensitive data but add performance overhead, challenging the balance between privacy and system efficiency without sacrificing scalability or user experience.

To this end, we propose a comprehensive solution to address these challenges. By enhancing privacy protection and data security, mitigating the risks posed by malicious nodes, and achieving an optimal balance between performance and privacy, we aim to establish a reliable and efficient blockchain ecosystem tailored for e-commerce transactions.

We propose a blockchain-based e-commerce transaction system, PBTMS, to address challenges such as node credibility, privilege abuse, and the balance between privacy protection and performance. The system prevents third-party access to user and transaction information while ensuring the validity and privacy of transactions. It achieves this through privacy protection mechanisms, access control, and tolerance to malicious nodes. By leveraging blockchain technology, PBTMS ensures secure data transmission and reliable transaction storage. It integrates hybrid encryption, zero-knowledge proofs, digital signatures, and Pedersen commitments to provide high privacy protection and data security for online physical goods transactions. The main reasons why we use these security techniques include the following:

- Pedersen commitment: Due to its strong tamper-resistance and content-hiding properties, Pedersen's commitment ensures transaction data privacy and consistency verification in blockchain-based e-commerce. It prevents information leakage while maintaining data integrity and transparency, meeting the system's privacy protection requirements.
- Schnorr signatures: Schnorr signatures provide high security while reducing computational overhead. They enable efficient signature generation and verification in blockchain environments, improving transaction verification efficiency and supporting high performance for large-scale transactions, ensuring the e-commerce platform maintains privacy protection without compromising system efficiency.
- Bulletproofs: As a non-interactive zero-knowledge proof technology, Bulletproofs verify data validity without revealing specific data. They are ideal for verifying users' purchasing power in e-commerce systems. Compared to other zero-knowledge proofs, Bulletproofs offer smaller proof sizes and higher computational efficiency, reducing storage needs and enhancing performance while maintaining strong security.
- ECC and AES shared key technology: ECC is widely used for its shorter key lengths and efficient computation. Combined with AES, it provides robust encryption to ensure data security during transactions. In blockchain-based e-commerce systems, the combination of ECC and AES offers an optimal balance of security and performance,

making it an ideal choice for privacy protection in high-frequency trading and large-scale data processing.

The system adopts an on-chain/off-chain architecture to simplify data storage and distribution processes, improving efficiency. PBTMS uses a decentralized design, including an off-chain committee composed of blockchain miners, integrating consensus mechanisms and reputation systems to reduce the impact of malicious nodes. It strengthens node behavior constraints while addressing the trade-off between privacy and performance, establishing a reliable trust mechanism for e-commerce transactions.

In summary, our main contributions in this paper are listed as follows:

- We propose a blockchain-based trading system with on-chain storage and off-chain computation, which effectively protects private information and balances performance. Smart contracts store trading data identifiers and access authorizations, while complex operations such as encryption, verification, and computation are executed off-chain. Fine-grained data access control is achieved through shared keys.
- The concrete implementation of hybrid encryption and multi-level cryptography is proposed. For data privacy, hybrid chunked data encryption combining ECC and AES ensures the security of trading information and transaction amounts. To enhance proof mechanisms, Schnorr digital signatures, bulletproof range proofs, and Pedersen's commitment are employed, thereby offering both efficiency and robust protection against information leakage. Meanwhile, in terms of storage, the integration of distributed storage mechanisms through smart contracts and IPFS ensures data tamper resistance while mitigating the challenges of large-scale on-chain storage. This multi-faceted approach effectively balances security, efficiency, and scalability.
- We conducted extensive experiments to evaluate the proposed system in complex e-commerce scenarios. The results show significant efficiency improvements, with decryption and authentication times reduced by 79.2% and 52.6%, respectively, and overall gas consumption decreased by 55.4%, lowering operational costs. Our findings highlight the PBTMS's effectiveness in ensuring privacy and security while demonstrating excellent scalability, practicality, and stability for secure and efficient e-commerce systems.

The remainder of this paper is structured as follows. Section 2 reviews the related work on privacy protection in e-commerce based on blockchain. Section 3 introduces the leading technologies used in this paper. Section 4 introduces the system overview, workflow, and enemy model. Section 5 presents the algorithm flow of the system in detail. In Section 6, we describe the safety and performance analyses. In Section 7, we experimentally evaluate the performance of the system and analyze the results. In Section 8, we summarize our main work and discuss future research.

2. Related Work

In this section, we introduce the related work on privacy protection under blockchain and e-commerce, mainly including blockchain for e-commerce and privacy enhancement in blockchain-based e-commerce.

2.1. Blockchain for Privacy Protection of E-Commerce

In recent years, blockchain technology's continuous development and expansion have demonstrated significant potential in privacy protection for e-commerce. The typical application scenes of blockchain in e-commerce include trusted trading platforms [19,20,36], reputation systems [21–24], secure search [25], identity authentication [26], price auditing [27], and sales mode selection [28]. For example, Li et al. [24] proposed a privacy-preserving reputation system based on blockchain for e-commerce, which supports all e-commerce

platforms to collaboratively share users' reputations and protect users' identities by resisting multiple rating attacks. Takakubo et al. [27] utilized smart contract technologies to propose a price auditing system, which solves a price discrimination problem and thus offers the secure storage of product data and high-quality e-commerce services. Li et al. [28] analyzed the impact of blockchain on strategic and member operational decisions of e-commerce supply chains and derived the conditions of production and blockchain operating cost to determine a win–win outcome among supply chain members.

In particular, blockchain offers trusted e-commerce physical commodity trading systems by solving two core issues for the feasibility and widespread adoption of blockchain applications, such as trading validity and fairness [19,20,37–39]. Xie et al. [19] designed a trusted blockchain-based framework for higher credible trading, which utilizes a peer blockchain architecture to implement transaction storage and a strong consensus algorithm to ensure secure transactions. Liu et al. [20] designed a blockchain-based normalized autonomous transaction settlement system for IoT-related e-commerce, which contains a three-layer sharding blockchain network for improving transaction efficiency and system scalability and a searchable decentralized public key encryption protocol for uncovering illegal and criminal transactions. In addition, there are other security protocols for transaction protection and data security in e-commerce, such as ZKCP verifying payment accuracy and legitimacy without transaction details [37], RZKPB [38] enhancing trading fairness and privacy without sensitive financial information storage, and FairSwap offering digital commodity trading under secure information exchange of digital assets and services [39]. However, the above protocols still rely on sensitive data in e-commerce transactions. That is, zero-knowledge proof in ZKCP is only applied to trading data, while payment verification requires disclosing actual data. In RZKPB, commitment values and related verification information must be recorded on-chain, requiring extra storage.

2.2. Privacy Protection Enhancement in Blockchain-Based E-Commerce

Blockchain-based privacy-preserving algorithms can be enhanced by integrating advanced security techniques such as perturbation, encryption (e.g., differential privacy and homomorphic encryption), Tor networks, onion routing, zero-knowledge proofs (ZKPs), and trusted hardware. These techniques aim to ensure data security, communication privacy, and smart contract confidentiality [40]. However, they face significant challenges, including high computational overhead, reliance on trusted third parties, and performance limitations. For instance, AIDTHS [41] employs identity threshold signatures to conceal the signer's identity but relies on trusted third parties. LedgerMaze [42] uses efficient zero-knowledge proofs to hide trading information, but its performance is affected by the amount of information it contains. MedShield et al. [43] proposed a blockchain and zero-knowledge proof-based medical data trading scheme, which depends on fully trusted regulators and assumes cloud servers to be semi-trusted, thereby introducing trust risks.

Compared to traditional privacy-preserving methods, zero-knowledge proofs (ZKPs) offer distinct advantages in ensuring data privacy while enabling efficient verification and reducing computational overhead. In blockchain and decentralized applications (DApps), ZKPs, particularly zk-SNARKs (succinct non-interactive zero-knowledge proofs), demonstrate immense potential. They provide clear advantages in proof generation and verification times, with zk-SNARKs exhibiting linear time complexity for proof generation. This means that as data volume increases, the proof generation time remains manageable, avoiding computational bottlenecks. Additionally, zk-SNARKs offer small proof sizes and low space complexity, making them ideal for efficient storage and rapid verification.

As shown in Table 1, compared to other privacy-preserving technologies, zk-SNARKs excel in computational efficiency, with proof generation time of O(n), verification time

of O(1), and small proof sizes, making them highly suitable for efficient storage and fast verification. Table 2 demonstrates the security advantages of zk-SNARKs, which exhibit strong resistance to replay attacks, tampering, and man-in-the-middle attacks, effectively safeguarding data integrity and privacy. Considering computational efficiency, security, and storage requirements, zk-SNARKs emerge as the preferred solution in privacy-preserving technologies, particularly for applications that require efficient verification and robust privacy protection.

Table 1. Comparison of computational overhead for privacy-preserving techniques (homomorphic encryption (HE), ring signatures (RS), and coin join protocol (CJP)).

Technique	Proof Generation Time	Verification Time	Proof Size
Zk-SNARK	$O(n)$	$O(1)$	Small
Zk-STARK	$O(n \log n)$	$O(1)$	Larger
HE	$O(n^2)$	$O(n^2)$	Larger
RS	$O(n)$	$O(1)$	Small
CJP	$O(n)$	$O(1)$	Medium

Table 2. Comparison of resilience level of privacy-preserving techniques to various attacks (homomorphic encryption (HE), ring signatures (RS), and coin join protocol (CJP)).

Technique	Replay Attacks	Man-in-the-Middle Attacks	Modification Attacks	Selective Disclosure Attacks	Side-Channel Attacks
Zk-SNARK	Strong	Strong	Strong	Strong	Strong
Zk-STARK	Strong	Strong	Strong	Strong	Strong
HE	Strong	Moderate	Strong	Moderate	Moderate
RS	Moderate	Moderate	Moderate	Moderate	Moderate
CJP	Moderate	Moderate	Moderate	Moderate	Moderate

In addition to privacy protection in blockchain-based e-commerce, access control and malicious node tolerance mechanisms have also become significant areas of focus. Blockchain-based permission control provides an overview of defining access rules through smart contracts by role-, attribute-, or policy-based control mechanisms [44]. BlockAuth offers storing access policies in smart contracts and decentralized authentication and pseudo-anonymous identity protection to achieve high security and privacy [45]. LockChain implements fine-grained permission management by using smart contracts, zero-knowledge proofs, and distributed collaboration, optimizing dynamic policy updates and multi-node verification [46]. However, these solutions store access policies on the blockchain and thus consume significant storage space. Malicious node tolerance mechanisms in distributed systems are commonly implemented through consensus, signature verification, sharding hierarchical networks, and incentives. However, high computational overhead remains a challenge. For instance, the Galaxy framework adopts shard-based BFT consensus with improved leader rotation but relies on the reputation of computing nodes to prevent malicious nodes from becoming leaders [47]. If multiple malicious nodes act simultaneously, system performance may still be affected. Additionally, pre-adapter signatures combined with P2SH scripts can provide offline fault tolerance without relying on smart contracts [48]. However, they involve multiple participants and dynamically generate adapter signatures, complicating key management and storage.

As shown in Table 3, we explore blockchain-based protection approaches for e-commerce, integrating blockchain with IPFS to reduce computational complexity. We leverage stable proof sizes, permission management, and consensus mechanisms to enhance privacy and reliability and minimize single-point dependencies. Smart contracts handle permissions, with off-chain management of access policies for better system performance. Additionally, a decentralized off-chain committee and reputation system ensure node credibility and accountability and reduce the impact of malicious actors.

Table 3. Comparison of blockchain-based privacy protection for e-commerce transactions.

Research	Method	Reference	Advantage	Disadvantage
Blockchain for E-Commerce	Fair Exchange	[37]	no middleman, secure payments, enforce conditions, data safeguard	publicly verified payment amounts
		[38]	sensitive data off-chain, no trust dependencies, identity systems	on-chain storage overhead of commitment values and verification data
		[39]	automated execution, no middleman, flexible economic incentives	limited privacy protection
Privacy Enhancement in Blockchain-based Commerce	Privacy-preserving	[41]	balance between privacy and traceability for compliant blockchain	reliance on specific roles, decentralization reduction
		[42]	strong privacy and flexible auditing	performance closely related to data amount, privacy leaks of auditing risks
		[43]	efficiency and scalability of data trading	fully trusted regulators
	Access Control	[45]	dynamic policies, traceable permissions, complex scenarios	policies stored on-chain consume storage
		[46]	private policies and attributes, fine-grained and dynamic management	policies broadcast across the network
	Malicious Node Tolerance	[47]	sharding-based random allocation for scalability	reliance on node reputation, high computational cost
		[48]	offline tolerance without smart contracts	complex key management with multiple participant signatures

3. Preliminaries

In this section, we introduce the preliminaries of security techniques used in our designed private e-commerce system, mainly including Bulletproof, Schnorr digital signature, and Pedersen's commitment.

3.1. Bulletproof

Bulletproof is a non-interactive zero-knowledge proof protocol designed for efficient range proofs without the need for a trusted setup [49]. The function of the protocol is to verify whether a committed value v lies within a specified range without revealing v. Bulletproof satisfies the properties of zero-knowledge proofs, including completeness, soundness, and zero-knowledge. Formally, given public parameters G, H, l, the commitment is defined as follows:

$$\mathrm{Com}(v) = vG + rH, \quad v \in [0, 2^l - 1], \tag{1}$$

where v, r are private values, $\mathrm{Com}(v)$ is the public value, and l denotes the length of the range. In general, the Bulletproof protocol consists of three main phases:

- Setup: $\mathrm{Setup}(1^\lambda) \rightarrow \mathrm{pp}_R$. This phase generates public parameters pp_R based on security parameter λ.
- Generate range proof: $\mathrm{GenRangeProof}(\mathrm{pp}_R, \mathrm{Com}(v), w) \rightarrow \mathrm{RangeProof}$. This phase generates a range proof RangeProof using public parameter pp_R and private parameter w.
- Verify range proof: $\mathrm{VerRangeProof}(\mathrm{pp}_R, \mathrm{Com}(v), \mathrm{RangeProof}) \rightarrow b$. This phase verifies whether the range proof is valid. If successful, output $b = 1$; otherwise, $b = 0$.

3.2. Schnorr Digital Signature

The Schnorr digital signature scheme is a signature generation and verification process based on the discrete logarithm problem that utilizes elliptic curve cryptography to minimize computation costs [50]. In general, the main steps of the Schnorr scheme are listed as follows:

- Key generation: $\mathrm{KeyGen}(p, g, x) \rightarrow y$. The step first sets a large prime p and a generator g and then selects a private key $x < p$ to calculate the public key $y = g^x \mod p$.
- Signing: $\mathrm{Sign}(m, x) \rightarrow (e, s)$. For any message m, the step chooses a random $k < p$ and computes $r = g^k \mod p$ and hash $e = H(m \parallel r)$. Finally, the signature is derived as $s = (k - xe) \mod (p - 1)$.
- Verification: $\mathrm{Verify}(m, (e, s), y) \rightarrow \{\mathrm{True}, \mathrm{False}\}$. This step computes $v = g^s y^e \mod p$ and then recalculates the hash $e' = H(m \parallel r)$. The signature is derived as True if $e' = e$; otherwise, $s = \mathrm{False}$.

3.3. Pedersen Commitment

The Pedersen commitment is a cryptographic protocol providing a binding and hiding commitment scheme that ensures both the privacy of the committed value and the inability of the committer to change the value once committed. Formally, the Pedersen commitment scheme is defined over a group of prime order q with a generator g and an additional independent generator h. Given a value $v \in \mathbb{Z}_q$ and a random value $r \in \mathbb{Z}_q$, the commitment C is computed as follows:

$$C = g^v h^r \mod p, \tag{2}$$

where v is the value to be committed, r is a random blinding factor, g and h are generators of a cyclic group of order q. The commitment ensures that C does not reveal v.

4. Overview of the System

In order to address security risks in physical e-commerce, we design a blockchain-based privacy-preserving system named PBTMS that ensures data integrity and privacy and reduces reliance on middlemen. The system prevents unauthorized access, tampering, and data loss so as to ensure traceability of e-commerce trading and protect stakeholders' rights. Figure 1 shows the overview of PBTMS, including several core functions such as encryption, privacy preservation, distributed access control, verification, data traceability, and fair trading. These features collaboratively establish a secure, privacy-preserving system for online physical commodity trading that ensures the reliability and trustworthiness of e-commerce platforms.

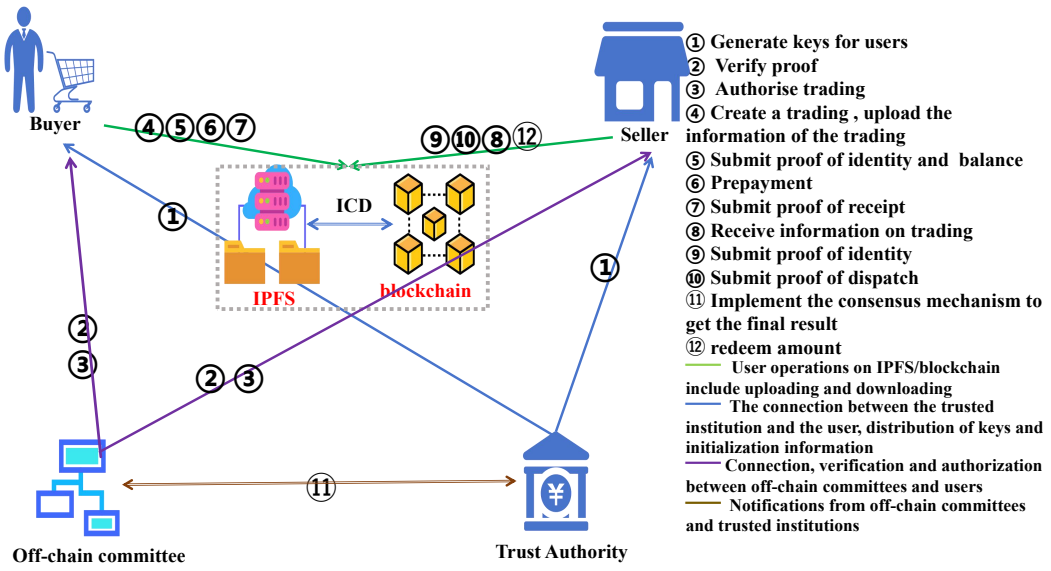


Figure 1. Overview of PBTMS.

4.1. Entities and Responsibilities

As shown in Figure 1, the entities of the system and their responsibilities are listed as follows:

- Buyer: The entity first initiates a trading request and submits its order to a seller. During these operations, the buyer should complete the identity authentication and balance proof to ensure the legitimacy and validity of the trading. After the amount of the trading is paid, the harvest proof should be submitted.
- Seller: Once receiving the trading request from the buyer, the entity should confirm the trading order, complete the authentication, and submit the proof of delivery.
- Trust authority: As a trusted institution, the entity is responsible for generating and publishing the key and authorizing the corresponding trading.
- Off-chain committee: The entity is responsible for two important functions, including (i) verifying identities, transaction validity, and completion status and (ii) authorizing access via smart contracts. It maintains member credibility with a reputation system, using multi-signature authorization and batch processing to improve throughput and ensure decentralized, secure, and efficient operation.
- Blockchain and IPFS: Blockchain stores trading metadata and permission management, while IPFS stores trading details and proofs and guarantees data integrity and privacy.

The five entities in the PBTMS collaborate to manage the entire lifecycle of e-commerce trading, including direct interactions between the buyers and sellers, verification services provided by the miner, coordination and oversight by the trusted authority, and distributed storage by the IPFS and blockchain. As a result, the system ensures trading privacy, security, and traceability and offers reliable technical support for e-commerce.

4.2. Off-Chain Committee Structure

In this paper, the off-chain committee in the PBTMS is primarily responsible for proof verification and authorization execution. To put it simply, the committee can be divided into two layers, i.e., verifying proofs and initiating multi-signature proposals in the first layer and reviewing the results in the second layer.

4.2.1. Operations of the Off-Chain Committee

In order to further ensure the security of the PBTMS, the off-chain committee adopts multiple key measures on the buyer and seller side, which are shown in Figure 2. At first, the buyer pre-pays the amount and uploads the order information without requiring review. After multi-signature approval by the first-layer committee, these operations take effect immediately, ensuring efficient transaction processing. Then, the seller downloads order information without any need for review after approval by the first-layer committee. After the first-layer committee initiates the proposal, it enters a time lock and requires review by the second-layer committee. This ensures that the seller's identity is legitimate. If the seller's identity is in doubt, the system will block the redemption of funds and close the transaction. Finally, if the seller's identity is in question, the incorrect seller, who does not possess the corresponding private key, cannot access the true transaction information. The system will automatically close the transaction and return the buyer's prepayment, avoiding malicious actions and ensuring fairness in the transaction.

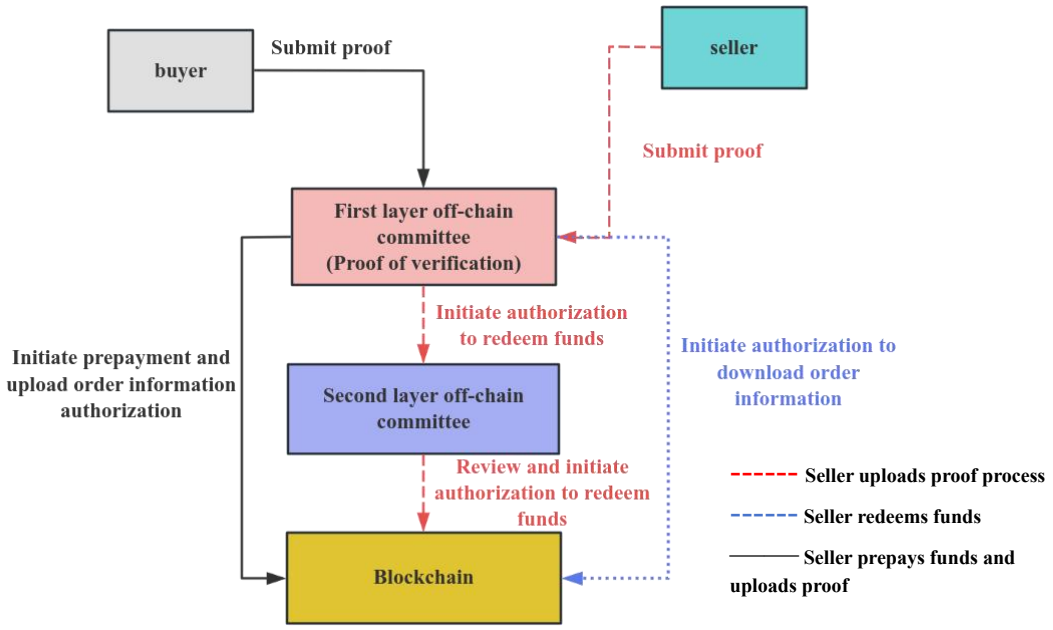


Figure 2. Execution process of the off-chain committee.

4.2.2. Reputation System in the Off-Chain Committee

To ensure the fairness of the results, this paper implements a reputation system that manages the credibility of off-chain committee nodes. The specific responsibilities and rules of the reputation system are listed as follows:

1. Responsibilities: The reputation system prevents malicious nodes from tampering with transactions, submitting invalid signatures, or being inactive for extended periods, which could affect the system's stability. The behavior of each node is measured based on a reputation score (RS), thereby ensuring the security and long-term sustainable operation of the committee.
2. Calculation rules: As shown in Table 4, each committee member's initial reputation score is set to 50 in the rules. The reputation score is dynamically adjusted based on the behavior of the member.
3. Credit score calculation: Formally, the credit score is defined as

$$RS_{t+1} = RS_t + G \cdot \frac{1}{1 + e^{-k(N-N_0)}}, \tag{3}$$

in which RS_t is the reputation score at time t, G is the maximum reputation gain, N is the number of successful participations, N_0 is the threshold for reputation increase, and k is the growth rate constant.

4. Impact of credit score: The credit score is defined as

$$W(RS) = \left(\frac{RS}{100}\right)^2, \tag{4}$$

where $W(RS)$ is the weight of the node, and RS is the reputation score of the node. This formula prevents the system from being significantly affected by nodes with low reputation.

Table 4. Credit calculation rules.

Behavior	Credit Change	Description
Successful Participation	+5	Ensure long-term participation Enhance awareness efficiency
Correct Signature Generation	+5	Ensure security of limited signatures
Long-Term Inactivity	Removed	Penalize unparticipated points Prevent user hijacking
Non-Participation	−20	Ensure participation in key points of the system
Incorrect Judgment on Submission	−10	Prevent mistakes or invalid operations
Incorrect Signature Submission	−30	Prevent invalid operations and identity fraud
Credit Below 50	Removed	Prohibit participation in the system

4.3. Workflow of the System

The PBTMS ensures trading privacy, integrity, and security through encryption technologies, distributed verification, and decentralized management mechanisms. Figure 3 shows the workflow of how the buyer and seller complete the physical commodity trading in our designed privacy-preserving e-commerce. To put it simply, the workflow contains six parts, including trading initiation and key generation, identity verification and balance proof, authorization and data storage, data encryption and trading information retrieval, commodity delivery and status update, and dispute resolution. In the following section, we will introduce every part in detail.

4.3.1. Trading Initiation and Key Generation

The first key part of e-commerce trading is to initialize trading and generate the corresponding key. The detailed operations are listed as follows:

- ECC key generation: When a user creates an account, the trusted authority generates an ECC key based on the user's information and timestamp. The key is sent to the user on the e-commerce platform via a TLS secure channel, and the user's ECC key is periodically updated.
- Shared key generation: During the initiation of the transaction, the trusted authority encrypts predefined information (transaction ID, transaction start timestamp, and both parties' public keys) and sends it to both parties via the TLS secure channel. The

recipient decrypts the data to generate a shared key, ensuring confidentiality, security, and integrity.

- Order creation and fragment encryption: The buyer creates the order, ECC encrypts the prepayment amount and uploads it to the smart contract. The amount address, obfuscated (i.e., encrypted using AES), is placed into the transaction message. The transaction information is then fragmented, and the shared key is used to encrypt the order.

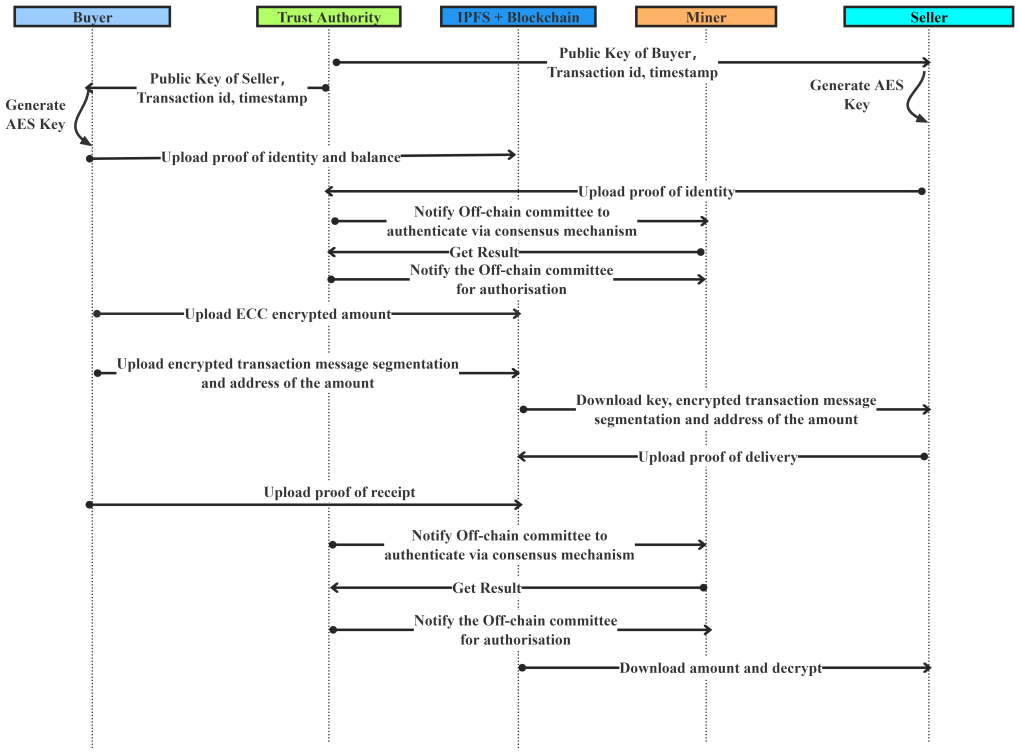


Figure 3. Workflow of PBTMS.

4.3.2. Identity Verification and Balance Proof

Both identity verification and balance proof are very important in the PBTMS and ensure trading security. The corresponding operations include the following:

- Schnorr signature generation: Both the buyer and seller generate a Schnorr signature using their public keys, the timestamp when requesting identity verification, and the transaction ID for authentication.
- Balance proof: The buyers generate Bulletproof for balance verification.
- Proof information contract: Balance proofs and signatures are uploaded to the blockchain.
- Proof information upload: The balance proof and signature are uploaded to the blockchain.
- Off-chain committee validation: The agent notifies the off-chain committee, where miners verify the buyer's Schnorr signature and balance proof. The result is determined via a consensus algorithm and the result is notified to a trust authority.

4.3.3. Authorization and Data Storage

This part contains authorizing the trading and storage of the encrypted data by trusted entities in the PBTMS. They are as follows:

- Authorization proposal: After validating the legitimacy of the buyer's and seller's proofs, the off-chain committee batch initiates proposals. Multiple users can be processed in one transaction.

- Authorization execution: Once the proposal is approved and execution conditions are met, the off-chain committee grant the buyer permission to upload data to the contract.
- Encrypted data upload: The buyer uploads the encrypted data fragments to IPFS and stores the corresponding hash data (ICD) in the transaction information contract on the blockchain.

4.3.4. Data Decryption and Trading Information Retrieval

During trading between the buyer and the seller, sensitive information should be encrypted. Correspondingly, data decryption and trading information retrieval include the following:

- AES key decryption: The seller uses the private key to decrypt the AES key.
- Data fragment retrieval and reconstruction:
  - Upon successful verification, the seller retrieves the encrypted data fragments corresponding to the ICD from the IPFS network.
  - The retrieved encrypted fragments are reassembled, and decryption is performed using the AES key to reconstruct the complete trading dataset, including transaction details and exchanged assets.

4.3.5. Commodity Delivery and Status Update

After the commodity delivery is finished, the system should update the status of all the participants. They are as follows:

- Seller shipment: The seller ships the commodity according to the trading details and updates the status to "success".
- Buyer confirmation of receipt: The buyer receives the commodity and updates the status to "success".
- Commitment generation and upload: The buyer and seller use the trading details, trading creation time, and AES key to generate their respective Pedersen commitments and upload them to the contract.

4.3.6. Dispute Resolution

In the event of a dispute between the buyer and the seller, the third party involved in the dispute (e.g., bank, logistics company, etc.) and the parties to the trade are asked to provide the Pedersen commitment for verification purposes. The outcome of the dispute is decided based on the three commitments.

4.3.7. User Balance Update

Once the transaction status is marked as successful, both the buyer and the seller need to update their user balances. The balance is protected using ECC homomorphic encryption, and the update operation is performed within the encrypted domain without the need for decryption. The buyer's balance decreases while the seller's balance increases. The updated encrypted balance is stored in the blockchain ledger, thereby ensuring transaction security and user privacy.

4.4. Security Risk from Internal and External Adversaries

In PBTMS, there are several common types of privacy and security risks from the user's perspective, mainly including the following:

- Data leakage: The data may be personal or transaction information. That is, personal information leakage refers to attackers stealing the user's registration information (e.g., name, address, and contact details) or transaction records (e.g., buyer and seller

identities, amounts, goods, etc.). Once the user's information is leaked, they may face risks such as identity theft or fraud. Transaction information leakage means that the user's transaction credentials (e.g., transaction password and payment information) may be stolen. Attackers can use this to access the user's trading account, potentially stealing funds or accessing personal transaction data.

- Service interruption: There are two possible cases that cause service interruption. On the one hand, attackers may consume system resources or block communication channels so as to affect platform operation and cause transaction delays or failures, which disrupt the user's normal trading experience. On the other hand, attackers targeting critical system components may cause some transactions or validation processes to fail, affecting transaction integrity and preventing the user's transactions from being processed or validated in a timely manner.
- Loss of trust: Improper actions by the user (e.g., fraudulent transactions or data manipulation) may raise doubts about their credibility among other users, significantly decreasing the user's reputation and affecting their future transaction opportunities.
- Payment and delivery risks: One possible risk is a mismatch between payment and delivery. After payment, the user may not receive the expected goods or services, resulting in financial loss. This may be caused by altered transaction data, forged transaction records, or similar reasons. Another risk may be unmet expected returns. The user may not receive the expected reward for goods or services as agreed, leading to a mismatch between actual returns and expected returns, causing financial losses.

In the PBTMS, we consider two classes of potential adversaries, i.e., internal and external adversaries. As shown in Table 5, these adversaries pose various levels and threats to the system's security.

Table 5. Potential adversaries in PBTMS.

Category	Adversary	Threats	Users Affected
Internal	Malicious Buyer	Forge receipt proofs Leak seller's sensitive information Impersonate other buyers	Decrease user credibility Leak data Payment and delivery risks
	Malicious Seller	Provide false shipping proofs Leak buyer's sensitive information	User credibility decreases, Data leakage Payment and delivery risks
	Malicious Miner	Block permission allocations Forge validation results Disrupt consensus protocols Manipulate verifications	Decreases user credibility Payment and delivery risks
External	Passive Attacks	Analyze data to infer details Conduct traffic analysis	Leak data Payment and delivery risks
	Active Attacks	Intercept and tamper with communication Corrupt proofs or manipulate trade data Impersonate users Launch DoS attacks on critical components	Service interruption Payment and delivery risks

4.4.1. Internal Adversaries

Internal adversaries are legitimate participants in the PBTMS, but they may maliciously exploit their access and permissions. The possible internal adversaries include the following:

- Malicious buyer: A buyer may forge receipt proofs that cause inconsistencies between delivery and receipt proofs or leak the seller's sensitive information that results in losses for the seller. A buyer may also impersonate other buyers to gain unauthorized benefits.

- Malicious seller: A seller may provide false shipping proofs that enable fake deliveries or leak the buyer's sensitive information that causes harm to the buyer.
- Malicious miners: Miners may block permission allocations, forge validation results, or deliberately disrupt the execution of consensus protocols. They may also provide false delivery or receipt verification results so as to manipulate the determination of an e-commerce trade's success or failure.

4.4.2. External Adversaries

External adversaries are attackers outside the system whose aim is to compromise its confidentiality, integrity, and availability through passive observation or active intervention. The typical attack modes include the following:

1. Passive attacks:
    - Accessing public data: Attackers analyze public blockchain data to infer sensitive transaction information (e.g., identities and amounts), compromising user privacy.
    - Traffic analysis and data correlation: Attackers analyze network traffic to infer transaction patterns, which can lead to more precise attacks by identifying transaction times and volumes.
2. Active attacks:
    - Data tampering: Attackers intercept and alter transaction data, potentially compromising correctness and causing financial loss.
    - Impersonating legitimate users: Attackers forge identities to participate in transactions, leading to loss of funds or goods and damaging the system's credibility.
    - DoS attacks: Attackers target critical system components (e.g., off-chain committees and IPFS storage) to disrupt service, leading to downtime or delayed transactions.

Faced with the threat of internal and external adversaries, the PBTMS should take advantage of various robust security mechanisms to defend them well and guarantee the security of trading and the participants' sensitive information.

5. Implementation of Privacy Preservation in PBTMS

In order to protect the e-commerce platforms, the PBTMS must achieve trading data privacy, security, and integrity through encryption algorithms, distributed verification, consensus mechanisms, and permission control. Below is the detailed algorithm design with corresponding expressions.

5.1. Hybrid ECC and AES-Based Chunked Data Encryption for Data Security

For trading data, we propose a hybrid ECC and AES-based chunked data encryption approach to ensure its security, which is shown in Algorithm 1. The idea of the algorithm is to divide a large amount of trading data into multiple chunks and then encrypt them in parallel. The objective of Algorithm 1 is to ensure efficient and secure encryption and derive a symmetric AES encryption key. Formally, the encryption of each chunk is computed as

$$\text{EncryptedChunk}_i = \text{AES}_{\text{Encrypt}}(\text{Chunk}_i, \text{DerivedKey}, \mathbf{H}), \tag{5}$$

in which Chunk_i is the i-th chunk of the input data, $\text{AES}_{\text{Encrypt}}$ is the AES encryption function with the derived symmetric key, DerivedKey is the symmetric encryption key

derived from the shared ECC key using HKDF, and **H** is a hash function calculated from the timestamp and the transaction ID. Then, all the encrypted chunks are aggregated as

$$\text{EncryptedData} = \bigcup_{i=0}^{N} \text{EncryptedChunk}_i, \tag{6}$$

where N is the total number of chunks. It should be noted that the aggregation process is parallelized using a thread pool.

Algorithm 1: ECC-based AES encryption and decryption.

Input: Data: *data*, Chunk size: *chunk_size*, Number of threads: *num_threads*
Output: Encrypted data: *encrypted_data*, Decrypted data: *decrypted_data*

```
/* Step 1: Key Generation */
private_key ← generate_private_key(ec.SECP256R1)
peer_private_key ← generate_private_key(ec.SECP256R1)
peer_public_key ← peer_private_key.public_key()
shared_key ← private_key.exchange(ec.ECDH, peer_public_key)
aes_key ←
  HKDF(SHA256, length=32, info=Hash (transaction ID + timestamp)).derive(shared_key)
/* Step 2: AES Encryption */
iv ← generate_random_iv()
cipher ← AES(aes_key, mode=GCM, iv)
encrypted_data ← []
for i = 0; i < len(data); i = i + chunk_size do
    chunk ← data[i : i + chunk_size]
    encrypted_chunk ← cipher.encrypt(chunk)
    encrypted_data.extend(encrypted_chunk)
/* Step 3: AES Decryption */
cipher ← AES(aes_key, mode=GCM, iv)
decrypted_data ← []
for i = 0; i < len(encrypted_data); i = i + chunk_size do
    chunk ← encrypted_data[i : i + chunk_size]
    decrypted_chunk ← cipher.decrypt(chunk)
    decrypted_data.extend(decrypted_chunk)
return encrypted_data, decrypted_data
```

In addition, the ECC key exchange between any pair of participants in the PBTMS is performed as

$$\text{SharedKey} = \text{PrivateKey.exchange}(\text{ECDH}, \text{PeerPublicKey}), \tag{7}$$

in which PrivateKey represents the ECC private key, and PeerPublicKey is the public key of the peer. Finally, the symmetric key can be derived as

$$\text{DerivedKey} = \text{HKDF}(\text{SHA256}, \text{SharedKey}, \text{info}, \text{length} = 32), \tag{8}$$

where HKDF ensures that the derived key is cryptographically secure for use in AES encryption, and info consists of the transaction ID and timestamp to ensure the uniqueness of the key.

For the proposed AES-GCM and key management, we have demonstrated that it has some key characteristics for data security based on the following theorems.

Theorem 1 (**Confidentiality**). *Under the computational Diffie–Hellman (CDH) assumption and the AES-GCM confidentiality assumption, an attacker cannot recover the plaintext data.*

Proof. The CDH assumption holds that solving the discrete logarithm problem (DLP) in the group is computationally hard, so an attacker cannot recover the shared key. HMAC-based key derivation function (HKDF) is secure and satisfies indistinguishability, meaning the attacker cannot gain any information about the shared key from the derived key. Since AES-GCM mode is IND-CPA secure in the random oracle model, even if an attacker obtains multiple ciphertext blocks, they cannot recover the plaintext data. □

Theorem 2 (**Replay Attack Resistance**). *If the data block includes a timestamp, the attacker cannot replay the same encrypted data.*

Proof. In the AES-GCM encryption process, the nonce must be unique. Otherwise, it breaks the encryption's security. Since the nonce is derived from the hash of the timestamp and sequence number, it will always be unique, preventing an attacker from replaying old valid transactions within the system. □

Theorem 3 (**CCA Resistance**). *Assuming AES-GCM is IND-CCA2-secure, the attacker cannot forge a valid ciphertext.*

Proof. In AES-GCM mode, each ciphertext includes an integrity check value (authentication tag). If the ciphertext is modified, the decryption will fail the verification, and the receiving party will reject the message. Therefore, an attacker cannot forge a ciphertext that passes verification. □

Theorem 4 (**Key Leak Prevention**). *Under the key management mechanism combining ECDH, HKDF, and AES-GCM, even if some information is leaked, the attacker cannot recover the complete key.*

Proof. Each ECDH exchange generates a new session key, and ECDH only exposes the public key. The CDH assumption ensures that an attacker cannot derive the shared secret from the public keys. HKDF provides irreversibility, ensuring that the shared key and the final symmetric key are not directly recoverable from each other. The keys generated by ECDH are uniformly distributed, and HKDF further obscures the key, so even partial leakage does not allow the attacker to recover the full key. □

5.2. ECC-Based Signature for Data Integrity Verification

Data integrity verification consists of two operations: signature computation and integrity verification. At first, the signature with respect to some data is computed as

$$\text{Signature} = \text{ECC}_{\text{Private}}(\text{Hash}(D)), \tag{9}$$

where Signature is the ECC signature for data D, $\text{Hash}(D)$ is the hash value of the data D, and $\text{ECC}_{\text{Private}}$ is the ECC private key used for signing. After receiving the signature, the receiver can verify the integrity of data by the following equation:

$$\text{Hash}(D) \stackrel{?}{=} \text{ECC}_{\text{Public}}(\text{Signature}), \tag{10}$$

in which $\text{ECC}_{\text{Public}}$ is the ECC public key used for verification.

5.3. Trading Verification and Consensus

Trading verification and consensus in PBTMS include three important components: identity authentication, balance proof verification for funds guarantee, and delivery and receipt proof. Algorithm 2 shows the complete process of proof generation and verification for trading verification and consensus. In the following, we will introduce every component in detail.

Algorithm 2: Proof generation and verification using Schnorr and Bulletproof.

Input: Order ID: O, Buyer's private key: K_B^{priv}, Trading amount: A, Buyer's balance: B, Number of miners: n

Output: Final verification result: R_{final}

`/* Step 1: Proof Generation */`

$O_{\text{bytes}} \leftarrow \text{bytes}(O)$, $H_O \leftarrow \text{SHA256}(O_{\text{bytes}})$

`/* Generate Schnorr proof of identity */`

$r \leftarrow \text{Random}(1, p-1)$, $R \leftarrow r \cdot G$, $c \leftarrow \text{SHA256}(H_O||R)$, $s \leftarrow r + c \cdot K_B^{priv} \mod p$, $P_B \leftarrow (R, s)$

`/* Generate Bulletproof of balance */`

$R_{\text{range}} = \{B - A \geq 0\}$, $C_B \leftarrow B \cdot G + r \cdot H$, $P_{BP} \leftarrow \text{Prove}(R_{\text{range}}, C_B, K_B^{priv})$

`/* Step 2: Proof Verification by Miners */`

$r \leftarrow [0, 0, \ldots, 0]$

for $i = 0; i < n; i++$ **do**

  $c \leftarrow \text{SHA256}(H_O||R)$

  $R' \leftarrow s \cdot G - c \cdot K_B^{pub}$

  if $R' = R$ **then**

    $V_S = \text{true}$

  else

    $V_S = \text{false}$

  $V_{BP} \leftarrow \text{Verify}(P_{BP}, C_B)$

  if $V_S = true$ ***and*** $V_{BP} = true$ **then**

    $r[i] \leftarrow 1$

  else

    $r[i] \leftarrow 0$

`/* Step 3: Aggregate Miner Results */`

$S \leftarrow 0$

for $i = 0; i < n; i++$ **do**

  $S \leftarrow S + r[i]$

if $S > \frac{2n}{3}$ **then**

  $R_{\text{final}} \leftarrow 1$

else

  $R_{\text{final}} \leftarrow 0$

return R_{final}

5.3.1. Schnorr Signature Verification

The transaction summary is generated using the transaction ID and the sender's public key along with the following timestamp:

$$m = H(\text{Transaction ID} \parallel \text{Timestamp} \parallel \text{Sender's public key}). \tag{11}$$

Using the Schnorr signature scheme to verify the identity of the buyer and the seller, the expressions are as follows:

$$R = g^k \mod p, \tag{12}$$

$$s = k + x \cdot H(m, R, y) \mod q, \tag{13}$$

$$\sigma = (R, s), \tag{14}$$

where p and q are parameters for the signature, R is the random value, k is the random number generated by the secure random number generator (CSPRNG) to prevent leakage, x is the private key, m is the transaction summary, and H is the hash function.

The corresponding signature verification method is as follows:

$$g^s = R \cdot y^{H(m,R)} \mod p, \tag{15}$$

where y is the public key.

For the Schnorr signature, we have demonstrated that it has several key security characteristics based on the following theorems.

Theorem 5 (**Replay Attack Prevention**). *Suppose the hash function H is modeled as a random oracle (ROM). If the attacker tries to replay an old transaction, it will be rejected by the system.*

Proof. Assume the attacker intercepts a valid transaction (m, R, s) and attempts to replay it. Since the transaction's m is unique for every transaction, under the ROM assumption, the output $H(m, R, y)$ is unpredictable, and thus the attacker cannot replay and perform a new transaction. If the attacker tries to reuse m, it becomes invalid. Hence, the attacker cannot replay an old transaction, which prevents a valid proof. □

Theorem 6 (**Collision Resistance**). *If the attacker can observe two signatures using the same random number k, the attacker can compute the secret x.*

Proof. Suppose there are two different messages m_1, m_2 but the random number k is the same.

$$s_1 = k + c_1 \cdot x \mod q, \tag{16}$$

$$s_2 = k + c_2 \cdot x \mod q, \tag{17}$$

$$s_1 - s_2 = (c_1 - c_2) \cdot x \mod q. \tag{18}$$

Since the random number k is chosen correctly, the attacker cannot deduce x. □

Theorem 7 (**Anti-CCA**). *In the case of the discrete logarithm assumption (DLP) and ROM model, the attacker cannot forge a valid Schnorr signature.*

Proof. Assume the attacker can select a ciphertext and obtain its signature. Since the signature is dependent on $H(m, R, y)$, the attacker cannot generate a valid signature through a forged ciphertext. □

Theorem 8 (**Public Key Substitution Attack Prevention**). *If the attacker substitutes a public key y with a different one, the attacker cannot forge a valid signature.*

Proof. Since H is collision-resistant, $H(m, R, y) \neq H(m, R, y')$. Therefore, the attacker cannot forge a valid signature using a different public key. □

5.3.2. Balance Proof Verification

The transaction has a unique identifier computed as follows:

$$\mathrm{ID}_{\mathrm{tx}} = H(\text{Transaction ID} \parallel \text{Timestamp}), \tag{19}$$

where the *Transaction ID* is the unique identifier of the transaction, such as the order number or a hash value from a blockchain transaction, and the *Timestamp* is the time of the transaction, which helps prevent replay attacks.

The objective of balance proof verification is to check whether or not the buyer's balance is sufficient for trading. It consists of two operations, i.e., balance proof computation and verification. At first, the balance proof is derived by

$$\mathrm{Proof}_{\mathrm{Balance}} = \mathrm{Bulletproof}(\mathrm{Amount}, \mathrm{Balance}), \tag{20}$$

where $\mathrm{Proof}_{\mathrm{Balance}}$ is the zero-knowledge proof generated by Bulletproof, Amount is the trading amount, and Balance is the buyer's current balance. Then, it verifies the balance proof by the following equation:

$$\mathrm{Verify}(\mathrm{Proof}_{\mathrm{Balance}}) \stackrel{?}{=} \mathrm{True}, \tag{21}$$

where Verify is the verification algorithm of Bulletproof.

For the balance proof, we have demonstrated that it satisfies some unique security characteristics based on the following theorems.

Theorem 9 (**Replay Attack Prevention**). *The attacker intercepts the transaction and attempts to replay it but fails.*

Proof. Each transaction proof includes a unique identifier formed by the transaction ID and the timestamp, ensuring that the proof for any transaction is unique. □

Theorem 10 (**Forgery Prevention**). *Under the random oracle model (ROM) and the discrete logarithm assumption (DLP), the attacker cannot forge a valid proof.*

Proof. The core idea of Bulletproofs is based on the Pedersen commitment, and since Pedersen commitments are homomorphic, the attacker cannot choose a valid message to forge a valid commitment to either the balance or the amount in the transaction. □

Theorem 11 (**Information Leakage Attack**). *Under the random oracle model (ROM) and the Pedersen commitment's hiding property, the attacker cannot infer any sensitive information from multiple proofs in the same transaction.*

Proof. The Bulletproof proof either returns `True/False`, and does not reveal any values inside the commitment or the commitment's randomness. Pedersen commitments maintain non-interactivity, meaning that the same proof value can be reused without leakage. □

Theorem 12 (**Prevention of Proof Modification**). *Under the constraints of the Pedersen commitment's binding property and the Bulletproof proof system's integrity, the attacker cannot modify the proof.*

Proof. The Bulletproof system uses a hash value $H(m, R, y)$ to bind the commitment to the transaction's amount. The commitment is binding and cannot be modified without

invalidating the proof. Once the commitment is generated, it cannot be changed without causing the proof to fail. □

5.3.3. Delivery and Receipt Proof

The objective of delivery and receipt proof is to verify consistency between delivery and receipt proofs. In particular, it uses the Pedersen commitment to implement the above objective, which can be divided into three operations. The first operation is that the buyer generates a random number r by encrypting the order timestamp with the AES key:

$$r = \mathrm{AES}_K(\mathrm{Timestamp}). \tag{22}$$

The second operation is that both the buyer and seller generate Pedersen commitments based on the following equation:

$$C = g^{x_{\mathrm{Order}}} \cdot h^r, \tag{23}$$

where x_{Order} is the order information, r is the random number, and g, h are public parameters. The last operation is to verify the commitment, which is expressed as

$$C_{\mathrm{Delivery}} \stackrel{?}{=} C_{\mathrm{Receipt}}. \tag{24}$$

For delivery and receipt proof, we have demonstrated that it satisfies the following security characteristics.

Theorem 13 (**Replay Attack Prevention**). *The attacker cannot replay an old transaction to create a fraud.*

Proof. Due to the randomness of the time-stamp value r generated by AES, which ensures that each instance is different, the attacker fails if they try to replay an old transaction with the same timestamp. □

Theorem 14 (**Forgery Prevention**). *Under the assumption of the binding property of the Pedersen commitment, the attacker cannot modify the transaction information.*

Proof. Due to the difficulty of the discrete logarithm problem (DLP), the attacker cannot determine a new x'_{order} that satisfies $C' = C$. □

Theorem 15 (**Information Leakage Attack Prevention**). *Under the hiding property of the Pedersen commitment, the attacker cannot infer the transaction information.*

Proof.

$$\log(C) = x_{\mathrm{order}} + \log h^r, \tag{25}$$

which ensures that if r is chosen randomly, the x_{order} is non-trivial and cannot be deduced. If the two different transactions have distinct commitments, under the DLP assumption, the attacker cannot decide if the two transactions are the same. □

5.4. Permission Management

Permission management is composed of two important tasks, i.e., permission allocation and authorization. For permission allocation, the main objective is to control access to sensitive data, which is expressed as

$$\text{Permission}(U, A) = \begin{cases} 1 & \text{if Verified}(U) \\ 0 & \text{otherwise} \end{cases}, \tag{26}$$

where U is some user, and A is the user's action (e.g., upload and download).

In addition, the objective of permission authorization is to achieve consensus among distributed miners for authorization and verification tasks. The final result is determined based on the agreement of more than two-thirds of the participating miners. To this end, we propose a consensus mechanism for distributed authorization and verification, which is composed of four key steps. The first step is for the miner to compute its local verification result for individual miner verification. The result is derived as

$$V_i = \begin{cases} 1 & \text{if verification is successful} \\ 0 & \text{if verification fails} \end{cases}, \tag{27}$$

where V_i is the result of miner i. The second step is that the miners broadcast the result. Correspondingly, the set of received results is represented as follows:

$$\mathcal{V} = \{V_1, V_2, ..., V_n\}, \tag{28}$$

where n is the total number of miners participating in the consensus. The third step is majority voting that determines the final consensus result. The voting result is expressed as

$$\text{Result}_{\text{final}} = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} V_i * W_i > 2n/3 \\ 0 & \text{otherwise} \end{cases}, \tag{29}$$

where $2n/3$ is the threshold for consensus. Finally, the final decision is made. That is, if $\text{Result}_{\text{final}}$ is 1, the authorization or verification task is considered successful, and the trust authority proceeds with the next step (e.g., granting permissions or updating the blockchain). Otherwise, the task fails, and corrective actions are triggered. The above process of permission management is shown in Algorithm 3.

5.5. Dispute Resolution

Dispute resolution between trading parties by verifying Pedersen commitments is provided by the buyer, seller, and relevant third parties (e.g., bank and logistics) for dispute-related data, such as tracking numbers or trading IDs. The concrete process is divided into two parts. The first part is a cross-party data comparison that compares the verified data from all parties by the following equation:

$$C_{\text{Buyer}} \stackrel{?}{=} C_{\text{Seller}} \stackrel{?}{=} C_{\text{ThirdParty}}. \tag{30}$$

If the data values match, the dispute is resolved in favor of consistency. Otherwise, further investigation is required. The second part is the final judgment based on the results of commitment verification and data comparison. If inconsistencies are found, the PBTMS should identify the responsible party. For example, if C_{Buyer} matches $C_{\text{ThirdParty}}$ but differs from C_{Seller}, the seller may be at fault. More importantly, the PBTMS should resolve the dispute by either rejecting or validating the trading and recording the decision on the blockchain.

Algorithm 3: Proof generation and verification for shipping and trading.

Input: Logistics info of buyer: L_B, Seller: L_S, Trust authority: L_T, Trading key: K_T, Delivery time: T, Elliptic curve base points: G, H, Number of miners: n

Output: Final trading completion status: S

`/* Step 1: Generate Pedersen Commitments */`

$r_B, r_S \leftarrow \text{RandomScalar}()$

$P_{\text{ship}} \leftarrow H(K_T||T||L_B) \cdot G + r_B \cdot H$, $P_{\text{receive}} \leftarrow H(K_T||T||L_S) \cdot G + r_S \cdot H$

`/* Step 2: Verify Proofs by Miners */`

$V \leftarrow [0, 0, \ldots, 0]$

for $i = 0; i < n; i++$ **do**

  $v_1 \leftarrow P_{\text{ship}} \stackrel{?}{=} P_{\text{receive}}$

  if $v_1 = true$ **then**

    $V[i] \leftarrow 1$

  else

    $V[i] \leftarrow 0$

$R \leftarrow 0$

for $i = 0; i < n; i++$ **do**

  $R \leftarrow R + V[i]$

if $R > \frac{2n}{3}$ **then**

  $S \leftarrow 1$

else

  $S \leftarrow 0$

if $S = 1$ **then**

  return S

`/* Step 3: Dispute Resolution */`

$r_T \leftarrow \text{RandomScalar}()$

$P_{\text{ThirdParties}} \leftarrow H(K_T||T||L_T) \cdot G + r_T \cdot H$

for $i = 0; i < n; i++$ **do**

  $v_1 \leftarrow P_{\text{ThirdParties}} \stackrel{?}{=} P_{\text{receive}}$, $v_2 \leftarrow P_{\text{ThirdParties}} \stackrel{?}{=} P_{\text{ship}}$

  if $v_1 = true$ ***and*** $v_2 = true$ **then**

    $V[i] \leftarrow 1$

  else

    $V[i] \leftarrow 0$

$R \leftarrow 0$ **for** $i = 0; i < n; i++$ **do**

  $R \leftarrow R + V[i]$

if $R > \frac{2n}{3}$ **then**

  $S \leftarrow 1$ `// Final result: Approved`

else

  $S \leftarrow 0$ `// Final result: Rejected`

return S

5.6. Smart Contract for Access Control and Data Storage

In the PBTMS, we use smart contracts to implement two contracts, i.e., an access control contract for permission management and a data storage contract for storage management. As shown in Figures 4 and 5, the smart contract is mainly used to manage the upload and download permissions and to perform data upload and download.

```
// Executive authorisation
function executeProposal(bytes32 proposalId) internal {
Proposal storage proposal = proposals[proposalId];
require(proposal.approvals >= requiredApprovals, "Not enough approvals");
require(!proposal.executed, "Proposal already executed");
if (keccak256(bytes(proposal.action)) == keccak256("grantUpload")) {
uploadPermissions[proposal.user] = true;
emit UploadPermissionGranted(proposal.user);
} else if (keccak256(bytes(proposal.action)) == keccak256("revokeUpload")) {
uploadPermissions[proposal.user] = false;
emit UploadPermissionRevoked(proposal.user);
} else if (keccak256(bytes(proposal.action)) == keccak256("grantDownload")) {
downloadPermissions[proposal.user] = true;
emit DownloadPermissionGranted(proposal.user);
} else if (keccak256(bytes(proposal.action)) == keccak256("revokeDownload")) {
downloadPermissions[proposal.user] = false;
emit DownloadPermissionRevoked(proposal.user);
}
proposal.executed = true;
emit ProposalExecuted(proposalId);
}
```

Figure 4. Access control contract.

```
// Executive authorisation
function executeProposal(bytes32 proposalId) internal {
Proposal storage proposal = proposals[proposalId];
require(proposal.approvals >= requiredApprovals, "Not enough approvals");
require(!proposal.executed, "Proposal already executed");
if (keccak256(bytes(proposal.action)) == keccak256("grantUpload")) {
uploadPermissions[proposal.user] = true;
emit UploadPermissionGranted(proposal.user);
} else if (keccak256(bytes(proposal.action)) == keccak256("revokeUpload")) {
uploadPermissions[proposal.user] = false;
emit UploadPermissionRevoked(proposal.user);
} else if (keccak256(bytes(proposal.action)) == keccak256("grantDownload")) {
downloadPermissions[proposal.user] = true;
emit DownloadPermissionGranted(proposal.user);
} else if (keccak256(bytes(proposal.action)) == keccak256("revokeDownload")) {
downloadPermissions[proposal.user] = false;
emit DownloadPermissionRevoked(proposal.user);
}
proposal.executed = true;
emit ProposalExecuted(proposalId);
}
```

Figure 5. Data storage contract.

5.6.1. Access Control Contract

The objective of an access control contract is for the access control component to dynamically manage permissions for different users during the trading process. The contract includes two main operations, i.e., grant and revocation. The grant permission refers to when a user U is granted permission R, i.e., $P_R \leftarrow P_R \cup \{U\}$, and any pending requests from U for permission R are re-evaluated. On the other hand, the revocation permission means that when a user U has their permission R revoked, i.e., $P_R \leftarrow P_R \setminus \{U\}$, any ongoing tradings requiring permission R for U are immediately terminated. Based on the contract, access control uses dynamic permission sets to manage access, which are defined as

$$P_{\text{upload}} = \{U_i \mid U_i \text{ is authorized to upload data}\}; \tag{31}$$

$$P_{\text{download}} = \{U_j \mid U_j \text{ is authorized to download data}\}. \tag{32}$$

As a result, the corresponding access decisions are based on the following rule:

$$\text{Access}(U, R) = \begin{cases} 1, & \text{if } U \in P_R \\ 0, & \text{otherwise} \end{cases}. \tag{33}$$

5.6.2. Data Storage Contract

The data storage contract is composed of two core functions: upload and download, which manage the storage and retrieval of trading data. Both functions ensure that only authorized users can perform the respective operations. The upload function allows authorized users (e.g., buyers) to store trading-related data references on the blockchain. It first checks whether the user is authorized to upload data based on the following rule:

$$\text{Access}(UserID, P_{\text{upload}}) = \begin{cases} 1, & \text{if } UserID \in P_{\text{upload}} \\ 0, & \text{otherwise} \end{cases}. \tag{34}$$

If a user is authorized, data are stored on the blockchain. The upload operation is performed by

$$\text{Upload}(UserID, Data) \rightarrow \text{Status}, \tag{35}$$

where UserID is an identifier of the requesting user, Data is the trading-related data reference (e.g., ICD), P_{upload} is a set of users authorized to upload data, and Status is confirmation of the data storage and updated trading state.

On the other hand, the download function allows authorized users (e.g., sellers) to retrieve stored data references from the blockchain. It first checks whether the user is authorized to download data based on the following equation:

$$\text{Access}(UserID, P_{\text{download}}) = \begin{cases} 1, & \text{if } UserID \in P_{\text{download}} \\ 0, & \text{otherwise} \end{cases}. \tag{36}$$

If a user is authorized, he/she retrieves the data reference from the blockchain based on the following rule:

$$\text{Download}(UserID, OrderID) \rightarrow \text{Data}, \tag{37}$$

where P_{download} is a set of users authorized to download data, and Data is the requested data reference (e.g., ICD).

5.6.3. Security Theorems for Smart Contracts

Theorem 16 (**Re-entrancy Attack Prevention**). *The attacker cannot gain unauthorized funds or permissions through a re-entrancy attack.*

Proof. The contract uses the `ReentrancyGuard` library and `nonReentrant` modifier to prevent re-entrancy attacks. Through these mechanisms, the contract ensures that each operation involving fund transfers or permission modifications can only be executed once. Even if an external call is initiated during the contract's execution, the contract's state cannot be maliciously modified by reentry, thereby avoiding the loss of funds or abuse of permissions due to re-entrancy attacks. □

Theorem 17 (**Front-running Attack Prevention**). *The attacker cannot interfere with or tamper with a proposal before the transaction is executed by using a front-running strategy.*

Proof. To prevent front-running attacks, the contract implements a time-lock mechanism and a review process. The time-lock ensures that all approved proposals cannot be executed immediately, preventing malicious nodes from observing the proposal and submitting competing transactions. Additionally, the review mechanism provides a second-layer confirmation for all proposals. Even if a malicious node attempts to interfere, the second-layer committee's review can effectively block the occurrence of front-running. □

Theorem 18 (**Integer Overflow and Underflow Prevention**). *Mathematical operations within the contract do not cause integer overflow or underflow.*

Proof. All mathematical calculations in the contract use the `SafeMath` library, which prevents integer overflow or underflow issues. Even when handling large amounts of funds or transactions, `SafeMath` automatically checks for overflow or underflow, as well as subtraction, multiplication, and division operations. If any issues are detected, an exception (revert) is triggered, ensuring that the contract's execution will not result in the loss of funds due to overflow errors. □

5.6.4. Smart Contract Auditing and Verification

The auditing and verification process of the smart contract includes the following key steps:

1. Static code auditing: The contract's source code is analyzed by using tools like `Slither` and `MythX` to automatically scan the code for potential vulnerabilities.
2. Functional verification: Testing frameworks (such as `Truffle`) are used for unit testing and integration testing, thereby ensuring that the contract's functionality meets the expected behavior.
3. Security testing: Common attacks, such as re-entrancy attacks, front-running attacks, integer overflow, etc., are simulated to check the security of the contract and ensure that it can withstand these types of attacks.
4. Monitoring and updates: After the contract is deployed, continuous monitoring is performed to check for any new security vulnerabilities or abnormal behavior. An automatic alert system will notify developers or administrators of any issues. If potential vulnerabilities or issues are discovered, the system will propose fixes or prompt for an urgent update to the contract.

In particular, we use OpenZeppelin Defender for real-time monitoring. That is, OpenZeppelin Defender provides real-time monitoring of contract operations, detecting potential malicious activities, reentrancy attacks, or other misuse of the contract. Through real-time

transaction monitoring, contract event listening, and automatic alerts, issues can be detected and addressed promptly. Defender's automation features also ensure that critical operations in the contract are executed as scheduled and can automatically update the contract or pause operations with security risks.

5.7. PBTMS: Put Them in Together

Based on the above operations (e.g., data encryption and signature, trading verification and consensus, permission management, and dispute resolution), the flow chart of PBTMS is shown in Figure 6. It should be noted that the yellow section is for encrypted trading information and amounts, the pink section is for proof of identity and balance, the blue is for proof of receipt and upload, and the purple is for dispute view and amount redemption. The PBTMS achieves robust data privacy, distributed verification, and permission control, ensuring secure, reliable, and private trading.

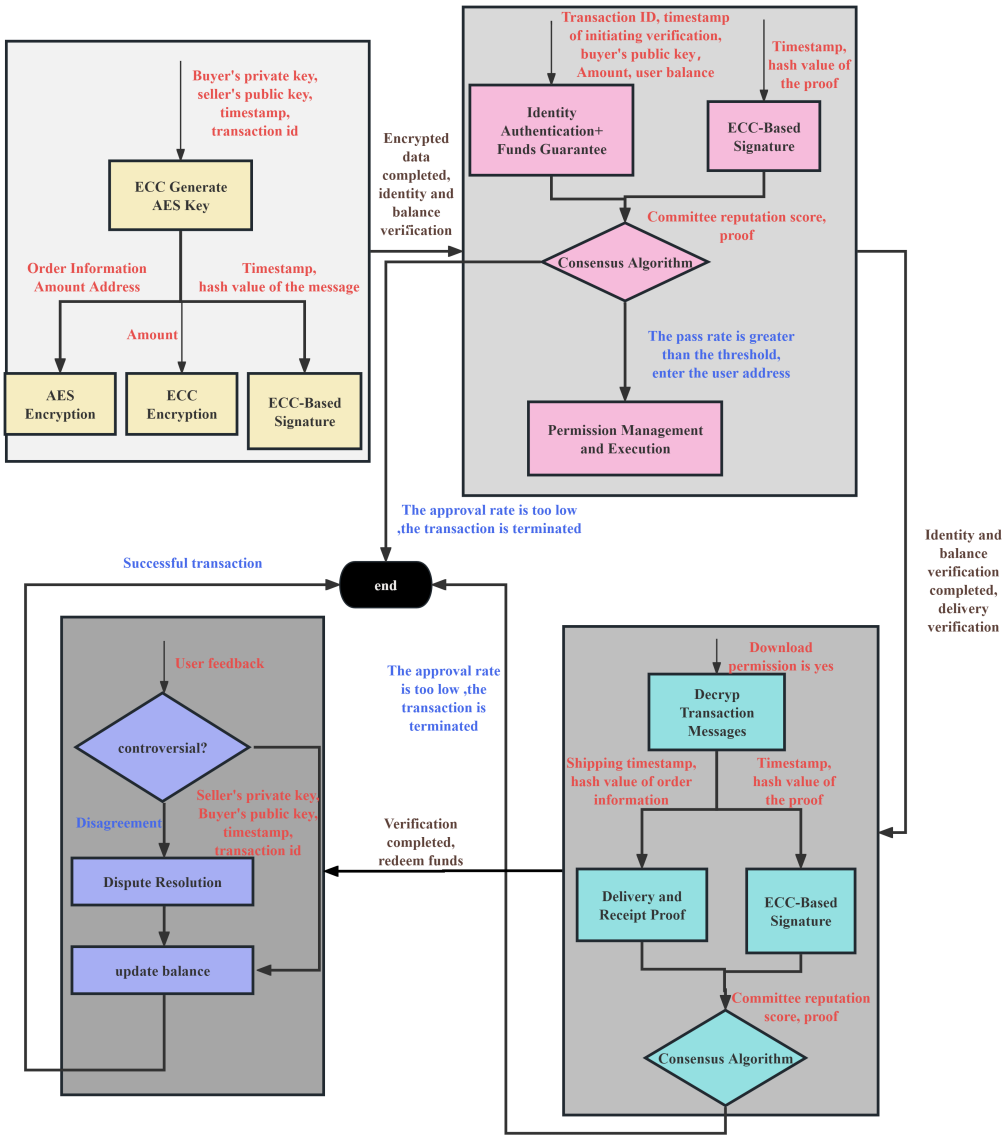


Figure 6. Flow chart of PBTMS.

6. Security and Performance Analysis of PBTMS

In this section, we analyze the security and performance of PBTMS in terms of multiple metrics, such as security and privacy, authentication and accountability, fairness, availability, throughput, response time, gas consumption, proof size, and scalability.

6.1. Security Analysis of PBTMS

The PBTMS addresses security issues in e-commerce trading, as shown in Table 6, including security and privacy, authentication and accountability, fairness, and reliability. The focus of these security issues is listed as follows:

- Security and privacy: It refers to protecting sensitive trading data from unauthorized access and ensuring it cannot be tampered with during transmission or storage.
- Authentication and accountability: It refers to verifying the legitimacy of trading participants to prevent them from denying their actions in the trading process.
- Fairness: It refers to ensuring that all participants are treated equally to avoid losses due to fraud or unfairness.
- Reliability: It refers to maintaining system functionality even under adversarial conditions.

This section analyzes potential attack scenarios based on the defined internal and external adversary models and discusses the system's defense mechanisms and security guarantees.

Table 6. Potential attack on the PBTMS and the corresponding defense.

Metric	Potential Attack	Defense Mechanism
Security and Privacy	Passive/Active Attacks, Data Leakage, Forgery	AES Key Encryption, Obfuscated Addresses, ECC Signature, Blockchain Storage
Authentication	Identity Forgery, MITM Attacks	Schnorr Signatures, Distributed Validation
Fairness	False Delivery/Receipt, Miner Bias	Commitment Verification, Consensus Mechanisms
System Availability	DoS Attacks, Miner Disruption	IPFS Storage, Robust Consensus, Collaborative Management

6.1.1. Security and Privacy

The potential security and privacy attacks on the PBTMS are listed as follows:

- External passive attack: Adversaries monitor blockchain or network traffic to extract sensitive trading information, such as trading amounts, order details, or user identities.
- Internal data leakage: Malicious miners exploit privileges to access fragmented data, leaking sensitive trading details.
- External active attack: Adversaries intercept and alter trading data or results during transmission.
- Internal forgery: Malicious participants or miners forge delivery or receipt proofs, disrupting the trading process.

Faced with these attacks, the PBTMS can adopt the following security measures:

- Shared AES key, slice, and encryption of data: Data are fragmented and encrypted using a shared AES key, preventing reconstruction from intercepted fragments.

- Obfuscated address: The system obfuscates addresses of contract to protect amount. As a result, it ensures that any tampering with data invalidates the signature.
- On-chain storage: Blockchain's distributed consensus and tamper-proof structure prevent unauthorized modifications and ensure data consistency.

6.1.2. Authentication and Accountability

The potential authentication and accountability attacks on the PBTMS are listed as follows:

- Identity forgery: Adversaries impersonate legitimate users to gain unauthorized benefits.
- Man-in-the-middle (MITM) attacks: Adversaries intercept and modify identity verification data.

Correspondingly, the defense mechanisms of the PBTMS include the following:

- Schnorr signature: The system authenticates trading participants so that only legitimate users can sign and engage in trading.
- Distributed validation by miners: Independent miners validate signatures to reduce the risks of tampering or forgery.

6.1.3. Fairness

The potential fairness attacks on the PBTMS include the following:

- False delivery: Malicious sellers forge delivery proofs to claim payments without delivering commodity.
- False receipt: Malicious buyers forge receipt proofs to avoid payment or request refunds.
- Miner bias: Malicious miner nodes manipulate validation results, causing unfair trading outcomes.

In order to ensure the fairness of the PBTMS, the proper defense mechanisms are listed as follows:

- Commitment-based proof verification: Buyers and sellers generate Pedersen commitments for order data. Miners verify these commitments for authenticity.
- Distributed consensus: Miners use consensus mechanisms to prevent manipulation and ensure fairness.

6.1.4. System Availability

The potential attacks on the PBTMS's availability may be as follows:

- Denial-of-service (DoS) attacks: External adversaries launch DoS attacks on validation miners or IPFS, attempting to disrupt trading verification or data storage services.
- Malicious miner disruption: Malicious miners refuse to allocate permissions or participate in consensus, attempting to interrupt the trading process.

In order to guarantee the availability of the PBTMS, the corresponding protection mechanisms include the following:

- Distributed storage in IPFS: Data and proofs are fragmented and stored, with the result of avoiding single points of failure.
- Robust consensus mechanism: The system can ensure the functionality even if up to 1/3 of miners are malicious or fail.
- Collaborative node management: Miners work together to manage permissions and validations, avoiding interruptions.

As shown in Table 7, some studies [51,52] employed an escrow protocol based on a Bitcoin script that uses threshold signatures combined with zero-knowledge proofs, as well as multi-signatures and verifiable cryptographic signatures, to ensure that tradings are fair. However, none address measures to protect the anonymity of the identities of the parties to the trading and the confidentiality of the trading amount. In contrast, Zheng et al. [53]

implemented a trustless trading process using smart contracts, encrypted addresses, and a bidding mechanism. However, verifying the user's balance needs to be recognized, and it is impossible to determine whether the user has enough balance to purchase the commodity. Lee et al. [54] paid attention to verifying identity and balance based on privacy protection but did not consider the consistency of receiving and shipping information, which cannot guarantee the trading's fairness. Our design is more comprehensive in security, enhancing trading safety, privacy, and fairness while addressing the shortcomings of existing schemes.

Table 7. Comparison of safety properties.

References	Security & Privacy	Fairness	Authentication & Accountability	Availability
[51]	✗	✓	✓	✓
[52]	✗	✓	✓	✓
[53]	✓	✓	✗	✓
[54]	✓	✗	✓	✓
PBTMS	✓	✓	✓	✓

Note: ✓ indicates that the property is satisfied, while ✗ means it is not satisfied. The references correspond to the following works: ref. [51]—Escrow Protocols for Cryptocurrencies: How to Buy Physical Goods Using Bitcoin, ref. [52]—A survey on privacy protection of blockchain: The technology and application, ref. [53]—SPENDER: A platform for secure and privacy-preserving decentralized P2P e-commerce, ref. [54]—Design and implementation of Web3-Based e-commerce cryptocurrency payment system.

6.1.5. Defense Analysis of Sybil Attacks

A Sybil attack refers to an attack where the attacker creates a large number of fake nodes to influence the consensus results, thus manipulating transaction validation or permission management [55,56]. For off-chain committees, a Sybil attack may result in vote manipulation, threshold signature manipulation, and even the breakdown of the permission management system [57].

Here, we consider four common Sybil attack approaches and present the corresponding defense measures. At first, Sybil nodes manipulate voting, which means that the attacker creates a large number of new nodes (initial reputation 100) and uses their voting power to influence the consensus, forcing the system to accept invalid transactions or incorrect identity verification. Faced with the attack, there are three proper measures, including that (i) newly joined nodes have an initial reputation of 50 (instead of 100) to prevent sudden influence on consensus, (ii) reputation must be improved through long-term correct behavior, and (iii) voting rights based on the reputation score adopt exponential decay, with low-reputation nodes having a lower voting weight.

Then, the attacker temporarily boosts reputation to manipulate consensus. The attacker controls a batch of Sybil accounts, performs legitimate consensus tasks in the short term to boost their reputation, and then forges transactions or submits incorrect signatures to influence consensus decisions. The corresponding defense measures can be severe penalties for incorrect behavior and Removal of inactive nodes in the long term to prevent attackers from temporarily boosting their reputation and then stagnating to await the opportunity for attack.

Next, the attacker attempts to manipulate multi-party proposals. The attacker uses Sybil accounts to gain proposal rights, forge signatures, or block legitimate transactions. There are two possible defense measures. The first one is that high-reputation nodes are prioritized to enter the proposal group. Only nodes with a reputation score (RS) ≥ 80 are allowed to join the proposal group. Nodes with RS < 50 are immediately removed from the proposal group. Another one is that the proposal weight is related to the reputation score.

Lastly, the attacker creates fake identities at a low cost. In the off-chain committee, attackers may try to create fake identities to increase their influence by maliciously controlling multiple committee nodes, tampering with the reputation system to gain higher

voting rights, or colluding to manipulate consensus. Faced with the attack, each committee node must be bound to a Web3 DID (decentralized identity) or pass a KYC (know your customer) certification to ensure the uniqueness of each identity. The DID is linked to the reputation score. New nodes must undergo sufficient consensus cycles to accumulate a high reputation, preventing attackers from gaining control over the committee in a short period. Reputation scores are public and automatically adjusted based on historical behavior, preventing artificial manipulation that could lead to reputation abuse.

6.2. Performance Analysis of PBTMS

As shown in Table 8, the performance of the PBTMS has the advantages of high throughput, low response time, optimized gas consumption, lightweight proof size, and high scalability. It provides an efficient and cost-effective solution for e-commerce trading scenarios through technologies such as distributed storage and authentication, low-complexity consensus mechanisms, and zero-knowledge proofs. It should be noted that the number of black stars indicates the performance score in Table 8. TPS and scalability, though representing the best achievable performance under the current experimental setup, score lower due to consensus mechanism limitations and performance declines with increasing nodes. The other three indicators perform well and receive high scores.

Table 8. Performance metrics with custom star ratings.

Metric	Effect	Methods
High Throughput (TPS)	★★★☆☆	off-chain permission verification , off-chain storage
Low Response Time	★★★★★	Schnorr and Pedersen, ECC-AES slice encryption and storage, parallel task
Optimized Gas Consumption	★★★★★	off-chain calculations, IPFS index storage
Lightweight Proof Size	★★★★★	Bulletproofs, Pedersen commitments
High Scalability	★★★★☆	task separation, distributed storage, multi-node collaboration

* **Note:** The star ratings indicate the effectiveness of each metric. More black stars represent a higher rating and better performance.

In order to ensure the high throughput of the PBTMS, it adopts two measures, including that (i) PBTMS reduces the complexity of contract authorization by verifying permissions off-chain, and (ii) storing non-critical data to IPFS reduces stress on on-chain data storage. The response time of e-commerce trading in the PBTMS is rather low for three reasons. They are that (i) the system uses ECC to generate AES-shared keys for encryption and slices storage to speed up data encryption and upload, (ii) the efficient Schnorr signature and Pedersen commitment generation algorithms significantly reduce computation time, and (iii) authentication and authorization miners perform tasks in parallel to reduce latency. The PBTMS has optimized gas consumption since it uses the specific smart contract. Particularly, smart contract operations are streamlined by moving complex calculations (e.g., commitment validation) off-chain, reducing redundant on-chain calculation steps. Furthermore, smart contracts store only IPFS indexes to reduce uploaded data size. PBTMS uses bulletproofs and Pedersen commitment as proof mechanisms so that the proof size is significantly smaller than that of traditional schemes. The PBTMS also has high scalability because of two operations. The first one is that separating on-chain and

off-chain tasks allows the system to maintain low latency even after node expansion. The second one is that distributed storage and multi-node collaboration ensure high availability of data access and avoid a single point of failure.

7. Performance Evaluation

In this section, we perform extensive experiments to evaluate our designed system in terms of multiple evaluation metrics.

7.1. Experimental Setup

7.1.1. Experimental Environment

The performance testing of the PBTMS was conducted on an Intel Xeon 16-core CPU, 32 GB RAM, 1 TB SSD, running Ubuntu 20.04 LTS, using Ganache (7.9.2) and Truffle (5.11.5) as the blockchain platform, which is shown in Table 9.

Table 9. Experimental environment configuration.

Category	Configuration Details
Hardware	
CPU	Intel Xeon 16-core
RAM	32 GB
Storage	1 TB SSD
Software	
Operating System (OS)	Ubuntu 20.04 LTS
Ganache	version 7.9.2
Truffle	version 5.11.5
web3	version 1.10.0
IPFS	version 0.6.0
Programming language	
Python	version 3.10
Rust	version 1.83.0
Solidity	version 0.4.26

7.1.2. Parameter Setup

The experiments were conducted under well-defined parameters and categorized into blockchain settings and cryptographic components. Blockchain parameters included PoW consensus, a 10-node network, and a gas limit of 20,000,000, while cryptographic techniques utilized included Schnorr signatures, ECC-AES encryption, Bulletproof range proofs, and Pedersen commitments for privacy and efficiency. The detail is shown in Table 10.

Table 10. Blockchain parameters.

Parameter	Details
Blockchain Parameters	
Consensus Mechanism	PoW (Proof of Work)
Network Scale	Initial nodes: 10
Gas Limit	30,000,000
Cryptographic Components	
Schnorr Digital Signature	Signature size: 64 bytes
ECC-AES Encryption	ECC curve: secp256k1;
	AES key length: 128 bits
Bulletproof Range Proof	Input range: 0–100,000
Pedersen Commitment	Commitment size: 32 bytes

7.1.3. Comparison with Existing Schemes

We selected [42,58] as a comparison scheme. Ref. [58] targets similar e-commerce scenarios but uses a less efficient authentication method. Our system adopts a concise proof mechanism and streamlines delivery and receipt verification, significantly improving performance (see Section 7.2.3). Similarly, while [42] employs comparable encryption and proof processes, our use of an efficient signature mechanism and an on-chain/off-chain model reduces response time, as detailed in Section 7.2.1. These enhancements demonstrate the advantages of our framework in terms of performance and efficiency.

7.1.4. Evaluation Metrics

Performance testing focuses on the following key metrics:

- Response time: This refers to the time from trading initiation to completion, including the delay of each phase.
- Throughput (TPS): This refers to the number of tradings processed by the system per second.
- Proof size: This refers to the size of proofs such as Bulletproofs and the Pedersen commitment, which affects storage and transmission efficiency.
- Gas consumption: This represents the amount of gas consumed during smart contract execution, which is used to evaluate trading costs.

7.1.5. Scenarios

The following test scenarios were designed to fully evaluate the performance of the PBTMS:

- Encryption and proof efficiency: This simulates the generation of transaction information, the transaction amount, encryption of transactions, and proof generation process. It also evaluates the impact of message size and proof size on the complexity of transaction time. The expected results include the following:
  - As the data size increases, the encryption time grows linearly, with a relatively slow growth rate.
  - As the number of parallel transactions increases, the change in encryption time becomes smoother, indicating that the system has good scalability for parallel processing.
  - In each transaction processing stage, compared to existing solutions, the PBTMS system demonstrates significant optimization in time consumption, especially in encryption and proof generation efficiency.
- Throughput: This simulates the processing capability of concurrent uploads, downloads, and authorized transactions under different node quantities. The expected results include the following:
  - As the number of nodes increases, the system's throughput (TPS) is expected to show a gradual upward trend and will peak when a certain number of nodes is reached. At the same time, the download TPS will significantly exceed upload TPS and authorization TPS, indicating significant optimization in data processing efficiency, especially in data transmission and processing.
- Gas consumption: This measures the gas consumption of each entity at different transaction stages. The expected results include the following;
  - During the transaction, the gas consumption of each entity, compared to existing solutions, shows significant optimization in the PBTMS system. This optimization indicates effective improvements in resource utilization efficiency, reducing transaction costs and enhancing overall system performance.

- Proof size: This evaluates the impact of different proof sizes on the range proof. The expected results include the following:
  - In range proofs, as the proof range expands, the growth rate of proof size exhibits a slow linear increase, indicating that the proof scheme can effectively support large-scale transactions while maintaining high efficiency in scalability and performance.
  - The data size generated during the transaction, especially encrypted data and proof data, shows a significant reduction compared to existing solutions. This optimization demonstrates that the system has higher efficiency in data processing and storage, effectively reducing resource consumption and improving overall performance.
- Stress test: This simulates the system's capacity to handle high loads and evaluate its performance under high-concurrency transactions, increased data volumes, and changes in the number of nodes. The evaluation primarily focuses on simulating high-concurrency transactions, increasing data volumes, and adding nodes to assess the system's big data processing ability and scalability. The expected results include the following:
  - As concurrent transactions increase, the system throughput (TPS) is expected to rise gradually and eventually level off at a peak value after reaching a certain number of transactions.
  - As the system load increases, the response time for transactions may experience some delay, but the overall trend should remain within a reasonable range, indicating that the system can handle high loads.
  - Gas consumption is expected to increase as the load rises, but it should remain within a controllable range to avoid resource bottlenecks.

7.2. Experimental Results and Analysis

7.2.1. Encryption and Proof Efficiency

Our hybrid encryption scheme uses ECC to generate a shared key and AES to encrypt data, combining ECC's security with AES's efficiency. As shown in Figure 7, it performs similarly to AES for small data but is significantly faster for large data due to ECC's one-time key generation amortizing overheads. In contrast, AES's complex logic increases overhead in multi-threaded environments. This approach ensures high security and better performance for large data volumes.

In integrity verification, we use ECC cryptographic hashes for signature verification, and as can be seen from Figure 8, it shows a significant time advantage compared to the RSA scheme. Due to its lightweight characteristics, ECC cryptography has lower computational complexity in signing and verifying. It especially scales better in highly parallel scenarios, with a much lower time increase than RSA. Therefore, the ECC scheme is more suitable for multipoint consensus and high-concurrency scenarios, such as blockchain transaction verification or IoT authentication.

The scheme proposed by [59] spends 593.84ms in the trading processing part, compared to the total trading processing time of 506.9ms for our proposed scheme, significantly improving time performance. Moreover, according to Figure 9, compared with [42], the scheme proposed by PBTMS shows higher efficiency in all phases, especially in the decryption and signature phases, where the optimization is most significant. Overall, PBTMS is more suitable for application scenarios with high-performance requirements.

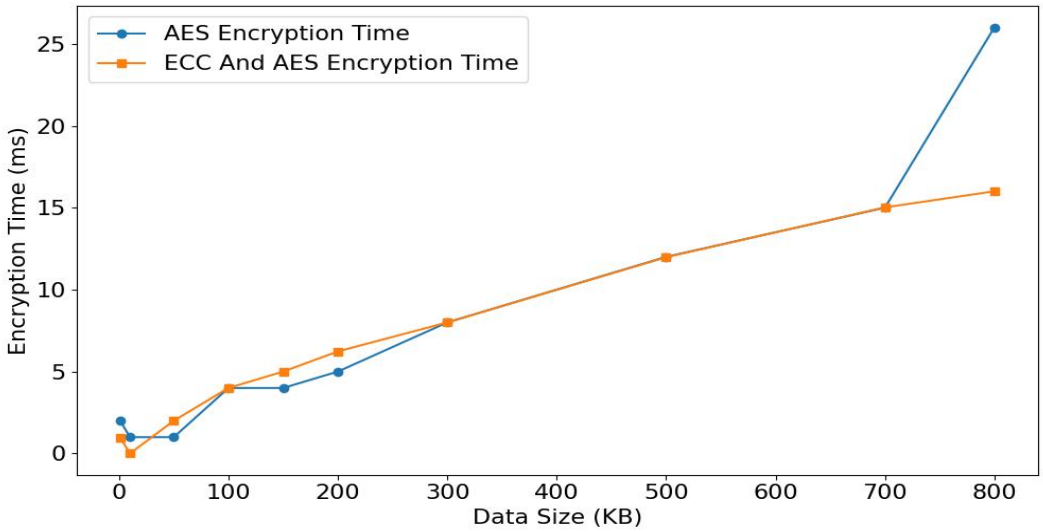


Figure 7. Encryption time vs. data size.

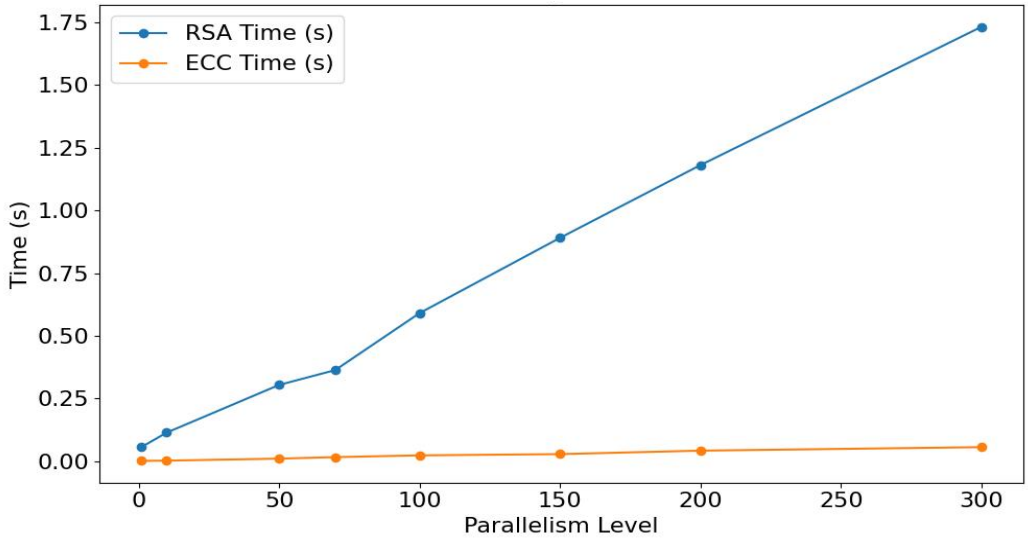


Figure 8. Time vs. increasing parallelism levels.

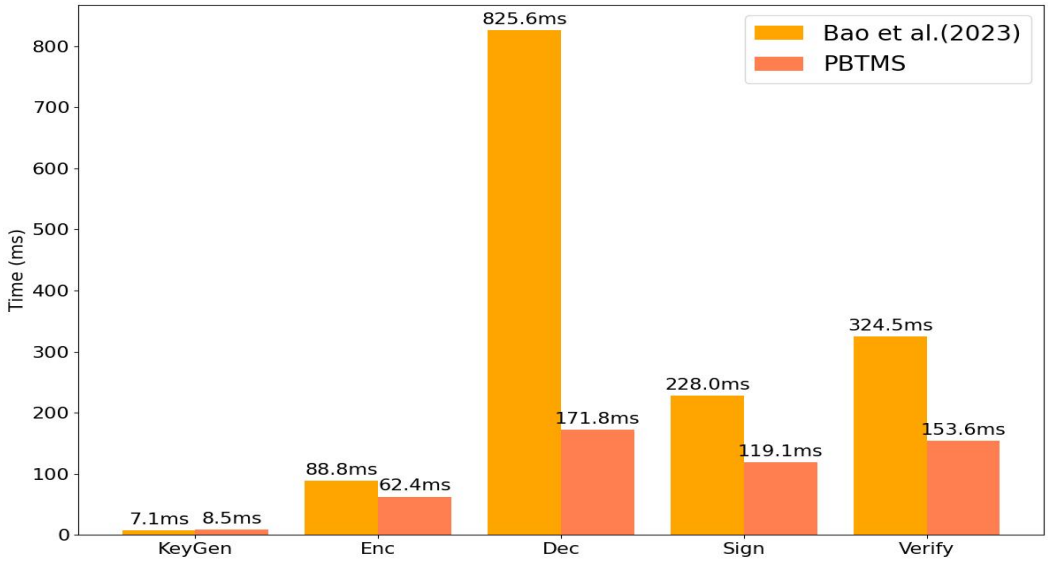


Figure 9. Time costs for all phases Based on [42].

7.2.2. Throughput (TPS)

The test is conducted for three task scenarios: download, upload, and authorization, with the number of nodes set to 10, 20, 40, 60, 80, and 100, respectively.

As shown in Figure 10, TPS peaks at around 20 nodes, with download TPS reaching 750 and upload and authorization TPS at 50% and 45% of that, respectively. This efficiency stems from effective multithreading (eight cores per node), controlled consensus overhead, and balanced computational loads. However, as nodes increase, quadratic growth in consensus time and rising network latency reduce TPS, exposing limitations in the consensus mechanism. While performance is strong with a medium number of nodes, optimizing the consensus algorithm and network efficiency remains a key area for future research.

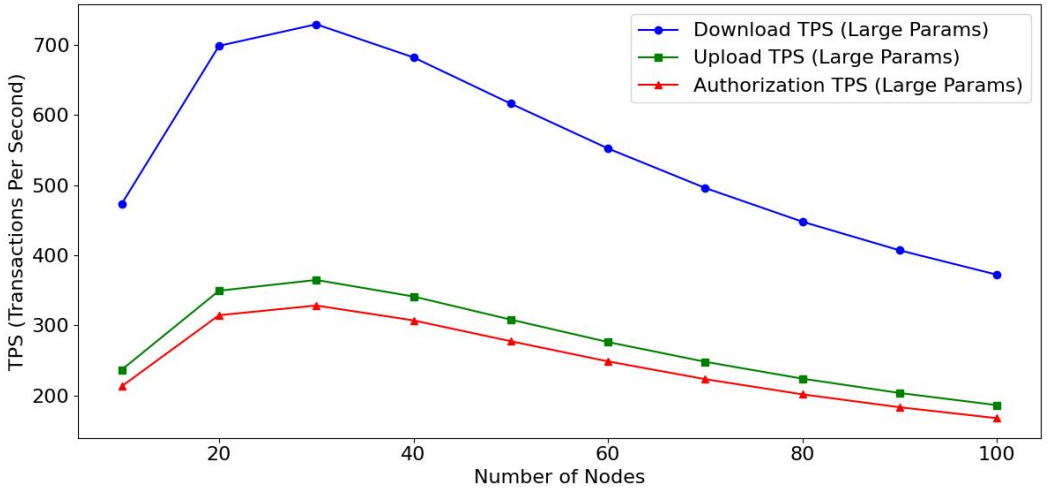


Figure 10. TPS vs. number of nodes.

7.2.3. Gas Consumption

Table 11 shows the test results of PBTMS's gas consumption. Combined with the data in Figure 11, it can be seen that compared with [58], PBTMS shows higher efficiency in all entity operations, and total resource consumption, especially the Gas consumption of miners, is significantly reduced, which fully reflects its optimization capability. The aggregate data further illustrates the superior overall resource utilization of PBTMS.

Table 11. The execution costs and duration for all entities in the on-chain phase.

Entities	Transactions	GasUsed	Costs
Miners	8	211,640	Certificate Upload Authorization * 2 + Trading Information Upload Authorization + Trading Information Download Authorization + Proof of Receipt and Delivery Authorization * 2 + Trading Amount Upload Authorization + Trading Amount Download Authorization
Seller	2	160,940	Upload proof of identity + Upload proof of delivery
Buyer	4	321,880	Upload proof of identity and balance + Upload Trading Information + Upload Trading amount + Upload Proof of Receipt
Total 14 694,460			

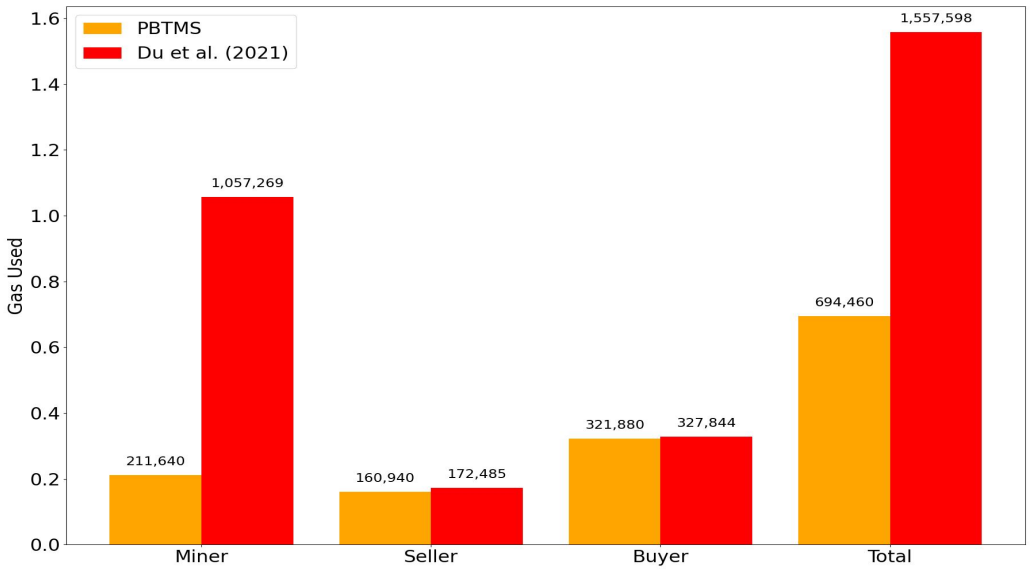


Figure 11. Comparison of gas used by entities based on [58].

7.2.4. File Size Impact on Performance

In our design, most cryptographic operations produce fixed-size outputs for efficiency and predictability due to having stable sizes derived from hashes and private keys. The only variable-sized proof is the Bulletproof balance proof, which grows logarithmically with range size, as shown in Figure 12. Unlike Sigma and CT proofs, which grow linearly and result in significantly larger sizes, Bulletproof maintains compactness, ensuring low resource consumption and minimal overhead. This makes it ideal for scenarios with large ranges or extensive participant data, where scalability and efficiency are critical.

As shown in Figure 13, the results highlight that PBTMS significantly reduces data sizes compared to [42]'s approach, demonstrating its efficiency in blockchain transactions. PBTMS reduces encrypted data size by 52.5% (6336 to 3008 bytes) and proof data size by 70% (4672 to 1403 bytes), showcasing the effectiveness of its ECC-AES encryption and compact proof mechanisms. While the data size slightly increases (16 to 18 bytes), this minor trade-off ensures enhanced functionality.

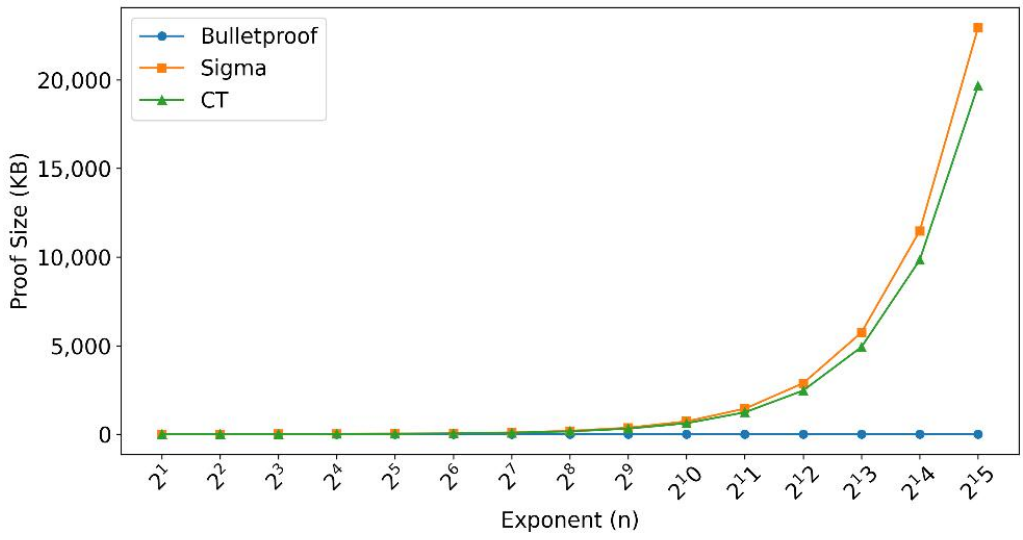


Figure 12. Proof sizes vs. range.

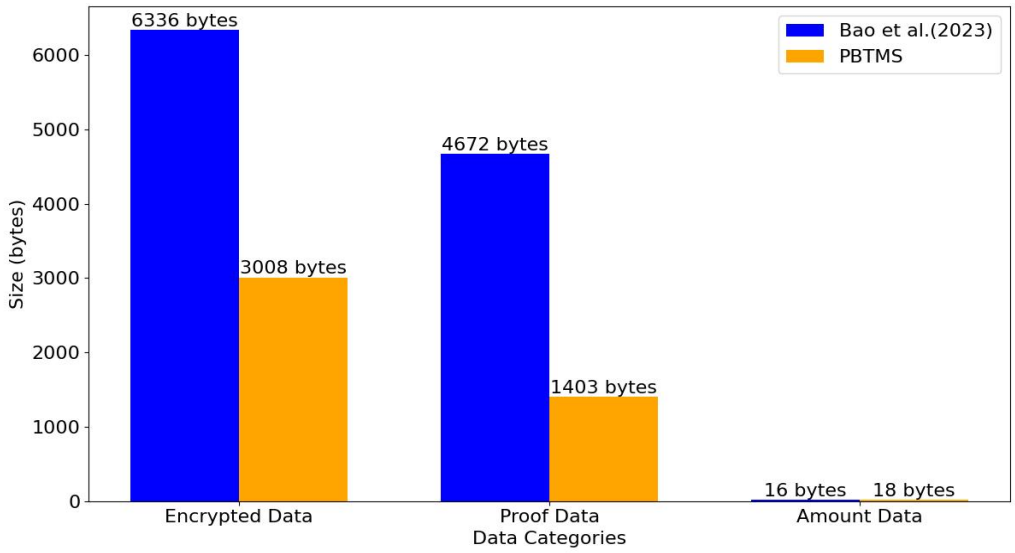


Figure 13. Comparison of data sizes based on [42].

7.2.5. Stress Test

The stress test results show that the system exhibits strong high-load capability and scalability. As the number of concurrent transactions increases, response time gradually rises but stays within a reasonable range. As shown in the Figure 14, when reaching 1000 concurrent transactions, the response time is 977.3 ms, indicating the system can handle high-concurrency transactions with stable performance. The increase in response time is mainly due to resource contention and consensus delays, but the system mitigates this through multi-node parallel processing, optimized consensus mechanisms, and efficient resource allocation.

Similarly, as shown in Figure 15, gas consumption increases from 694,460 to nearly 833,352 with higher concurrency. This rise reflects the added demand for CPU and bandwidth, particularly during storage and consensus processes. However, the system effectively controls gas growth through multi-container deployment and resource optimization, keeping it within a reasonable range and confirming good scalability. Figure 16 demonstrates that TPS increases as concurrency rises, peaking and stabilizing around 300 concurrent transactions. Even near the system's load limits, TPS remains high, thanks to optimized parallel processing, resource distribution, and consensus mechanisms, ensuring the system maintains stable performance under high load.

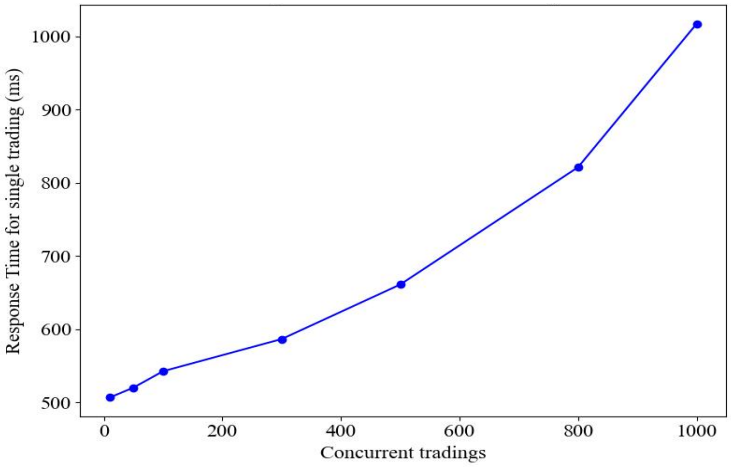


Figure 14. Response time vs. concurrent tradings.

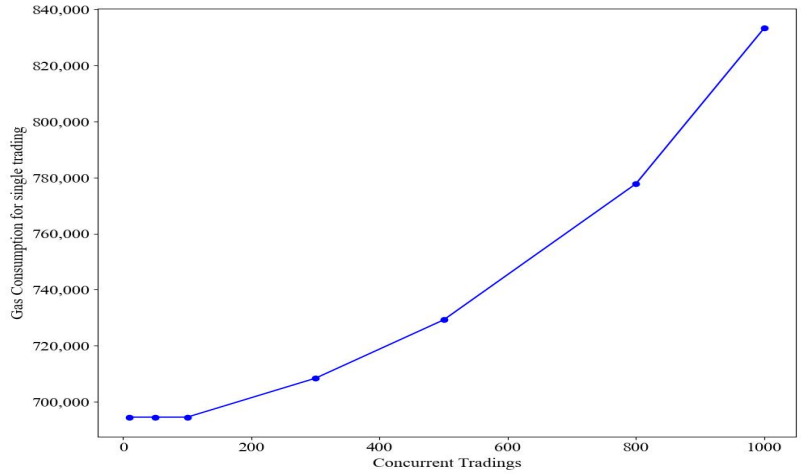


Figure 15. Gas consumption vs. concurrent tradings.

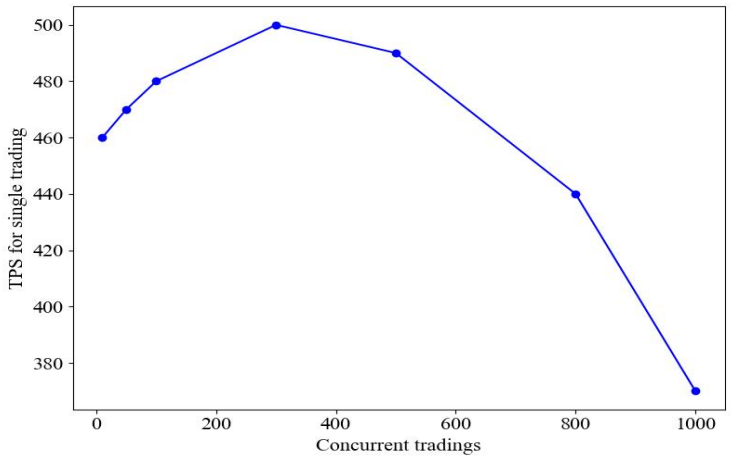


Figure 16. TPS vs. concurrent tradings.

8. Conclusions

In this paper, we proposed the PBTMS system, a blockchain-based e-commerce transaction platform designed to address key challenges such as node trustworthiness, privilege abuse, and the balance between privacy protection and performance. By leveraging decentralized data storage and management, PBTMS effectively prevents third-party access to sensitive user and transaction information, ensuring both transaction validity and privacy. Through its innovative integration of multiple cryptographic techniques and decentralized architectures, the system achieves a robust solution for secure, privacy-preserving transactions.

The major highlight of PBTMS lies in its multi-agent collaborative framework, which incorporates a range of key participants, including buyers, sellers, trusted authorities, miners, blockchain, and IPFS. This framework enables efficient integration of decentralized storage, permission management, and transaction validation, reducing security risks and enhancing the overall system efficiency. In contrast to existing solutions, PBTMS minimizes the reliance on centralized authorities, offering a more secure and transparent model for e-commerce transactions.

However, despite the notable improvements in privacy protection and transaction validation, PBTMS still faces certain limitations. Specifically, the consensus mechanism can face throughput bottlenecks in high-concurrency scenarios, particularly as the number of nodes increases. Furthermore, while the system performs well in simulated environments, further real-world validation and optimization are necessary, particularly for large-scale e-commerce platforms.

The PBTMS system holds significant potential for application across various sectors. It can be effectively deployed in cross-border payment systems to ensure secure, decentralized, and privacy-preserving transactions. Additionally, it offers a strong privacy protection mechanism for decentralized finance (DeFi) applications, enhancing security and preventing sensitive data leaks. In e-commerce platforms, PBTMS can provide robust privacy protections while ensuring transaction authenticity and compliance, thereby fostering greater user trust. Moreover, the system's features are highly applicable to supply chain management, where it can maintain privacy while ensuring data integrity and transparency throughout the supply chain.

Overall, PBTMS represents a significant advancement in addressing the challenges of privacy, security, and performance in e-commerce transactions, with considerable potential for future improvements and broader applications in various domains.

Author Contributions: Conceptualization, R.Z.; methodology, R.Z.; validation, R.Z., Y.L., and L.F.; formal analysis, Y.L.; investigation, R.Z.; resources, R.Z.; data curation, R.Z.; writing—original draft preparation, R.Z.; writing—review and editing, R.Z., Y.L., and L.F.; supervision, Y.L.; project administration, R.Z., Y.L., and L.F.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Beijing Natural Science Foundation Project of China under Grant L222045.

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request. Due to privacy and ethical considerations, some data may be restricted.

Acknowledgments: We would like to thank the Key Laboratory of General Wireless Communications of the Ministry of Education, Beijing University of Posts and Telecommunications, for their valuable support and resources provided throughout this research. Finally, we extend our gratitude to all fellow students and collaborators for their insightful discussions and constructive feedback, which significantly contributed to the completion of this work.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Hu, L.; Zhang, B.; Zhang, P.; Qi, J.; Cao, J.; Gao, D.; Zhao, H.; Feng, X.; Wang, Q.; Zhuo, L.; et al. A Virtual Character Generation and Animation System for E-Commerce Live Streaming. In Proceedings of the ACM Multimedia Conference, Virtual, 20–24 October 2021; pp. 1202–1211. [CrossRef]
2. Zhang, Y.; Liu, Y.; Xiong, H.; Liu, Y.; Yu, F.; He, W.; Xu, Y.; Cui, L.; Miao, C. Cross-Domain Disentangled Learning for E-Commerce Live Streaming Recommendation. In Proceedings of the 39th IEEE International Conference on Data Engineering, ICDE, Anaheim, CA, USA, 3–7 April 2023; pp. 2955–2968. [CrossRef]
3. Zhang, Y.; Feng, Y.; Zhou, W.; Ye, Y.; Tan, M.; Xiao, R.; Tang, H.; Ding, J.; Yu, J. Multi-Domain Deep Learning from a Multi-View Perspective for Cross-Border E-commerce Search. In Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI, Vancouver, BC, Canada, 20–27 February 2024; pp. 9387–9395. [CrossRef]
4. Xing, F.; Peng, G.; Liang, Z. Research on the Application of Blockchain Technology in the Cross-border E-Commerce Supply Chain Domain. In Proceedings of the 10th International Conference on Distributed, Ambient and Pervasive Interactions. Smart Living, Learning, Well-Being and Health, Art and Creativity, Virtual, 26 June–1 July 2022; Volume 13326, pp. 99–109. [CrossRef]
5. Sun, J.; Ge, N.; Chen, X.; Feng, W.; Lu, J. Truthful transaction protocol for e-commerce networks based on double auction. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 709–722. [CrossRef]
6. Yazdinejad, A.; Dehghantanha, A.; Karimipour, H.; Srivastava, G.; Parizi, R.M. A Robust Privacy-Preserving Federated Learning Model Against Model Poisoning Attacks. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 6693–6708. [CrossRef]
7. Yazdinejad, A.; Dehghantanha, A.; Parizi, R.M.; Hammoudeh, M.; Karimipour, H.; Srivastava, G. Block Hunter: Federated Learning for Cyber Threat Hunting in Blockchain-Based IIoT Networks. *IEEE Trans. Ind. Inform.* **2022**, *18*, 8356–8366. [CrossRef]

8. Shen, H.; Wu, G.; Xia, Z.; Susilo, W.; Zhang, M. A Privacy-Preserving and Verifiable Statistical Analysis Scheme for an E-Commerce Platform. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 2637–2652. [CrossRef]
9. Yu, J.; Wang, H.; Wang, X.; Li, Z.; Qin, L.; Zhang, W.; Liao, J.; Zhang, Y.; Yang, B. Temporal Insights for Group-Based Fraud Detection on e-Commerce Platforms. *IEEE Trans. Knowl. Data Eng.* **2025**, *37*, 951–965. . [CrossRef]
10. Pagey, R.; Mannan, M.; Youssef, A.M. All Your Shops Are Belong to Us: Security Weaknesses in E-commerce Platforms. In Proceedings of the ACM Web Conference, Austin, TX, USA, 30 April–4 May 2023; pp. 2144–2154. [CrossRef]
11. Tao, S.; Liu, Y.; Sun, C. Examining the inconsistent effect of privacy control on privacy concerns in e-commerce services: The moderating role of privacy experience and risk propensity. *Comput. Secur.* **2024**, *140*, 103794. [CrossRef]
12. Dijesh, P.; Babu, S.; Vijayalakshmi, Y. Enhancement of e-commerce security through asymmetric key algorithm. *Comput. Commun.* **2020**, *153*, 125–134. [CrossRef]
13. Albshaier, L.; Almarri, S.; Hafizur Rahman, M. A review of blockchain's role in E-Commerce transactions: Open challenges, and future research directions. *Computers* **2024**, *13*, 27. [CrossRef]
14. Kim, S.I.; Kim, S.H. E-commerce payment model using blockchain. *J. Ambient Intell. Humaniz. Comput.* **2022**, *13*, 1673–1685. [CrossRef]
15. Zou, Y.; Mhaidli, A.H.; McCall, A.; Schaub, F. "I've Got Nothing to Lose": Consumers' Risk Perceptions and Protective Actions after the Equifax Data Breach. In Proceedings of the Fourteenth Symposium on Usable Privacy and Security, SOUPS, Baltimore, MD, USA, 12–14 August 2018; pp. 197–216.
16. Yurcik, W.; Schick, A.; North, S.C.; Gastner, M.T.; de Miranda, F.R.; da Silva Avelino, R.; de Moraes Batista, A.F.; Pluta, G.; Brooks, I. Cybersecurity Monitoring/Mapping of USA Healthcare (All Hospitals)—Magnified Vulnerability due to Shared IT Infrastructure, Market Concentration, & Geographical Distribution. In Proceedings of the 2024 Workshop on Cybersecurity in Healthcare, HealthSec, Salt Lake City, UT, USA, 14–18 October 2024; pp. 45–52. [CrossRef]
17. Deng, S.; Cheng, G.; Zhao, H.; Gao, H.; Yin, J. Incentive-Driven Computation Offloading in Blockchain-Enabled E-Commerce. *ACM Trans. Internet Technol.* **2021**, *21*, 9:1–9:19. [CrossRef]
18. Wu, X.; Wu, T.; Khan, M.; Ni, Q.; Dou, W. Game Theory Based Correlated Privacy Preserving Analysis in Big Data. *IEEE Trans. Big Data* **2021**, *7*, 643–656. [CrossRef]
19. Xie, W.; Zhou, W.; Kong, L.; Zhang, X.; Min, X.; Xiao, Z.; Li, Q. ETTF: A Trusted Trading Framework Using Blockchain in E-commerce. In Proceedings of the 22nd IEEE International Conference on Computer Supported Cooperative Work in Design, CSCWD, Nanjing, China, 9–11 May 2018; pp. 612–617. [CrossRef]
20. Liu, C.; Xiao, Y.; Javangula, V.; Hu, Q.; Wang, S.; Cheng, X. NormaChain: A Blockchain-Based Normalized Autonomous Transaction Settlement System for IoT-Based E-Commerce. *IEEE Internet Things J.* **2019**, *6*, 4680–4693. [CrossRef]
21. Sun, Y.; Zhang, R.; Xue, R.; Su, Q.; Li, P. A Reputation Based Hybrid Consensus for E-Commerce Blockchain. In Proceedings of the 27th International Conference on Web Services, Honolulu, HI, USA, 18–20 September 2020; Volume 12406, pp. 1–16. [CrossRef]
22. Sun, Y.; Xue, R.; Zhang, R.; Su, Q.; Gao, S. RTChain: A Reputation System with Transaction and Consensus Incentives for E-commerce Blockchain. *ACM Trans. Internet Technol.* **2021**, *21*, 15:1–15:24. [CrossRef]
23. Zhou, Z.; Wang, M.; Yang, C.; Fu, Z.; Sun, X.; Wu, Q.M.J. Blockchain-based decentralized reputation system in E-commerce environment. *Future Gener. Comput. Syst.* **2021**, *124*, 155–167. [CrossRef]
24. Li, M.; Zhu, L.; Zhang, Z.; Lal, C.; Conti, M.; Alazab, M. Anonymous and Verifiable Reputation System for E-Commerce Platforms Based on Blockchain. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 4434–4449. [CrossRef]
25. Guan, Z.; Wang, N.; Fan, X.; Liu, X.; Wu, L.; Wan, S. Achieving Secure Search over Encrypted Data for e-Commerce: A Blockchain Approach. *ACM Trans. Internet Technol.* **2021**, *21*, 12:1–12:17. [CrossRef]
26. Li, G.; Fan, Z.; Wu, X. The Choice Strategy of Authentication Technology for Luxury E-Commerce Platforms in the Blockchain Era. *IEEE Trans. Eng. Manag.* **2023**, *70*, 1239–1252. [CrossRef]
27. Takakubo, T.; Li, R.; Nan, H.; Jin, Q.; Su, Z.; Wu, H. ECPAS: A Blockchain-based E-Commerce Price Auditing System. In Proceedings of the IEEE International Conference on Communications, ICC, Denver, CO, USA, 9–13 June 2024; pp. 1334–1339. [CrossRef]
28. Li, G.; Fan, Z.; Zhao, Q.; Sun, M. Blockchain Technology Application in an E-Commerce Supply Chain: Privacy Protection and Sales Mode Selection. *IEEE Trans. Eng. Manag.* **2024**, *71*, 8060–8074. [CrossRef]
29. Dahal, S.B. Enhancing e-commerce security: The effectiveness of blockchain technology in protecting against fraudulent transactions. *Int. J. Inf. Cybersecur.* **2023**, *7*, 1–12.
30. Taherdoost, H.; Madanchian, M. Blockchain-based e-commerce: A review on applications and challenges. *Electronics* **2023**, *12*, 1889. [CrossRef]
31. Ghesmati, S.; Fdhila, W.; Weippl, E. Analyzing UTXO-based blockchain privacy threats. *Cryptol. ePrint Arch.* 2023, *Preprint*.
32. Sasson, E.B.; Chiesa, A.; Garman, C.; Green, M.; Miers, I.; Tromer, E.; Virza, M. Zerocash: Decentralized anonymous payments from bitcoin. In Proceedings of the IEEE Symposium on Security and Privacy, San Jose, CA, USA, 18–21 May 2014; pp. 459–474. [CrossRef]

33. Li, Y.; Yang, G.; Susilo, W.; Yu, Y.; Au, M.H.; Liu, D. Traceable monero: Anonymous cryptocurrency with enhanced accountability. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 679–691. [CrossRef]
34. Bünz, B.; Agrawal, S.; Zamani, M.; Boneh, D. Zether: Towards privacy in a smart contract world. In Proceedings of the International Conference on Financial Cryptography and Data Security, Kota Kinabalu, Malaysia, 10–14 February 2020; Volume 12059, pp. 423–443. [CrossRef]
35. Guan, Z.; Wan, Z.; Yang, Y.; Zhou, Y.; Huang, B. BlockMaze: An efficient privacy-preserving account-model blockchain based on zk-SNARKs. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 1446–1463. [CrossRef]
36. Wu, X.; Ding, Y.; Zhou, X.; Xu, Y.; Wang, S.; Xu, X.; Qi, L. Fuzzy Federated Learning for Privacy-Preserving Detection of Adolescent Idiopathic Scoliosis. *IEEE Trans. Fuzzy Syst.* **2024**, *32*, 5493–5507. [CrossRef]
37. Li, Y.; Ye, C.; Hu, Y.; Morpheus, I.; Guo, Y.; Zhang, C.; Zhang, Y.; Sun, Z.; Lu, Y.; Wang, H. ZKCPlus: Optimized fair-exchange protocol supporting practical and flexible data exchange. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Virtual, 15–19 November 2021; pp. 3002–3021. [CrossRef]
38. Li, B.; Wang, Y. RZKPB: A Privacy-Preserving Blockchain-Based Fair Transaction Method for Sharing Economy. In Proceedings of the International Conference on Trust, Security and Privacy in Computing and Communications, New York, NY, USA, 31 July–3 August 2018; pp. 1164–1169. [CrossRef]
39. Dziembowski, S.; Eckey, L.; Faust, S. Fairswap: How to fairly exchange digital goods. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 967–984. [CrossRef]
40. Wen, B.; Wang, Y.; Ding, Y.; Zheng, H.; Qin, B.; Yang, C. Security and privacy protection technologies in securing blockchain applications. *Inf. Sci.* **2023**, *645*, 119322. [CrossRef]
41. Tian, J.; Zhao, Y.; Yang, X.; Zhao, X.; Chen, R.; Yu, Y. Identity-based threshold (multi) signature with private accountability for privacy-preserving blockchain. *High-Confid. Comput.* **2024**, *4*, 100271. [CrossRef]
42. Bao, Z.; He, D.; Wei, W.; Peng, C.; Huang, X. Ledgermaze: An efficient privacy-preserving non-interactive zero-knowledge scheme over account-model blockchain. *IEEE Trans. Comput.* **2023**, *72*, 3489–3502. [CrossRef]
43. Li, D.; Ke, X.; Zhang, X.; Zhang, Y. A trusted and regulated data trading scheme based on blockchain and zero-knowledge proof. *IET Blockchain* **2024**, *4*, 443–455. [CrossRef]
44. Golightly, L.; Modesti, P.; Garcia, R.; Chang, V. Securing distributed systems: A survey on access control techniques for cloud, blockchain, IoT and SDN. *Cyber Secur. Appl.* **2023**, *1*, 100015. [CrossRef]
45. Ali, G.; ElAffendi, M.; Ahmad, N. BlockAuth: A blockchain-based framework for secure vehicle authentication and authorization. *PLoS ONE* **2023**, *18*, e0291596. [CrossRef]
46. Wu, N.; Xu, L.; Zhu, L. A blockchain based access control scheme with hidden policy and attribute. *Future Gener. Comput. Syst.* **2023**, *141*, 186–196. [CrossRef]
47. Zhang, Y.; Wang, X.; He, X.; Zhang, N.; Zheng, Z.; Xu, K. Galaxy: A Scalable BFT and Privacy-Preserving Pub/Sub IoT Data Sharing Framework Based on Blockchain. *IEEE Internet Things J.* **2023**, *11*, 5222–5236. [CrossRef]
48. Chen, C.; Yang, G.; Li, Z.; Xiao, F.; Chen, Q.; Li, J. Privacy-Preserving Multi-Party Cross-Chain Transaction Protocols. *Cryptography* **2024**, *8*, 6. [CrossRef]
49. Malik, S.; Dedeoglu, V.; Kanhere, S.S.; Jurdak, R. PrivChain: Provenance and privacy preservation in blockchain enabled supply chains. In Proceedings of the IEEE International Conference on Blockchain (Blockchain), Espoo, Finland, 22–25 August 2022; pp. 157–166. [CrossRef]
50. Fuchsbauer, G.; Wolf, M. Concurrently secure blind schnorr signatures. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, 26–30 May 2024; pp. 124–160. [CrossRef]
51. Goldfeder, S.; Bonneau, J.; Gennaro, R.; Narayanan, A. Escrow Protocols for Cryptocurrencies: How to Buy Physical Goods Using Bitcoin. In Proceedings of the International Conference on Financial Cryptography and Data Security, Sliema, Malta, 3–7 April 2017; pp. 321–339. [CrossRef]
52. Wang, D.; Zhao, J.; Wang, Y. A survey on privacy protection of blockchain: The technology and application. *IEEE Access* **2020**, *8*, 108766–108781. [CrossRef]
53. Zheng, S.; Luo, J.; Dong, E.; Chen, C.; Liu, X. SPENDER: A platform for secure and privacy-preserving decentralized P2P e-commerce. *arXiv* **2022**, arXiv:2206.07215. [CrossRef]
54. Lee, S.S.; Lee, S.J. Design and implementation of Web3-Based e-commerce cryptocurrency payment system. In Proceedings of the IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Naples, Italy, 4–6 December 2023; pp. 303–308. [CrossRef]
55. Xiao, L.; Greenstein, L.J.; Mandayam, N.B.; Trappe, W. Channel-based detection of Sybil attacks in wireless networks. *IEEE Trans. Inf. Forensics Secur.* **2009**, *4*, 492–503. [CrossRef]
56. Zhao, W.; Yang, X.; Qi, S.; Wei, J.; Dong, X.; Yang, X.; Qi, Y. Secure blockchain-based reputation system for IIoT-enabled retail industry with resistance to sybil attack. *Future Gener. Comput. Syst.* **2025**, *166*, 107705. [CrossRef]

57. Hou, R.; Yu, H.; Sun, Y. Selfied: Sybil defense in permissionless blockchains via in-protocol bandwidth consumption. *Comput. Netw.* **2025**, *256*, 110890. [CrossRef]
58. Du, Y.; Xu, C.; Zhang, Y. A blockchain-based online transaction system for physical products trading with fairness, privacy preservation, and auditability. In Proceedings of the IEEE 9th International Conference on Smart City and Informatization (iSCI), Shenyang, China, 20–22 October 2021; pp. 15–22. [CrossRef]
59. Lin, C.; Huang, X.; Ning, J.; He, D. Aca: Anonymous, confidential and auditable transaction systems for blockchain. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 4536–4550. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.