This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2025.3547143

Decentralized and Fair Trading via Blockchain: The Journey So Far and the Road Ahead

Hao Zeng, Helei Cui, Man Li, Bo Zhang, Chengjun Cai, Zhiwen Yu, and Bin Guo

Abstract—Centralized trading platforms have long been the preferred choice for users, despite growing concerns regarding data privacy. Users have to place their trust in these platforms and provide sensitive personal information, like identities and financial accounts. However, these centralized platforms often lack transparency, making it challenging to ensure fairness, privacy, and security against both external and internal risks. In contrast, a decentralized fair trading paradigm, harnessing the potential of blockchain technology, is rapidly emerging. It empowers individuals to engage in the exchange of digital assets with others while guaranteeing fairness, efficiency, and privacy. In this paper, we conduct a comprehensive survey of decentralized fair trading. We commence by providing fundamental definitions of fair trading and tracing its evolution over time. We then delve into the essential framework of on-chain and off-chain trading and highlight key improvements that enhance the efficiency of decentralized fair trading within various application scenarios. Furthermore, we undertake a thorough analysis of privacy and security enhancements within the scope, summarizing defenses against known attacks. Finally, we outline the challenges and offer insights into the future prospects of decentralized fair trading, with the aim of inspiring the development of more innovative and promising designs in this evolving trend.

Index Terms—Decentralized fair trading, blockchain, privacy-preserving, zero-knowledge proof.

1 Introduction

With the development of big data and artificial intelligence, more and more valuable data are required to be traded, such as financial data, medical data, and social media data, which is collected and utilized to promote the development of society [1]. Institutions conduct trading generally through online market platforms (*e.g.*, Gnip [2], Factual [3], Terbine [4]) to obtain relevant data to expand their business and make decisions that maximize benefits.

As numerous trading is conducted over the Internet, concerns regarding trading fairness increasing in importance. The **fairness** means that it is infeasible for malicious buyers to obtain digital commodities or services from sellers without payments and malicious sellers can't get payments without providing valid digital commodities or services.

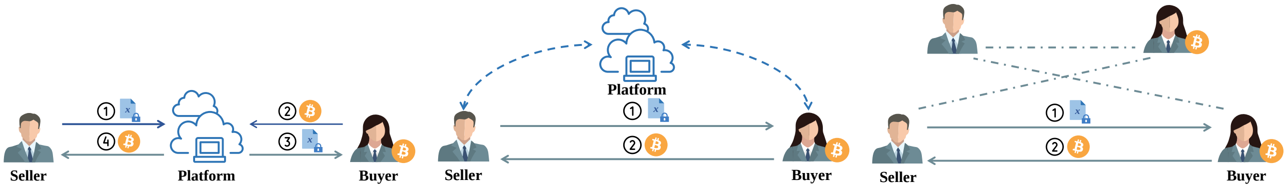
However, some malicious behaviors may disrupt trading fairness and compromise the participants' benefits. For instance, the malicious seller may sell invalid digital commodities or services that compromise the buyer's benefit. While the malicious buyer may resell digital commodities provided by the seller without the latter's approval.

As demonstrated in Fig. 1(a), to resist the malicious behaviors mentioned above and ensure that trading is conducted correctly, the traditional approach is to conduct trading on a centralized trading market platform consisting of the market platform, seller, and buyer. Specifically, the seller transmits the digital commodities to a mutually trusted platform and sets an appropriate price. The buyer peruses commodities within the platform's interface and chooses the expected commodities before initiating an order with accompanying payment. Upon receiving the payment from the buyer, the platform sends commodities to the buyer and transfers the payment to the seller simultaneously, and it gets the commission from trading. The platform can verify whether digital commodities from the seller are consistent with the description. Moreover, the platform holds the function of detecting the duplication and alteration of commodities to detect whether commodities have been resold or not, thus guaranteeing the benefit of participants. However, participants are concerned about the powerful control of the platform, because they need to trust the platform completely and are required to provide sensitive information such as the resident identity number and bank card number. The malicious platform may covertly analyze and process the commodity and identity information of the user, without the latter's approval.

To circumvent these concerns, some designs are proposed to reduce dependence on the platform, thus reducing the impact of the platform's powerful control. As shown in Fig. 1(b), the market platform is not involved in trading unless a participant deviates from the expected behavior which needs the platform to arbitrate and maintain fairness [5], [6], [7], [8], [9], [10]. We refer to these designs as **partially decentralized fair trading**. However, it only reduces the dependence on the platform and does not eliminate the impact of the platform's powerful control.

Blockchain [11], as a chain-based data structure, incorporates distributed network, cryptographic, and consensus algorithms to achieve the features of transparency, immutability, and reliability. Since blockchain can guarantee the correctness and validity of commodities and trading

- *H. Zeng, H. Cui, M. Li, B. Zhang, and B. Guo are with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China. E-mail: {hao.zeng, liman, bo.zhang}@mail.nwpu.edu.cn, {chl, guob}@nwpu.edu.cn (Corresponding author: H. Cui)*
- *C. Cai is with City University of Hong Kong (Dongguan), Songshan Lake, China. E-mail: chengjun.cai@cityu-dg.edu.cn*
- *Z. Yu is with Northwestern Polytechnical University, Xi'an 710129, China, and also with Harbin Engineering University, Harbin 150009, China. E-mail: zhiwenyu@nwpu.edu.cn*



(a) centralized fair trading (b) partially decentralized fair trading (c) decentralized fair trading

Fig. 1. Simplified trading for digital commodity.

information through consensus algorithms, it presents a feasible approach to implementing decentralized fair trading without relying on platforms. As the pioneering instantiation of blockchain, Bitcoin capitalizes on the consensus-based distributed network to facilitate decentralized fair trading for digital commodities and services, devoid of the central issuer [12], [13], [14], [15]. As illustrated in Fig. 1(c), complex trading can be executed in a decentralized manner through smart contracts without involving the platform. Smart contracts are executed autonomously by distributed nodes of the blockchain, ensuring that no single participant controls or manipulates smart contracts' behavior. We refer to these designs as **decentralized fair trading**, also known as **blockchain-based fair trading**.

Researchers promote the development of decentralized fair trading by combining it with diverse application scenarios. For example, decentralized fair trading for digital commodities (*e.g.*, digital signatures, machine learning models, crowdsensing data, and IoT data) and decentralized fair trading for digital service (*e.g.*, outsourcing services, computing resources, and electricity resources). Furthermore, extensive designs are proposed to improve the efficiency of decentralized fair trading for different application scenarios.

The fundamental design of decentralized fair trading relies primarily on Ethereum smart contract [16], [17] and Bitcoin script [13], [18]. Although these designs liberate users from the constraints and control of platforms, the transparent nature of blockchain may expose sensitive information such as trading information and commodity information. To bolster the privacy and security of decentralized fair trading, researchers engage in comprehensive studies, integrating some advanced privacy-preserving computation techniques. We analyze the enhancements for privacy and security in decentralized fair trading combined with specific scenarios in Section 5.

In this work, we aim to provide a clear knowledge and foundation for researchers and practitioners interested in understanding, applying, and expanding decentralized fair trading designs and application scenarios. The main contributions of this survey can be summarized as follows:

- We present the basic definitions and development of fair trading, and summarize the fundamental design of decentralized fair trading, providing an entry point for researchers interested in the field.
- We detail the fundamental framework of on-chain trading and off-chain trading, and explore the enhancements for the efficiency of decentralized fair trading for diverse application scenarios.
- We analyze the enhancements for privacy and security in decentralized fair trading for diverse scenarios combined with different techniques and summarize the protection against common attacks.
- We summarize the challenges and future perspectives of decentralized fair trading to motivate researchers to explore and improve the efficiency, privacy, and security of decentralized fair trading.

Organizations. The background and fundamental design of decentralized fair trading are introduced in Section 2. The on-chain and off-chain trading for diverse application scenarios are discussed in Section 3 and Section 4. The enhancements for privacy and security in decentralized fair trading are presented in Section 5. The challenges and future perspectives are summarized in Section 6. Finally, Section 7 concludes the whole paper.

2 Overview of Decentralized Fair Trading

In this section, we commence by presenting some basic definitions in fair trading, followed by the development of fair trading, and finally provide the fundamental design of decentralized fair trading.

2.1 Basic Definitions in Fair Trading

2.1.1 Property Definitions

We provide definitions for two properties in fair trading: fairness and completeness. These definitions closely align with [8], [18], [19].

1) Fairness. The fairness means that the trading is resilient to any malicious behaviors. Specifically, for the malicious buyer $\mathcal{B}$, it is impossible to obtain digital commodities or services from the seller $\mathcal{S}$ without payments; while for the malicious $\mathcal{S}$, it is infeasible to get payments without providing valid digital commodities or services. Particularly, if the malicious $\mathcal{B}$ resells commodities without the $\mathcal{S}$'s approval, $\mathcal{S}$ is able to get enough compensation from $\mathcal{B}$; while if the malicious $\mathcal{S}$ fails to provide valid commodities or services, $\mathcal{B}$ is able to get enough compensation from $\mathcal{S}$.

2) Completeness (a.k.a correctness, soundness). The completeness means that if participants act honestly, $\mathcal{B}$ obtains digital commodities/services, and $\mathcal{S}$ receives payments.

2.1.2 Blockchain Definitions

As a decentralized, public, and tamper-proof ledger, blockchain relies on core components like the transaction, consensus, smart contract, and oracle to support fair trading.

1) Transaction. ❶ **On-chain transactions** occur on the blockchain, including the transfer of Cryptocurrencies such as BTC and ETH from one address to another, which are transparently and immutably recorded on the blockchain.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2025.3547143

Moreover, on-chain transactions serve as records of interactions among participants, maintaining integrity, trust, and security without relying on centralized platforms. ❷ **Off-chain transactions** are processed through secondary layers or other settlement methods, which handle partial transactions off the blockchain to improve transaction efficiency and reduce fees. Thus, it can enhance scalability and alleviate congestion in blockchain networks, ideal for small-scale and high-frequency payments. Based on on-chain and off-chain transactions, we divide decentralized fair trading into on-chain trading and off-chain trading.

2) Consensus Mechanism. The blockchain state is updated transparently and securely through consensus mechanisms, eliminating the need for centralized platforms and ensuring a unique and unambiguous transaction sequence. Common consensus mechanisms include Proof-of-Work [11], which secures the network through computation efforts; Proof-of-Stake [20], where decisions are made based on staked assets; Byzantine fault tolerance [20], used in permissioned blockchains for fault tolerance. Each mechanism balances security, efficiency, and scalability to suit different blockchain platforms and application requirements.

3) Smart Contract. The smart contract is a self-executing program stored on blockchain that automatically enforces the terms of an agreement when predefined conditions are met. For example, the scripting language implemented in Bitcoin restricts contract complexity to basic conditions like multi-signatures and time-locks [11]; Ethereum provides the Ethereum virtual machine, which is Turing Complete, supporting more complex and dynamic smart contracts [21]. In decentralized fair trading, the smart contract automates the agreements as predefined without the need for centralized platforms, minimizing potential fraud and disputes.

4) Oracle Service. The oracle service acts as a bridge that fetches and supplies real-world data from outside sources (*e.g.*, weather conditions and stock prices) to the blockchain in a secure and verifiable manner [21]. With accurate and reliable data, the smart contract can trigger actions such as adjusting prices and executing trading. However, relying on a single oracle node may compromise the core principle of decentralization by introducing a single point of failure and data manipulation. To ensure reliability, security, and integrity, it is crucial to adopt decentralized oracles that aggregate data from multiple independent sources. For example, Chainlink [22], a typical decentralized oracle network, uses multiple independent oracle nodes to provide accurate and reliable data for smart contracts.

2.2 The Development of Fair Trading

Centralized fair trading, which relies on market platforms to offer digital commodities for users, is nearing maturity. For example, as an open platform, Factual selects to concentrate on geographical and place data. Gnip and Datasift specialize in offering social media data streams, in particular Twitter. These platforms incentivize users to share their data, resulting in enhanced data quality (*e.g.*, geographical and place data, and social media data streams), while acting as arbiters to guarantee fairness and completeness in trading.

However, the users of platforms need to trust these platforms completely. It raises concerns about the potential leakage of sensitive information. For instance, powerful platforms can access digital commodities uploaded by $\mathcal{S}$ without the latter's approval, which may lead to the leakage of commodities. Additionally, they can obtain users' identity information, and even resell sensitive information for profit.

As demonstrated by extensive research, fair trading without a platform is impossible without additional assumptions [23], [24]. To address this inherent challenge, partially decentralized fair trading (a weaker security model) has been explored, where the platform doesn't need to be involved in the trading and is only necessary for arbitration in the case of trading anomalies [5], [6], [7], [8], [9], [10].

However, partially decentralized fair trading still relies on the platform when disagreements and anomalies arise in trading. Thus, there are still some trust issues, for example, the platform has the ability to access users' sensitive information and even digital commodities without approval from the latter, and it is impractical to find a fully trusted platform to serve as the arbiter in real-world trading.

Fortunately, the emergence of blockchain [11] offers a noval approach to implementing fair trading. One may view blockchain-based solutions as improved partially decentralized fair trading, where blockchain takes the role of the platform. The decentralized, transparent, and immutable nature of blockchain provides a feasible approach to avoiding relying on platforms during the trading process and preventing malicious behaviors of platforms and participants. Therefore, we refer to it as decentralized fair trading or blockchain-based fair trading.

2.3 Fundamental Design of Decentralized Fair Trading

Centralized fair trading and partially decentralized fair trading have many trust issues due to the presence of potentially malicious platforms. To avoid relying on the platform and enable $\mathcal{S}$ to have control over the digital commodity x, many studies propose decentralized fair trading based on the blockchain [13], [16], [17], [18], [25], [26], [27], [28]. A unifying aspect of these designs is that smart contracts are deployed on the blockchain to play the role of the platform to handle disagreements and anomalies that arise in trading, thus implementing decentralized fair trading.

The fundamental design of decentralized fair trading primarily draws upon the utilization of Ethereum smart contract [29] or Bitcoin script [11]. In the following, we present fundamental designs of decentralized fair trading based on different public blockchains and conduct an in-depth comparative analysis.

2.3.1 Ethereum-based Fundamental Design

The formalization and mathematical model of decentralized fair trading based on Ethereum smart contract is as follows. The digital commodity x is partitioned into n file chunks, *i.e.*, $x = (x_1, ..., x_n)$, where x_i is a bit string of length λ and is represented as the leaf node of the Merkle Tree. Model the predicate functions ϕ as a circuit with m gates, where each gate is an element in the set of operations Γ, taking $x = (x_1, ..., x_n)$ as input, where x satisfies the predicate functions ϕ, *i.e.*, $\phi(x) = 1$.

The main cryptographic primitive is the hash function combined with the algorithms **Commit** and **Open** to submit a commitment of the digital commodity. Moreover,

TABLE 1: Comparison among fundamental designs of partially decentralized fair trading and decentralized fair trading.

Design	Compatibility			Verification		Proof size	Communication complexity	On-chain arbitration complexity	Rounds
	Platform	Bitcoin	Ethereum	Phase	Method				
Asokan's design [5], [8], [9]	●	○	○	#AT	Commit-and-open	$6\|\sigma\|$	$10\|\sigma\|$	N/A	7
Bao's design [6]	●	○	○	#BT	CEMBS	$\|\sigma\| + \|k\|\lambda$	$5\|\sigma\| + 3k\lambda$	N/A	3
Ateniese's design [7]	●	○	○	#BT	Verifiable encryption	$2\|\sigma\|$	$7\|\sigma\|$	N/A	3
Küpçü's design [10]	●	○	○	#AT	Verifiable escrow	$4\|\sigma\| + 4\|\mathbb{H}\|$	$10\|\sigma\| + 8\|\mathbb{H}\| + 2k\lambda$	N/A	7
Fairswap [16]	○	○	●	#AT	Proof of misbehavior	$3\lambda \log \|k\|$	$\|k\|\lambda + 3\|\mathbb{H}\| + 3\|\sigma\| + \|\mathbb{G}\|$	$\mathcal{O}(\lambda \log \|k\|)$	5
Optiswap [17]	○	○	●	#AT	Proof of misbehavior	$2(\lambda + \|\sigma\|) \log \|k\|$	$(a_\phi + 1)\|k\|\lambda + 3\|\mathbb{H}\| + 3\|\sigma\| + \|\mathbb{G}\|$	$\mathcal{O}((\lambda + \|\sigma\|) \log \|k\|)$	$5 + 2a_\phi$
FairTrade [25]	○	○	●	#BT	Random sampling	$\|k'\|(\lambda + \|\sigma\|)$	$\|k\|\lambda + (\|k\| + \|k'\|)\|\sigma\|$	$\mathcal{O}(\|\sigma\|)$	3
Li's design [26]	○	○	●	#AT	PCE	$\|k'\|\lambda + 4\|\sigma\|$	$(\|k\| + \|k'\|)\lambda + 6\|\sigma\|$	$\mathcal{O}(\|k'\|\lambda)$	3
Delgado's design [27]	○	●	○	#BT	Cut-and-choose	$\|k'\|\lambda$	$(\|k\| + \|k'\|)\lambda + 5\|\sigma\|$	$\mathcal{O}(\|\sigma\|)$	5
Bentov's design [13]	○	●	○	#AT	Claim-or-refund	$\|\mathbb{H}\| + \|\sigma\|$	$4k\lambda a_\phi + \|\mathbb{H}\| + 4\|\sigma\|$	$\mathcal{O}(\|\sigma\|)$	3
FairComp [28]	○	●	○	#AT	Commit-and-open	$6\|\sigma\|$	$6\|\sigma\| + 2\|\mathbb{H}\| + 6\|\mathbb{G}\|$	$\mathcal{O}(\|\sigma\|)$	6
BPay [18]	○	●	●	#AT	Checking-proof	$\|\mathbb{H}\| \log \|k\|$	$\|k\|\lambda + \|\sigma\| \log \|k\| + \|\mathbb{H}\| \log \|k\|$	$\mathcal{O}(\|\sigma\| \log \|k\|)$	7

● indicates compatibility; ○ indicates incompatibility; k denotes the set of data chunks; k' denotes the subset of k; λ denotes the size of each encrypted data chunk;
$\mathbb{H}$ refers to the size of a single hash function output; $\mathbb{G}$ refers to the group element; $|\sigma|$ refers to the signature size; a_ϕ denotes the challenge limit property for every circuit;
N/A means not applicable; CEMBS: Certificate of Encrypted Message Being a Signature; PCE: Plaintext Checkable Encryption; #AT: After trading; #BT: Before trading.

Encode, Extract, and **Judge** algorithms formally construct decentralized fair trading based on Ethereum smart contract. The following is the sequence of the trading process.

(i) $\mathcal{B}$ deposits p coins and a predicate function ϕ into the smart contract. $\mathcal{S}$ uses **Encode** to generate the encoding of x, then a commitment c about x and an opening value d are generated by **Commit**.

(ii) $\mathcal{B}$ can obtain the encoding of x and the decryption key k after depositing p coins. $\mathcal{S}$ can obtain p coins by publishing x that satisfies $\phi(x) = 1$.

(iii) $\mathcal{B}$ can extract x by **Extract**, which takes the encoding of x as input, and then check whether $\phi(x) = 1$ via **Open**. If $\phi(x) \neq 1$, $\mathcal{B}$ will send an arbitration request to the smart contract.

(iv) The smart contract verifies whether x satisfies $\phi(x) = 1$ via **Judge** and subsequently determines whether to return p coins to $\mathcal{B}$ based on the result.

The fundamental design of decentralized fair trading based on Ethereum smart contract can refer to [16], [17], [25], [26] and the code implementation of these designs can refer to FairTrade [1] [25], FairSwap [2] [16], and OptiSwap [3] [17]. Although the design performs well when participants remain honest, it suffers from escalating the on-chain cost for verification in case of disputes, which constrains the efficiency, particularly for large-scale digital commodities.

2.3.2 Bitcoin-based Fundamental Design

The formalization and mathematical model of decentralized fair trading based on Bitcoin script is similar to the design based on Ethereum smart contract. Every user i has an elliptic curve digital signature algorithm key pair, denoted as (pk_i, sk_i), where sk_i and pk_i are used to sign and verify the transactions. The Bitcoin script, the stack-based scripting language, allows $\mathcal{B}$ flexibility in defining the condition on how the transaction containing p coins be redeemed. It also requires the algorithms **Commit** and **Open** to submit the commitment of the digital commodity and signature. The procedural sequence of the trading process is as follows.

(i) $\mathcal{B}$ indicates to $\mathcal{S}$ the digital commodity he is willing to buy. Upon reception, $\mathcal{S}$ generates a pair key (pk_s, sk_s) and sends pk_s, the digital commodity x encrypted by pk_s, a commitment c of x generated by **Commit**, a correctness proof f of x and the price p to $\mathcal{B}$.

(ii) $\mathcal{B}$ receives the correctness proof f of x from $\mathcal{S}$ and then verifies the correctness proof.

(iii) $\mathcal{B}$ requests a signature sig_{prev} over a nonce message performed with sk_s generated by $\mathcal{S}$ and verifies that the signature is correct by pk_s.

(iv) $\mathcal{B}$ builds a private key locked transaction to perform the atomic exchange between sk_s and p coins. $\mathcal{S}$ can obtain p coins by publishing x that satisfies **Open**$(c, d, x) = 1$.

The fundamental design of decentralized fair trading based on Bitcoin script can refer to [13], [18], [27], [28]. However, the Bitcoin scripting language is not Turing Complete, and the scripts it generates are more complex for large-scale digital commodities. In contrast, Solidity [21] in Ethereum is Turing Complete, which means it can perform more complex and advanced computation operations in a more straightforward and convenient manner.

As illustrated in Table 1, we summarize some fundamental designs of partially decentralized fair trading and decentralized fair trading. Specifically, we compare each design in terms of compatibility, verification phase & method, proof size, communication complexity, on-chain arbitration complexity, and rounds. The verification phase & method analyze how each design ensures the validity of digital commodities/services. The proof size and communication complexity accurately quantify proofs/messages delivered between participants. The on-chain arbitration complexity leverages $\mathcal{O}$ notation instead of specific values to measure the storage for resolving disputes on the blockchain, avoiding being influenced by implementation details or external factors. The rounds reflect the number of interactions for trading, where fewer rounds mean the trading is more efficient. Thus, we adopt the compatibility and rounds to analyze the latency of different designs accurately, which avoids the inherent differences in the underlying architecture (*i.e.*, centralized platforms and blockchains[11], [29]).

3 On-Chain Trading

Decentralized fair trading (*i.e.*, blockchain-based fair trading) is initially aimed at small-scale digital commodities without the involvement of platforms. However, real-world application scenarios are complex and diverse, not limited

[1] https://github.com/DougZaoldyeck/FairTrade
[2] https://github.com/lEthDev/FairSwap
[3] https://github.com/CryBtoS/OptiSwap

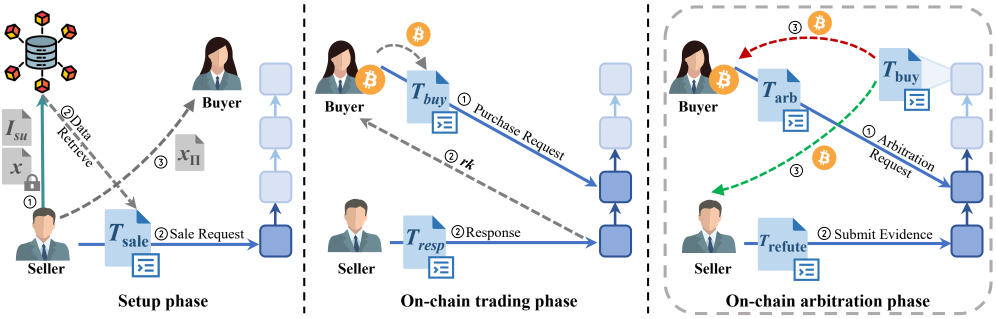


Fig. 2. The fundamental framework of on-chain trading. The new transaction is contained in the new block, which is represented in dark blue, and the old block in light blue. The gray dashed border indicates that the on-chain arbitration phase is optional. During the on-chain arbitration phase, the green line represents that the behavior of $\mathcal{S}$ is as expected and receives the rewards; while the red line represents $\mathcal{B}$ succeeds in arbitration and receives the payments.

to small-scale digital commodities. The trading object may encompass digital commodities (*e.g.*, lightweight digital signatures, massive crowdsensing data and IoT data) that are intuitive commodities and digital services (*e.g.*, outsourcing services, computing resources, and electricity resources) that can't be directly integrated into the blockchain. Based on the definitions of on-chain and off-chain transactions, we divide decentralized fair trading into on-chain and off-chain trading. We first summarize a fundamental framework of on-chain trading by integrating similar phases and functions from existing studies [16], [17], [19], [25], [26]. Then, we explore the enhancements made for on-chain trading combined with diverse scenarios.

3.1 Fundamental Framework of On-Chain Trading

On-chain trading guarantees that all transactions are recorded on the blockchain in real time. In brief, on-chain trading necessitates the utilization of smart contracts to ensure fairness and autonomy in trading, where $\mathcal{B}$ is obligated to pay for digital commodities/services and $\mathcal{S}$ is required to deliver correct commodities/services for rewards.

The fundamental framework of on-chain trading is built on the underlying blockchain, usually including three entities: seller $\mathcal{S}$, buyer $\mathcal{B}$, and miner $\mathcal{M}$. Drawing from existing studies [16], [17], [19], [26], [25], we take the on-chain trading for digital commodities as an example and divide it into three phases, *i.e.*, setup phase, on-chain trading phase, and on-chain arbitration phase (optional), as shown in Fig. 2.

Setup phase: The user registers and obtains his identity public-private key pair. By hashing the identity public key, the user's account address can be obtained, which is used to receive rewards.

i) $\mathcal{B}$ and $\mathcal{S}$ obtain their identity public-private key pairs $(pk_{\mathcal{B}}, sk_{\mathcal{B}})$ and $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ after registration.

ii) $\mathcal{S}$ sends the encrypted digital commodity x and the description information I_{su} into the decentralized storage network (such as IPFS, Storj, and Filecoin). Meanwhile, $\mathcal{S}$ creates a smart contract for x and submits transaction T_{sale} onto the blockchain, which contains the price p, the description information I_{su}, the hash root of x, the commitment c of the key k used for encrypting x, and the three time windows t_1, t_2, t_3.

iii) $\mathcal{B}$ leverages the cut-and-choose protocol [27], [25], zero-knowledge proofs [30], [31], or other cryptographic methods to verify the correctness of x without exposing sensitive information. The verification is an off-chain operation, performed by $\mathcal{B}$ and $\mathcal{S}$ with a set of pieces or a zero-knowledge proof from the plaintext digital commodity, which is denoted as x_{Π}.

On-chain trading phase: Upon miners successfully confirming transaction T_{sale}, the smart contract procedurally transitions into the on-chain trading phase.

i) $\mathcal{B}$ generates transaction T_{buy} by invoking the purchase function in the smart contract, accompanied by transmitting p coins to initiate a request for purchase.

ii) $\mathcal{S}$ responds to $\mathcal{B}$ by invoking the selling function and generating the transaction T_{resp} within the time window t_1. He encrypts k to obtain rk using $pk_{\mathcal{B}}$, which is formally denoted as $rk = \textbf{Enc}_{pk_{\mathcal{B}}}(k)$. Subsequently, he transmits T_{resp} which contains rk onto the blockchain.

iii) After receiving T_{resp}, $\mathcal{B}$ can decrypt rk using $sk_{\mathcal{B}}$ and subsequently verify whether k satisfies condition $\textbf{Open}(c, d, k) = 1$ within the time window t_2. Then he leverages k to decrypt the encrypted digital commodity and compares x with the description information.

iv) If $\mathcal{B}$ doesn't dispute the correctness of x before the time window t_3, the p coins will be transferred to $\mathcal{S}$ and the transactions will be recorded on the blockchain.

(Optional) On-chain arbitration phase: If any of the verifications fail, *e.g.*, k doesn't consist with the commitment value, x doesn't match the description, and so on, $\mathcal{B}$ can initiate a complaint for T_{resp} within the time window t_2 and the smart contract will formally enter the on-chain arbitration phase. Both $\mathcal{B}$ and $\mathcal{S}$ are required to submit evidence substantiating the legality of their respective actions by generating transactions T_{arb} and T_{refute}, respectively. The p coins remain locked in the smart contract until miners verify the evidence from $\mathcal{B}$ and $\mathcal{S}$.

i) $\mathcal{B}$ initiates the creation of transaction T_{arb} by invoking the arbitration function. The transaction T_{arb} contains the key k satisfying the condition $\textbf{Enc}_{pk_{\mathcal{S}}}(k) = rk$. In cases where k fails to meet the condition $\textbf{Open}(c, d, k) = 1$, then p coins will be transferred to $\mathcal{B}$. Otherwise, the p coins will continue being locked until $\mathcal{S}$ submits transaction T_{refute}.

ii) $\mathcal{S}$ is supposed to generate transaction T_{refute} via the seller-refute function within the time window t_3. The transaction should contain the hash path H_i to the root. In cases where all the hash values along the path are verified successfully, then the p coins will be transferred to $\mathcal{S}$. However, if the hash of any two child nodes in the hash path does not match the hash value of their parent node, then the p coins will be refunded to $\mathcal{B}$. The trading result will be recorded on the blockchain.

From the presentation of the fundamental framework above, it is clear that all transactions generated during trading are recorded on the blockchain in real time, which is the key feature of on-chain trading. However, the requirements of on-chain trading vary in different application scenarios, leading to corresponding adjustments based on the fundamental framework. In the following, we demonstrate the enhancements made for on-chain trading by summarizing the requirements of diverse application scenarios and providing on-chain trading designs for specific application scenarios.

3.2 Enhancements for On-Chain Trading

In this section, we review on-chain trading for digital commodities (such as digital signatures, machine learning models, crowdsensing data, IoT data, and cryptocurrencies) and digital services (such as outsourcing services, computing resources, electricity resources, and bandwidth resources), and summarize the enhancements made for on-chain trading. Both are represented and traded as digital forms, and the value is derived either from the asset itself or the inherent utility. However, digital commodities are typically transferred as assets with intrinsic value [16], [17]. The participants' main interest revolves around retaining, trading, or leveraging these commodities as possessions that can change hands, much like traditional goods with ownership. By contrast, digital services focus on providing access to resources instead of ownership [19]. The buyer pays for resource access and usage, while the seller controls the underlying infrastructure. Digital services require precise management of resource allocation, availability, and service-level guarantees, as participants have expectations regarding performance, reliability, and scalability. Thus, the distinction lies in whether a buyer is purchasing assets with tangible or quantifiable value (digital commodities) or obtaining access to resources without acquiring ownership (digital services). The comparison among on-chain trading for different application scenarios is shown in Table 2.

3.2.1 Digital Commodities

Digital commodities encompass a diverse array of digital assets, ranging from the minutest digital signatures to the vast Internet of Things (IoT) data. Existing studies regarding decentralized fair trading for digital commodities mainly focus on digital signatures, machine learning (ML) models, crowdsensing data, IoT data, and cryptocurrencies.

Digital signatures. One crucial requirement of online contract signing is that participants can fair trading for digital signatures. However, the process of fair trading for digital signatures is asynchronous, which inevitably brings a certain degree of unfairness [63]. Due to the pivotal role that digital signatures play in any commercial transaction and their ease of representation with modest data volume, decentralized fair trading prioritizes the utilization of digital signatures (including RSA [64], Fiat-Shamir [65], etc). Since blockchain acts as a decentralized and trusted platform, it is employed to conduct on-chain trading for digital signatures that are demanded in online contract signing [32], [33], [34], [35]. On-chain trading for digital signatures is similar to the fundamental framework in Section 3.1, but it needs both participants to pay the deposit to the smart contract because they conduct trading for digital signatures with each other. However, the majority of on-chain trading designs for online contract signing primarily cater to two-party contract signing. When endeavors are made to extend it to encompass multi-party contract signing, the participants are required to verify each signature, thereby culminating in a substantial burden of verification [63]. Therefore, efficient on-chain trading for multi-party contract signing has become a pivotal challenge in online contract signing. Researchers introduce verifiably encrypted signatures [66], which can derive other digital signatures together with secret factors, utilizing the blockchain to achieve on-chain trading for secret factors, thus reducing the burden of verification [63].

ML models. To address data scarcity and data quality challenges, extensive studies focus on collecting high-quality data for training ML models via on-chain trading (refer to on-chain trading for IoT data) [67], [68], [69], [70]. Some studies also focus on conducting decentralized fair trading for ML models, which can be integrated with Machine Learning as a Service [71], [72] to enable $\mathcal{B}$ to obtain models from $\mathcal{S}$, incentivizing the circulation of good-quality models. On-chain trading for ML models requires $\mathcal{S}$ to commit his model onto the blockchain with a model report. Then, miners rigorously validate the model's performance through the benchmark samples to ensure its reliability, and the transaction of the validation report is recorded on the blockchain. However, on-chain trading for ML models raises concerns, such as ensuring models' accuracy matches $\mathcal{S}$'s claims and performing correct model benchmarking with privacy protection [36], [37], [38]. To tackle these concerns, zero-knowledge proofs [73] and trusted execution environments (TEE) [74] are applied to on-chain trading for ML models [31], [36], [38]. The ML model from $\mathcal{S}$ is loaded in a local TEE, and TEE checks whether or not the loading model is indeed the one committed by $\mathcal{S}$. If the checking result is true, the process continues to run the model evaluation on the samples by utilizing a trusted hardware-based zero-knowledge proof (*i.e.*, sealed-glass proofs [75]). The transactions generated during trading, including the evaluation results, are recorded on the blockchain.

Crowdsensing data. The employment of crowd wisdom for knowledge discovery and monetization has become increasingly popular nowadays [70], [76], [77]. However, crowdsensing usually faces several critical challenges such as inaccurate data from $\mathcal{S}$, dishonest ratings from $\mathcal{B}$, and vulnerability of centralized platforms [39], [68]. To solve these challenges, many on-chain trading designs are proposed to implement decentralized fair trading for crowdsensing data [40], [41], [42]. The basic strategy is leveraging smart contracts and zero-knowledge proofs to verify data quality without revealing data and identities, and enforce fair trading between workers' data and requesters' coins [41]. During the on-chain trading, $\mathcal{B}$ (requesters) deploy the task smart contract on the blockchain to recruit $\mathcal{S}$

TABLE 2: Comparison among on-chain trading for different application scenarios.

	Features	Scenario	Schemes	Advantages	Disadvantages
Digital commodities	Market-driven pricing Digital representations of tangible assets Provide ownership transfer of commodities Represent transferable assets (with ownership)	Digital signatures	Zhang *et al.* [32] Alper *et al.* [33] Wan *et al.* [34] Tian *et al.* [35]	Affordable on-chain storage costs	Complex off-chain identity integration Potential privacy concerns on public ledgers
		ML models	Golden Grain [36] FCH [37] ZKCMP [38]	Verifiable authenticity of model parameters Pricing by models' authentic performances	High (on-chain) computation costs for model validations Complex verification of model provenance
		Crowdsensing data	BCDT [39] PPQC [40] ZebraLancer [41] Dragoon [42]	Transparent reward and incentive mechanisms Encourage knowledge discovery by crowd wisdom	High block latency in verifying data High (on-chain) computation costs in verifying data
		IoT data	Zhao *et al.* [43] Kang *et al.* [44] Lin *et al.* [45] Li *et al.* [46]	Transparent reward and incentive mechanisms Automated trustless micropayment settlement	High throughput and storage demands Require robust off-chain infrastructure
		Cryptocurrencies	ACCS [47] Tian *et al.* [48] SPCEX [49]	Facilitate the liquidity of cryptocurrencies Support cross-chain operations of cryptocurrencies	High price volatility Require continuous interaction
Digital services	Subscription-based pricing Digital representations of intangible assets Provide on-demand and customizable functions Represent access to resources rather than ownership	Outsourcing services	zkQuery [50] OBFP [19] FEPOD [51] SC-RDoC [52]	Trustless verification for outsourcing computation Non-interactive verification Fraud resistance integrity assurance	Heavy proof size Expensive on-chain storage costs Expensive on-chain verification costs
		Computing resources	Sun *et al.* [53] Du *et al.* [54] BBDNS [55]	On-demand services Full use of computing resources Fair and dynamic resource pricing	High on-chain costs for resource matching Hard reflecting real-time status of computing resources
		Electricity resources	PETCON [56] V2GEx [57] PRAC [58]	On-demand services Full use of electricity resources	Frequent blockchain interactions High on-chain costs for supply-demand matching
		Bandwidth resources	Gringotts [59] VFD [60], [61] Lakhani *et al.* [62]	Quantify bandwidth contribution Encourage bandwidth sharing Timely Reliable delivery	Expensive on-chain costs Frequent blockchain interactions

(workers) to complete the sensing task. $\mathcal{S}$ submits crowdsensing data via a transaction pointing to the smart contract's address. $\mathcal{B}$ evaluates the data quality and provides proof to the smart contract to determine the reward for $\mathcal{S}$. To overcome the inefficiency of generic zero-knowledge proofs, Dragoon [42] restructures various non-trivial statements and converts encrypted data quality verification into particular verifiable decryption. Moreover, existing studies integrate homomorphic encryption [39], [78] and truth discovery [79], [80] into smart contracts to derive the average quality of crowdsensing data and rating of $\mathcal{S}$, and detect untruthful data and ratings. To detect whether $\mathcal{B}$ measures crowdsensing data (provided by $\mathcal{S}$) and returns the truthful qualities, the smart contract computes the encrypted rating scores by homomorphic encryption and conducts the verification on encrypted rating scores. However, the smart contract incurs substantial gas and delays in verifying encrypted rating scores and confirming transaction results, resulting in inefficient trading [68], [39].

IoT data. Data generated by IoT devices is centralized at cloud servers that can be traded for various data-intensive applications. To address transparency and trust issues, some on-chain trading designs for IoT data are proposed [43], [44], [45], [46], [81], similar to those for crowdsensing data. However, IoT data is generally large-scale, and has different requirements and challenges depending on specific scenarios, including guaranteeing the availability of IoT data and low latency. Reputation-based mechanisms combined with similarity learning or three-weight subjective logic model are proposed to select more reliable IoT sources and improve data availability [43], [44]. The double auction mechanism is utilized to increase incentives for participants to trade data and maximize social welfare [46], [39]. Mobile edge computing is integrated into network edge infrastructures to support massive data storage and minimize latency [44].

Cryptocurrencies. While cryptocurrencies facilitate secure and trustless transactions among participants, they remain isolated from one another, as blockchains lack the ability to communicate or exchange data with external systems [49]. Centralized exchanges are often the preferred routes for executing cryptocurrency transfers, but they may undermine the decentralized nature of cryptocurrencies [48]. To overcome this, some on-chain trading designs for cryptocurrencies are proposed [47], [48], [49]. As the most straightforward approach, atomic cross-chain swaps enable decentralized cryptocurrency trading via hashed timelock contracts [47]. Participants lock their cryptocurrencies in smart contracts on their respective blockchains, and the trading can only be completed when they fulfill the conditions within the specified time. If either participant fails, the trading will be canceled. However, this approach is interactive, inefficient, and relies on synchrony assumption. To avoid synchrony assumption, existing studies leverage Ethereum smart contracts for decentralized cryptocurrency trading, linking different cryptocurrencies by ETH and randomly selecting users to act as intermediaries [48], [49].

Discussion. A reliable pricing mechanism is essential to reflect the value of digital commodities like ML models and IoT data especially in a decentralized system. It helps mitigate price manipulation, ensures that $\mathcal{B}$ and $\mathcal{S}$ engage in fair trading, and fosters trust within the ecosystem. Well-designed pricing mechanisms can incentivize market participation and maintain supply-demand equilibrium. Specifically, the quality-based pricing mechanism [36], which uses the performance metrics to measure the model quality, can maximize the revenue of $\mathcal{S}$ and the utility of $\mathcal{B}$, for maintaining the long-term running of the model marketplace. The Stackelberg-based pricing mechanism [82] allows the leader (typically $\mathcal{S}$) to set prices that maximize its revenue while considering the reactions of the followers (typically $\mathcal{B}$), which can balance competition and foster stability.

3.2.2 Digital Services

Unlike digital commodities mentioned above, digital services represent intangible trading objects, such as outsourcing services, computing resources, electricity resources, etc., which cannot be stored in decentralized storage networks.

Outsourcing services. During on-chain trading for outsourcing services, $\mathcal{B}$ (the resource-constrained user) delegates complex computation tasks to one or more $\mathcal{S}$ (the resource-abundant worker) through smart contracts. To ensure the integrity and correctness of outsourcing computations, verifiable computation is required, enabling $\mathcal{B}$ to verify that $\mathcal{S}$ has performed the tasks correctly without supervising the whole process [83]. A common approach is using zero-knowledge proofs to generate proofs of computation results, which are deployed in smart contracts for verification without revealing private information [50]. The first practical implementation of the zero-knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK) proof system is Pinocchio [83], which revolutionizes verifiable computation with efficient, succinct, and non-interactive proofs. Other well-known zk-SNARK proof systems like Groth16 and Ligero further optimize efficiency and scalability (refer to Section 5.1 for more details). However, leveraging smart contracts to verify zero-knowledge proofs may incur significant on-chain costs, prompting some schemes [19], [51], [52] to adopt lightweight cryptographic techniques (such as hash functions, accumulators, commitments, and symmetric encryption) to reduce on-chain costs.

Computing resources. Unlike outsourcing services, which delegate entire tasks to resource-abundant workers, decentralized fair trading for computing resources involves directly purchasing computing resources to execute tasks [53], [54], [55]. Essentially, $\mathcal{S}$ (the provider) offers heterogeneous and distributed idle computing resources to $\mathcal{B}$ so that $\mathcal{S}$ can obtain payments and $\mathcal{B}$ can get on-demand computing resources to accomplish computing tasks [55]. During on-chain trading, essential steps such as service publishing, service requesting, and resource matching are managed by smart contracts [53], [55]. Smart contracts work as auctioneers between $\mathcal{S}$ and $\mathcal{B}$ via the double auction mechanism [54]. However, smart contracts may struggle to reflect the status of computing resources in real time, limiting their ability to optimize resource resources trading.

Electricity resources. Similar to on-chain trading for computing resources, $\mathcal{S}$ (the power plant or individual) provides electricity resources to $\mathcal{B}$ (the electric vehicle) to obtain payments. Auction mechanism is proposed to optimize electricity pricing and the amount of traded electricity and maximize social welfare [56], [84]. A micropayment scheme combining hashchains and smart contracts reduces costs for frequent small payments [57], [58], [85]. Although these studies have successfully optimized electricity pricing and minimized the cost for frequent payments of small values, they face challenges when dealing with the high frequency of trading and substantial payment amounts for Electric Vehicles [57]. Since the inherent requirements of recording all transactions on the blockchain, on-chain trading for electricity resources has efficiency issues.

Bandwidth resources. In the on-chain trading of bandwidth resources, $\mathcal{S}$ obtains appropriate rewards from $\mathcal{B}$ for providing bandwidth resources in the peer-to-peer content delivery [59], [60], [61], [62]. It primarily focuses on mitigating the risks of unfair bandwidth usage, where $\mathcal{S}$ (referred to as the deliverer in [61]), who transmits digital commodities to the buyer, receives no reward and the bandwidth used for transmission is wasted if $\mathcal{B}$ aborts the trading. By combining the verifiable fair delivery with the smart contract, $\mathcal{S}$ can use the latest receipt to count the number of transmitted chunks and prove to the smart contract the bandwidth contribution [60], [61]. Intuitively, $\mathcal{S}$ at most wastes the bandwidth of one chunk. However, the processing capacity of blockchain and on-chain costs may become limiting factors, causing the on-chain trading of bandwidth resources to face performance bottlenecks.

Discussion. Different from digital commodities [36], [82], digital services often offer unique features and personalized experiences, requiring more dynamic and flexible pricing strategies. The pricing mechanism allows $\mathcal{S}$ to effectively monetize their resources, allocate resources efficiently, and tailor pricing models for diverse requirements. For example, the auction-based pricing mechanism [54] fosters a dynamic and efficient marketplace where prices are determined by real-time supply and demand. It can also enhance transparency and competitiveness, and encourage improving service quality as $\mathcal{S}$ strives to offer the best value to attract $\mathcal{B}$. Moreover, the second-price sealed-bid auction (also called the Vickrey auction) [86] mechanism allows the winner to pay the second-highest bid. This mechanism encourages honest bidding and reduces the incentive for bidders to manipulate prices, avoiding the risk of price inflation that might occur in the first-price sealed-bid auction mechanism.

For on-chain trading, all transactions are confirmed and stored on blockchain, which makes it almost impossible to be falsified [20]. Besides, the candidate block and corresponding transactions are verified by miners. Although on-chain trading can be utilized in various application scenarios, the confirmation for trading is slow because all transactions generated in trading are waiting to be recorded onto the blockchain. Delays in verifying and confirming transactions and the inability to support trading with small payments present significant challenges in on-chain trading. To lower transaction fees, decrease confirmation time, increase blockchain throughput, and enhance the efficiency of decentralized fair trading, numerous studies have proposed off-chain trading for diverse application scenarios.

4 Off-Chain Trading

On-chain trading requires all transactions to be recorded chronologically on the blockchain, resulting in confirmation delays. It often fails to meet scenarios that require low latency and high throughput. Moreover, on-chain trading primarily takes place on the Bitcoin and Ethereum blockchains, where transaction fees are usually not cheap. For scenarios with micropayments, such as decentralized fair trading for electricity resources, transaction fees can be prohibitively expensive, discouraging users from participating in trading [57]. Therefore, numerous studies propose off-chain trading for enhancing efficiency and mitigating the transaction fees inherent in decentralized fair trading.

Off-chain trading requires transactions to be processed by a second-layer or other chain, meaning they are not

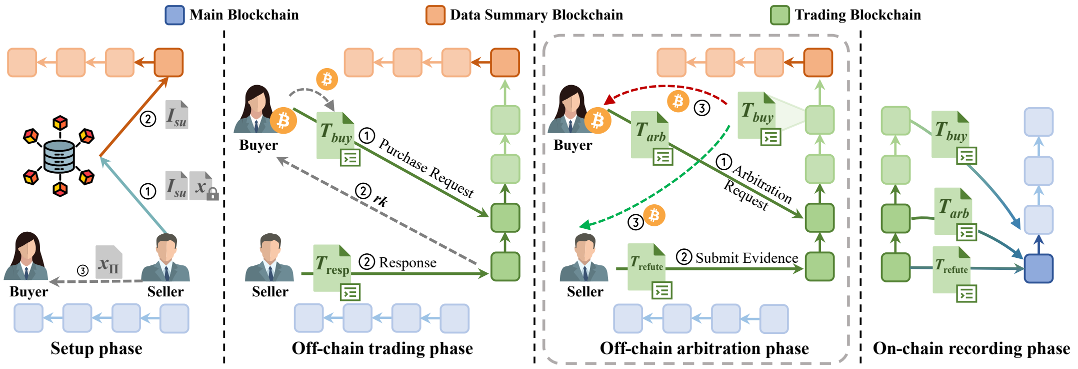


Fig. 3. The fundamental framework of off-chain trading. Each block in DSC records metadata about different digital commodities that have their own corresponding TC for trading. Similar to the on-chain trading in Fig. 2, dark blue, dark green and dark orange represent new blocks, while light blue, light green and light orange represent old blocks. The gray dashed border indicates that the off-chain arbitration phase is optional. During the on-chain recording phase, important transactions are recorded on the main blockchain which enables the main blockchain and sidechain to perform in parallel.

immediately recorded on the main blockchain. These transactions are later integrated into the main blockchain, thus reducing transaction costs and processing times. In this section, we summarize a fundamental framework of off-chain trading and explore the enhancements made for off-chain trading combined with diverse application scenarios.

4.1 Fundamental Framework of Off-Chain Trading

Off-chain significantly enhances efficiency and reduces transaction fees by processing transactions off the main blockchain, making it well-suited for decentralized fair trading with micropayments. It also enhances interoperability across multiple blockchains, enables complex smart contract interactions, and supports parallel transaction processing through payment channels, sidechains, and state channels.

The fundamental framework of off-chain trading is built on the underlying blockchain, where the main blockchain records the transaction results, while the sidechain stores the transaction contents. Drawing from existing studies [87], [88], we take the off-chain trading for digital commodities as an example and divide it into four phases, *i.e.*, setup phase, off-chain trading phase, off-chain arbitration phase, and on-chain recording phase, as shown in Fig. 3.

Setup phase: The user registers and obtains his identity public-private key pair. By hashing the identity public key, the user's account address can be obtained, which is used to receive rewards.

i) $\mathcal{B}$ and $\mathcal{S}$ obtain their identity public-private key pairs $(pk_{\mathcal{B}}, sk_{\mathcal{B}})$ and $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ after registration.

ii) $\mathcal{S}$ sends the encrypted digital commodity x and the description information I_{su} into the decentralized storage network before the trading. If a digital commodity with the same hash value $\text{Hash}(x)$ already exists on the data summary blockchain (DSC, a sidechain), x will be rejected; otherwise, I_{su} will be recorded on DSC.

iii) $\mathcal{B}$ verifies the correctness of x (refer to Section 3.1).

Off-chain trading phase: It procedurally transitions into the off-chain trading phase.

i) $\mathcal{B}$ generates transaction T_{buy} onto the trading blockchain (TC, a sidechain) by invoking the purchase function in the smart contract with requirements and p coins to initiate a formalized request for purchase.

ii) $\mathcal{S}$ responds to $\mathcal{B}$ by invoking the selling function and generating the transaction T_{resp} within the time window t_1. He encrypts k to obtain rk via $pk_{\mathcal{B}}$, which is formally denoted as $rk = \textbf{Enc}_{pk_{\mathcal{B}}}(k)$. Subsequently, he transmits the transaction T_{resp} which contains rk onto TC.

iii) After receiving T_{resp}, $\mathcal{B}$ decrypts rk via $sk_{\mathcal{B}}$ and verifies whether k satisfies the condition $\textbf{Open}(c, d, k) = 1$ within the time window t_2. Then he proceeds to obtain x by decrypting x by k and compare it with the provided description.

iv) If $\mathcal{B}$ doesn't dispute the correctness of x before the time window t_3, the p coins are transferred to $\mathcal{S}$.

(Optional) Off-chain arbitration phase: As the smart contract transitions into the off-chain arbitration phase, the p coins will remain under lock in the smart contract until miners verify the evidence from $\mathcal{B}$ and $\mathcal{S}$. It is similar to the on-chain arbitration phase (refer to Section 3.1), except that transactions are executed on TC.

On-chain recording phase: Many transactions (*e.g.*, T_{buy}, T_{sale}, T_{resp}, T_{refute}, and T_{arb}) are generated during the off-chain trading. However, to reduce the burden and avoid congestion on the main blockchain, it does not need to record information and status of all transactions in real time. The main blockchain and sidechain perform in parallel, where the main blockchain is responsible for validating, executing, and recording transactions. Moreover, only important transactions (like T_{buy}) will be recorded on the main blockchain after the trading is completed. If the off-chain arbitration phase is executed, T_{refute}, T_{arb}, and the arbitration result also will be recorded on the main blockchain.

As elucidated by the fundamental framework of off-chain trading presented above, the significant difference between off-chain trading and on-chain trading is that transactions do not need to be recorded on the main blockchain in real time. By integrating with the sidechain, the burden on the main blockchain is alleviated, scalability is improved, the execution of consensus algorithms is improved, and transaction latency and fees are significantly reduced.

TABLE 3: Comparison among off-chain trading for different application scenarios.

Scenario	Schemes	Efficiency enhancements
Digital signatures	Gao *et al.* [63]	Reduces on-chain verification costs by the certificateless aggregate verifiably signature scheme.
ML models	Golden Grain [36]	Design secure off-chain on-chain interaction protocol based on TEE.
Crowdsensing data	CrowdBC [69]	Leverage IPFS for storing task data and crowdsensing data off-chain.
	Cai *et al.* [89]	Develop a co-design of off-chain and on-chain computing to foster a secure and economical ecosystem.
	bHIT [90]	Simple commitment and efficient verification are achieved through AHCTree.
	CDT-B [91]	Leverage layered sharding blockchain based on membership degree to improve efficiency.
IoT data	Block-DM [92]	Introduce a set of supervising nodes to form an efficient consortium blockchain.
	Li *et al.* [46]	Leverage quantum particle swarm optimization to minimize the transmission delay and cost of caching.
	Dai *et al.* [93]	Design the Proof of Utility to accelerate the block verification process.
	Guo *et al.* [94]	Combine belief propagation-based caching with smart contracts to optimize dynamic cache allocation.
Cryptocurrencies	Yin *et al.* [95]	Employ efficient sidechain constructions for fast cross-chain transfers.
	CrossChannel [96]	Combine off-chain payment channels with chain relay protocol for fast and cheap cross-chain payments.
	monoCash [97]	Leverage trusted monotonic counters to build the first channel-free off-chain payment network.
Outsourcing services	Guan *et al.* [98], Dong *et al.* [99]	Combine game theory (off-chain) with smart contracts only requiring simple on-chain verification.
	Arbitrum [100]	Develop off-chain smart contracts to move storage and verification from the blockchain.
	Xiao *et al.* [101]	Leverage consortium blockchain for lower on-chain costs and faster transaction processing.
Computing resources	Yang *et al.* [102]	Leverage two-level Stackelberg game for pricing and matching off-chain.
	Xie *et al.* [103]	Leverage deep reinforcement learning for pricing off-chain.
	Paramart [104]	Leverage blockchain sharding for parallel processing transactions.
Electricity resources	DPTS [15]	Make most of the protocol steps off-chain, leaving only Bitcoin payments on-chain to improve efficiency.
	V2GEx [57]	Combine hashchain-based micropayments with smart contracts to reduce the cost of frequent payments.
	BBARS [105]	Combine the aggregate signature with the public key infrastructure to improve efficiency.
Bandwidth resources	Keizer *et al.* [106]	Leverage the Proof of Timely Relay for off-chain data verification.
	FairRelay [107]	Combine payment channels and Accumulative Hashed TimeLock Contracts to avoid on-chain costs.

4.2 Enhancements for Off-Chain Trading

Miners operate under a highest-fee-first-served policy. Consequently, participants in decentralized fair trading are compelled to offer non-trivial transaction fees as an incentive to ensure the timely processing of their transactions. It brings a significant hurdle for small-value transactions and causes inefficiency in decentralized fair trading. In order to solve the efficiency bottleneck and reduce the fees for decentralized fair trading, many studies convert on-chain trading to off-chain trading for diverse scenarios. In this section, we summarize the enhancements made by existing studies to improve trading efficiency and reduce fees. The comparison among off-chain trading for different application scenarios is shown in Table 3. Due to space limitations, we mainly discuss common scenarios, *i.e.*, crowdsensing data, IoT data, cryptocurrencies, outsourcing services, computing resources, electricity resources, and bandwidth resources.

4.2.1 Digital Commodities

Crowdsensing data. The inherent limitations of on-chain trading for crowdsensing data, such as high gas costs, latency, and limited scalability, prompt researchers to propose diverse off-chain trading designs [68], [69], [89], [90], [91], [108], [109]. These designs suggest that placing all transactions on-chain is uneconomical and it is more economical to consider the delicate joint on-chain and off-chain [69], [89]. For example, bHIT [90] avoids using expensive cryptographic tools like zero-knowledge proofs and does not send crowdsensing data to the blockchain. Instead, it uses the additively homomorphic-based commitment tree (AHCTree) for efficient commitment and verification, significantly reducing gas costs. The three-layer architecture [69], consisting of the application layer, blockchain layer, and storage layer, separates most transactions from the main blockchain, alleviating its burden and improving trading efficiency. Crowdsensing data is securely stored on cloud servers (storage layer), along with the reliability of $\mathcal{S}$ to serve as input for truth discovery. The truth knowledge is transmitted onto the blockchain, converting crowdsensing data trading into lightweight knowledge trading [68]. The inherent inefficiency of blockchain stems from the consensus mechanism requiring all miners to verify and store every transaction, with every consensus message broadcast across the entire network. To mitigate this issue, the Layered Sharding Blockchain [108], [109] groups nodes with higher transaction frequency, bolstering trading throughput [91].

IoT data. Decentralized fair trading for IoT data not only faces challenges similar to those of crowdsensing data but also requires low delivery latency and optimized data caching strategies. Researchers have implemented various strategies to tackle these challenges successfully. A consortium blockchain framework is harnessed to ensure secure storage of critical transactions, while less essential transactions are removed from the main blockchain for execution. This strategic maneuver facilitates the efficiency of decentralized fair trading for IoT data [44], [92]. In parallel, various caching strategies have been proposed to improve hit ratios and reduce delivery latency. Some of these strategies leverage quantum particle swarm optimization and belief propagation algorithms to optimize data caching and minimize delivery latency [46], [93]. These approaches can dynamically optimize caching space allocation by integrating smart contracts [94], [110]. Similar to [69], a comprehensive three-layer architecture is introduced, consisting of the physical network layer, blockchain edge layer and blockchain network layer, along with advanced caching strategies. The combination of these factors results in an elevated hit ratio, reduced data delivery latency, and enhanced the efficiency of decentralized fair trading for IoT data [94].

Cryptocurrencies. The on-chain trading of cryptocurrencies often encounters challenges such as low efficiency and high fees. Inspired by the Lightning Network, existing studies leverage the sidechain and off-chain payment channel to address these challenges [95], [96]. The two-way pegs enable cryptocurrencies to move securely between the mainchain and sidechain without intermediaries [95]. Once the cryptocurrencies are locked on the mainchain, an equivalent amount is minted or released on the sidechain. The CrossChannel [96] employs the chain relay mechanism to synchronize channel-related information across blockchains and leverages micropayment channels to enable two participants on different blockchains to establish an off-chain payment channel, facilitating fast and cheap cross-chain cryptocurrency trading. monoCash [97] leverages trusted execution environments to construct trusted monotonic counters, which ensure the order and consistency of transactions between different blockchains. Each transaction is assigned a unique and immutable counter that is synchronized across blockchains, enabling fast and secure cross-chain payments without intermediaries or payment channels.

4.2.2 Digital Services

Outsourcing services. To mitigate expensive storage and verification costs associated with executing complex smart contracts, researchers have developed some off-chain trading designs for outsourcing services [98], [99], [100], [101]. For example, game theory [98], [99] is leveraged to handle computations entirely off the blockchain, where multiple $\mathcal{S}$ are introduced to perform the same task, and smart contracts are designed to create tension, incentivize betrayal, and foster distrust between them. This strategic interaction enables smart contracts to directly verify their results without expensive cryptographic techniques, ensuring accountability and correctness. Moreover, researchers have also made improvements to the underlying blockchain infrastructure. For instance, consortium blockchain [101] achieves lower on-chain costs and faster transaction processing by operating within a permissioned network, where only authorized participants can validate transactions associated with outsourcing service trading; Arbitrum [100], the scalable smart contract, enables most of the computation to be performed off-chain, reducing on-chain costs and latency. The fraud-proof mechanism verifies the integrity and correctness of outsourcing services off-chain, requiring minimal interaction with the main blockchain only when disputes. With these improvements, decentralized fair trading for outsourcing services can achieve lower on-chain costs, greater scalability, and faster processing.

Computing resources. On-chain trading for computing resources relies on smart contracts to perform complex operations (such as resource matching), which incurs high on-chain costs and struggles to reflect real-time changes in resource status. The common scheme is to move complex operations off-chain [102], [103]. For instance, deep reinforcement learning is used to develop off-chain bidding strategies, with smart contracts matching $\mathcal{B}$ and $\mathcal{S}$ based on the bids [103]; the Stackelberg game-based computing resource trading mechanism facilitates off-chain pricing and matching, and smart contracts prevent illegal behaviors like delayed payment [102]. Additionally, blockchain sharding can accelerate transaction processing in computing resource trading by dividing computing resources into multiple shards and processing transactions in parallel across these shards [104]. These schemes can reduce on-chain costs and better adapt to real-time computing resource changes.

Electricity resources. Efficiency challenges encountered in electricity resources are similar to those in computing resources [57]. Given the high frequency of small-value transactions in electricity trading, on-chain trading may encounter Bitcoin Dust [111]. Similar to off-chain trading for digital services mentioned above [98], [101], off-chain trading for electricity resources follows a similar paradigm to reduce transaction confirmation latency on the main blockchain and improve trading efficiency. The main blockchain is solely responsible for the transfer of payments, and the majority of transactions take place off-chain [15], [57], [105]. Furthermore, payments can also be transferred off-chain through the utilization of the payment channel combined with the hashchain micropayment [112], thereby further improving the efficiency of decentralized fair trading for electricity resources [57]. Moreover, high throughput trading of electricity resources can be realized through the amalgamation of high throughput consensus algorithms and Sharding Blockchain [108], [109].

Bandwidth resources. To address performance bottlenecks in on-chain trading of bandwidh resources, payment channel networks offer a feasible approach [106], [107], where most transactions occur within the payment channel, significantly reducing the frequency of interactions with the blockchain. Moreover, the on-chain costs for verifying the bandwidth resources are constant, unaffecting by the content chunk size, the number of content chunks, or the number of participants. Notably, if the content delivery concludes without any disputes, no on-chain costs are incurred.

In the realm of off-chain trading, the predominant approach is to transfer most transactions from the main blockchain to off-chain execution. This shift mitigates the risk of transaction congestion on the main blockchain, which can cause inefficiency in decentralized fair trading. Furthermore, off-chain trading provides additional benefits, including the reduction of transaction fees and the alleviation of bitcoin dust in on-chain trading, especially for small-value transactions. However, significant concerns remain regarding the privacy and security in decentralized fair trading for some application scenarios [57], [68], [113].

5 Privacy and Security

Existing studies on decentralized fair trading not only endeavor to improve trading efficiency but also delve into addressing privacy and security issues. Preserving user anonymity and data confidentiality is crucial, leading researchers to develop privacy protection mechanisms to meet these fundamental requirements [114], [115], [116]. Furthermore, vulnerabilities within blockchain networks like Bitcoin and Ethereum have been identified, which can be exploited by attackers for illegal profits [111], [117], [118]. These vulnerabilities may lead to a range of potential attacks, prompting significant efforts to enhance the privacy and security of decentralized fair trading. In this section,

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2025.3547143

we analyze the enhancements for privacy and security in decentralized fair trading combined with specific scenarios and summarize the protection against common attacks.

5.1 Privacy Protection

When employing blockchain for decentralized fair trading, important transactions are immutably recorded on the blockchain, which is inherently transparent. Consequently, some privacy challenges emerge, such as ❶ **Data confidentiality:** ensuring that the data of digital commodities or digital services is not accessed by nonauthorized users; ❷ **User anonymity:** safeguarding the user's private information and unlinking it from transactions; ❸ **Transaction privacy:** guaranteeing that the contents of transactions are not accessed by nonauthorized users.

Since the emergence of privacy challenges raising concerns for decentralized fair trading, researchers have proffered many solutions, integrating diverse privacy-preserving computation techniques to guarantee privacy for diverse application scenarios, as shown in Fig. 4. We primarily analyze and discuss the following privacy-preserving computation techniques: (i) secret sharing, (ii) garbled circuits, (iii) zero-knowledge proofs, (iv) trusted execution environments, and (v) homomorphic encryption. We compare different privacy-preserving computation techniques and list their advantages and disadvantages in Table 4.

Secret sharing and garbled circuits are two key techniques for secure multi-party computation (MPC), enabling participants to collaboratively compute results without revealing their private inputs. MPC based on secret sharing includes protocols like BGW [119] and SPDZ [120], where BGW [119] uses polynomial secret sharing for secure computations in dishonest-majority settings and SPDZ [120] uses additive secret sharing with multiplicative preprocessing to reduce the computation cost during the online phase. MPC based on garbled circuits includes Yao's pioneering garbled circuit protocol [24] for Millionaire's Problem, and subsequent protocols like GMW [121] and TinyGarble [122], where GMW builds [121] on garbled circuits for Boolean function evaluation in multi-party settings and TinyGarble [122] optimizes this process for efficiency, particularly in resource-constrained environments. In decentralized fair trading, secret sharing is well-suited for ensuring fairness in simple scenarios like digital signatures and IoT data [92], [123], where privacy and security are prioritized; garbled circuits are more efficient for complex scenarios like ML models [124], where complex computations and Boolean logic are involved, and they can further encapsulate, store, and process digital commodities verifiably while preserving privacy [125]. In general, secret sharing provides robust security but is less efficient for computation-heavy scenarios, while garbled circuits excel in such scenarios but require more resources for communication and computation.

Zero-knowledge proofs verify information authenticity without revealing any private information [73], [129]. Among these, the zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK) [83] is widely adopted for its efficiency and practicality. The development of zk-SNARK began with Pinocchio [83], introducing succinct, non-interactive proofs and efficient verification but relying

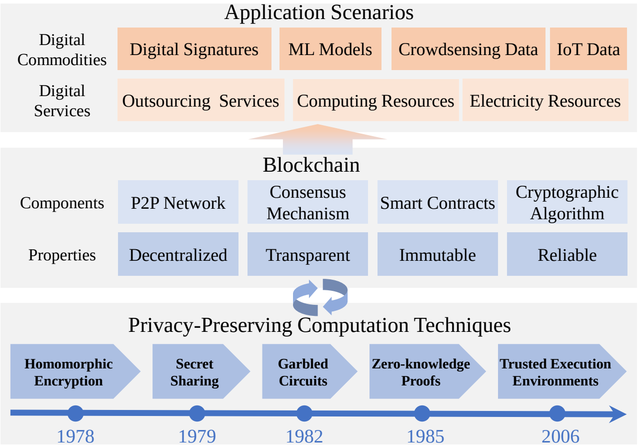


Fig. 4. Privacy-preserving computation techniques combined with blockchain.

on a trusted setup. Groth16 [130] improves upon this with smaller proof sizes and faster verification, becoming a standard for privacy-preserving systems like Zcash [131]. Subsequent advancements further addressed issues like trusted setups and scalability. Ligero [132] uses a transparent, recursive proof construction based on the Fiat-Shamir heuristic, generating efficient zero-knowledge proofs without relying on any initial trusted parameters. Bulletproofs [133] focuses on efficient range proofs without trusted setups. Marlin [134] offers universal and transparent setups, efficient verification, and adaptability to arbitrary circuits. These techniques enable applications like Zero-Knowledge Contingent Payment (ZKCP) [30], [31], verifiable computation [50], and digital commodity traceability [113]. Although zero-knowledge proofs enhance privacy and fairness in decentralized trading, they may face challenges like trusted setup reliance and expensive computation costs.

Homomorphic encryption allows arbitrary computations to be performed over encrypted data, eliminating the risk of exposing the data [78], [135]. Therefore, homomorphic encryption holds substantial promise in decentralized fair trading for crowdsensing data, where $\mathcal{S}$ is required to report the truthful crowdsensing data and $\mathcal{B}$ is forced to rate the reliability of $\mathcal{S}$ based on the data. Homomorphic encryption is utilized to direct computation over encrypted data and design privacy-preserving truth discovery algorithms combined with smart contracts [39], [127]. However, it is imperative to exercise caution when employing zero-knowledge proofs and homomorphic encryption due to the potential for significant computational overhead [16], [89].

Trusted execution environments (TEEs) are hardware-supported privacy techniques with dedicated modules that enhance system security and privacy. Unlike other privacy-preserving computation techniques, TEEs rely on the CPU to ensure the privacy and integrity of code and data, enabling efficient decentralized fair trading with lower computation overhead [74]. For scenarios where $\mathcal{B}$ only requires analysis results for data-driven decision-making, TEEs facilitate data hosting/exchange-as-a-service. Intel Software Guard Extensions (SGX) [136], a prominent TEE implementation, safeguards data processing, source data, and analysis results [36], [75], [126]. SGX enclaves decrypt data securely,

TABLE 4: Comparison among privacy-preserving computation techniques.

Techniques	Applications	Advantages	Disadvantages
Secret Sharing	Block-DM [92] CMFE [123]	Efficient for arithmetic-heavy scenarios Scalable for large numbers of participants	Multiple rounds of communications Inefficient for complex logical computations
Garbled Circuits	NN-EMD[124] PrivData [125]	Efficient for complex logical computations Constant and relatively low round complexity	Require constructing and encoding garbled circuits Heavy communication costs & large circuit sizes for complex circuits
Zero-knowledge Proofs	ZKCP [30] ZKCPlus [31] ZKDET [113]	Proof can be verified without revealing the secret Non-interactive variants simplify real-world applications	Proof generation and verification can be resource-heavy Difficult to scale for large datasets or complex statements
Trusted Execution Environments	SGPs [75] SDTE [126] Goldem Grain[36]	Ensure data and code integrity within the enclave Safeguard critical operations from system-wide threats	Susceptible to attacks that bypass hardware security Constrained in terms of memory and processing power
Homomorphic Encryption	EPOC [127] BCDT [39] Liu *et al.* [128]	Single-round secure interaction Low communication costs in encrypted computation	Only support certain types of mathematical operations Difficult to scale for large datasets due to performance overhead

preventing (malicious) $\mathcal{B}$ from accessing raw or intermediate data or analysis results. Therefore, SGX enclaves support private data analysis from $\mathcal{S}$ while ensuring confidentiality by providing full-fledged and customized leakage protection [75], [126], [137]. However, researchers must be aware of potential side-channel attacks in Intel SGX and similar TEE products to mitigate privacy risks [138], [139].

5.2 Security Enhancements

5.2.1 *Threat Model*

Decentralized fair trading mainly involves three entities: seller $\mathcal{S}$, buyer $\mathcal{B}$, and miner $\mathcal{M}$. These entities may act maliciously to disrupt fair trading for financial benefits. A malicious $\mathcal{S}$ may attempt to manipulate or misreport the details of digital commodities/services to deceive $\mathcal{B}$ and gain financial benefits [16], [41]. $\mathcal{S}$ may also collude with others or create multiple fake identities to inflate prices for financial benefits. A malicious $\mathcal{B}$ may initiate chargebacks or cancel payments after receiving digital commodities/services [140], [141]. In some cases, they even exploit vulnerabilities in the blockchain to avoid paying for commodities/services received. As validators of the blockchain, $\mathcal{M}$ holds significant power. A malicious $\mathcal{M}$ may collude with $\mathcal{B}$ or $\mathcal{S}$ to manipulate transaction ordering [99], [142]. They may also exploit their control over the consensus process to rewrite transaction history or reverse confirmed transactions to benefit a particular entity. Additionally, $\mathcal{M}$ can manipulate transaction fees or gas prices to prioritize certain trades or delay others for financial benefits.

5.2.2 *Common Attacks and Defense Strategies*

51% attack. The 51% attack occurs when a malicious miner or group of miners gains control over more than 50% of the computation resources in a blockchain network, particularly in Proof-of-Work blockchains. In this case, the attackers can perform disruptive actions like inserting fraudulent transactions, halting transaction confirmations, and rewriting transaction history [111]. To mitigate the 51% attack, existing studies propose strategies such as managing permissioned miners, implementing hybrid consensus mechanisms, and introducing finality safeguards, to limit the influence of malicious miners and make the cost much higher than the potential benefit [111], [113], [143].

Double-spending attack. The double-spending attack occurs when a malicious $\mathcal{B}$ attempts to spend the same cryptocurrency more than once by submitting conflicting transactions[140]. To resist the attack, a nonce (*i.e.*, a transaction counter in each account) or a unique serial number is set in each transaction [144]. Moreover, the security of blockchain systems is underpinned by the consensus algorithm and the computation resources, which means the malicious $\mathcal{B}$ need to exert control over more than 51 percent of the entire network's computation resources to conduct the double-spending attack [140], [141], [143].

Denial of Service (DoS) attack. The DoS attack is usually caused by excessive requests from malicious $\mathcal{B}$ or $\mathcal{S}$. Since some researchers recognize the inherent difficulty in completely avoiding DoS, they seek to increase the cost of performing DoS attacks. For example, requiring participants to pay fees for miners who maintain the blockchain, compensating participants being attacked [16], [36], [69], [126], and using smart contracts to terminate a user's authority when malicious frequent trading is detected [141], [143].

Sybil attack. The Sybil attack occurs when a malicious $\mathcal{B}$ or $\mathcal{S}$ creates multiple fake identities to gain unfair benefits. To counter this, leader election protocols like Proof-of-Work and Proof-of-Stake can be employed to prevent malicious actors from gaining control through fake identities [118]. Moreover, many strategies necessitate participants to post a deposit, which urges participants to complete the entire process as required and punishes malicious participants who launch the Sybil attack [69].

Side-channel attack. The TEE is utilized to protect the privacy of raw data, intermediate data, and analytical results in decentralized fair trading [36], [126]. However, current TEE implementations are vulnerable to side-channel attacks, which exploit information such as cache timing and power consumption to extract sensitive data from within the enclave [36], [138]. To counter side-channel attacks, techniques such as oblivious RAM, memory access obfuscation, and cache partitioning can be deployed to obfuscate memory traces and prevent the leakage of sensitive information [145], [138], [139], [146]. Although considerable efforts have been made, they still fail to address physical access-based attacks and lack comprehensive protections [146]. Consequently, researchers should remain aware of the potential side-channel attack when employing TEE products like Intel SGX and be diligent in seeking appropriate solutions to avoid issues such as privacy leakage [138], [139].

Wormhole attack. The wormhole attack occurs when several malicious participants collude to steal payment fees from others along one path of the payment channel network [147]. This attack compromises the integrity of the

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2025.3547143

payment channel and increases the risk of double-spending. To mitigate the wormhole attack, existing studies employ techniques like Hash Time-Lock Contracts and commitment schemes to ensure transaction validation and prevent unauthorized control over the payment channel [147], [148].

Sandwich attack. The sandwich attack occurs when an attacker places two transactions around a victim's transaction in an atomic exchange to manipulate the price. The attacker first places a transaction just before the victim's transaction, then another transaction right after, profiting from the price slippage caused by the victim's transaction [142]. To mitigate sandwich attacks, data-independent ordering, content-oblivious ordering, and front-running protection mechanisms [142], [149] are proposed. These mechanisms prevent attackers from predicting, manipulating, or exploiting transaction sequences based on their content or timing.

Free-riding and false-reporting attack. Both attacks occur when malicious $\mathcal{B}$ or $\mathcal{S}$ benefit without fair contributions [16], especially in decentralized fair trading for crowdsensing data. For example, $\mathcal{S}$ obtains rewards without providing correct digital commodities/services (called free-riding); $\mathcal{B}$ claims correct commodities are low-quality to avoid payment (called false-reporting). Existing studies typically leverage smart contracts to force participants to comply with agreed terms, along with reputation mechanisms to score their behavior and incentive mechanisms to reward honest participants [16], [41].

HashSplit attack. Due to the asynchronous nature of blockchain networks, the HashSplit attack can be launched by forking the blockchain or delaying block propagation [150]. Attackers manipulate block propagation time and order, causing miners to maintain conflicting views of the blockchain. To mitigate the attack, existing studies propose strategies, such as implementing lock-step synchronous networks, applying fork resolution mechanisms, and forming a complete graph topology [150]. Additionally, HashSplit attacks can be mitigated by converting public blockchains to consortium blockchains and integrating fully asynchronous protocols [151], which optimize block propagation and synchronization, making it more difficult for attackers to withhold blocks for forking the blockchain.

Partitioning attack. The partitioning attack occurs when the network is intentionally or unintentionally partitioned, preventing miners in different partitions from synchronizing blocks and transactions, which results in inconsistent blockchain views [152]. Attackers exploit network isolation or partitioning to achieve malicious purposes, such as disrupting consensus and causing blockchain forks. Mitigation measures include monitoring additional statistics, requesting blocks from multiple connections, and so on [152]. Moreover, fully asynchronous protocols, like asynchronous consensus [151] and asynchronous MPC [153], enable secure and fault-tolerant computations without relying on synchronized communication, providing strong resilience against attacks caused by network asynchrony and partitioning.

Routing attack. The routing attack exploits vulnerabilities in routing protocols or manipulates Bitcoin traffic to delay block and transaction propagation, and isolate miners [152]. These attacks can partition the network, increase fork rates, and enable double-spending by disrupting block propagation. Mitigation strategies include deploying secure routing protocols, using robust consensus algorithms (like HoneyBadgerBFT) [151], [152], and applying asynchronous MPC [153] to ensure computations without global synchronization despite miners encountering network partitions or block propagation delays.

6 Potential Future Research

Despite extensive research into decentralized fair trading, the efficiency of trading is still not satisfactory. In this section, we discuss some remaining challenges and propose potential future research for decentralized fair trading.

Efficient Decentralized Fair Trading. A promising research direction is minimizing conflicting transactions when concurrent blocks are added to the blockchain. While decentralized fair trading focuses on throughput and efficiency, it often overlooks the impact of conflicting transactions on the blockchain. Conflicting transactions may hinder the effective throughput, ultimately diminishing the efficiency of decentralized fair trading. One approach to mitigate this is to allocate transactions strategically, possibly prioritizing based on trading information and miner identities, to reduce conflicts between concurrent blocks and improve trading efficiency. Another potential research direction is to improve the efficiency of decentralized fair trading through Sharding Blockchain [108], [109]. In non-sharding systems, all miners must verify and store every transaction, while Sharding Blockchain strategically partitions transactions and miners to balance the computation load. However, this approach may have limitations in decentralized fair trading for crowdsensing data due to the limitations of workers's devices and the tasks completed by each worker are often similar [91]. Future research could focus on integrating Sharding Blockchain with decentralized fair trading to improve throughput, efficiency, and cross-shard transaction processing in various applications.

Cross-chain Interoperability. Blockchain promises to be the interconnected network of the future, where individuals can engage in decentralized fair trading without being bound by the limitations imposed by centralized platforms. However, relying on a single public blockchain is insufficient to fully realize its potential. The scalability issues may extend transaction confirmation time, which is detrimental to application scenarios with stringent latency requirements. Transactions waiting for verification by miners may suffer significant delays, which are fatal for latency-sensitive application scenarios. Beyond improving the efficiency of a single blockchain, significant improvements in trading efficiency can be achieved through the collaborative interaction of multiple blockchains. One potential research direction is the sidechain. The sidechain allows multiple blockchains to communicate with each other through the two-way peg, facilitating parallel processing of various transactions. As a result, the sidechain can assume responsibility for the processing of transactions generated in decentralized fair trading and establish bidirectional communication between the main blockchain and the sidechain, effectively reducing the load on the main blockchain. Existing studies focus on employing sidechains for storing summary information and evaluation information of data to enhance retrieval and validation efficiency in decentralized fair trading [87], [88].

However, future research is needed to optimize the transmission of critical transactions and achieve rapid consensus between the main blockchain and the sidechain.

Potential Application Scenarios. Decentralized fair trading has significantly advanced through extensive research, applying in various application scenarios. However, it is still an unresolved challenge that prove to the buyer that the digital commodity meets his expectations without disclosing details of the commodity. While zero-knowledge proofs can confirm ownership, they cannot verify quality. To address this, reputation mechanisms are used to log the evaluation of commodities, discouraging low-quality trades and incentivizing high-quality ones [39], [44], [88]. Future research could focus on evaluating digital commodities quality and aligning digital services with buyer expectations using privacy-preserving techniques like TEEs and zero-knowledge proofs. One potential research direction is further exploring application scenarios that combine with decentralized fair trading. In time-sensitive scenarios, such as IoT data and computing resources, low latency is crucial, one possible approach is to execute smart contracts on edge devices that support trusted hardware, reducing transmission latency. In decentralized storage [154], [155], [156], [157], researchers could focus on designing fine-grained pricing based on the latency, bandwidth, and security of storage service rather than solely storage capacity, and developing standards/protocols for interoperability, enabling data to be stored and retrieved seamlessly across multiple networks like IPFS, Storj, and Filecoin. Decentralized fair trading can also address geographic disparities and resource scarcity in computing resources [158], establishing decentralized trust for computing resources trading via the blockchain [84]. Moreover, research into unified pricing strategies for heterogeneous computing resources (*e.g.*, CPU, GPU) is another promising avenue for future exploration.

7 Conclusion

We present a comprehensive survey on decentralized fair trading, covering the fundamental framework for on-chain and off-chain trading, efficiency enhancements for various application scenarios, and defense strategies against common attacks. We outline the challenges and potential future research directions, aiming to inspire researchers and practitioners to explore and improve the efficiency, privacy, and security of decentralized fair trading across diverse application scenarios. Our primary objective is to present and summarize a collection of seminal studies for the benefit of researchers interested in decentralized fair trading.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (No. 2021YFB2900100), and in part by the National Natural Science Foundation of China (No. U22B2022, 62372383).

References

[1] "History of the Internet - Wikipedia," https://en.wikipedia.org/w/index.php?title=History_of_the_Internet, [Accessed 07-11-2023].

[2] "X - Enterprise APIs," https://developer.x.com/en/products/x-api/enterprise, [Accessed 07-11-2023].

[3] "Location Technology Unlocking Powerful Connections - FSQ," https://foursquare.com/, [Accessed 07-11-2023].

[4] "Terbine – Enabling the EV/AV Ecosystem," https://terbine.com/, [Accessed 07-11-2023].

[5] N. Asokan, M. Schunter, and M. Waidner, "Optimistic protocols for fair exchange," in *Proc. of ACM CCS*, 1997.

[6] F. Bao, R. H. Deng, and W. Mao, "Efficient and practical fair exchange protocols with off-line TTP," in *Proc. of IEEE S&P*, 1998.

[7] G. Ateniese, "Efficient verifiable encryption (and fair exchange) of digital signatures," in *Proc. of ACM CCS*, 1999.

[8] N. Asokan, V. Shoup, and M. Waidner, "Optimistic fair exchange of digital signatures," *IEEE JSAC*, vol. 18, no. 4, pp. 593–610, 2000.

[9] N. Asokan, V. Shoup, and M. Waidner, "Asynchronous protocols for optimistic fair exchange," in *Proc. of IEEE S&P*, 1998.

[10] A. Küpçü and A. Lysyanskaya, "Usable optimistic fair exchange," *Computer Networks*, vol. 56, no. 1, pp. 50–63, 2012.

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

[12] S. Goldfeder, J. Bonneau, R. Gennaro, and A. Narayanan, "Escrow protocols for cryptocurrencies: How to buy physical goods using bitcoin," in *Proc. of FC*, 2017.

[13] I. Bentov and R. Kumaresan, "How to use bitcoin to design fair protocols," in *Proc. of CRYPTO*, 2014.

[14] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek, "Secure multiparty computations on bitcoin," *Communications of the ACM*, vol. 59, no. 4, pp. 76–84, 2016.

[15] T. Dimitriou and A. Mohammed, "Fair and privacy-respecting bitcoin payments for smart grid data," *IEEE IOT*, vol. 7, no. 10, pp. 10 401–10 417, 2020.

[16] S. Dziembowski, L. Eckey, and S. Faust, "Fairswap: How to fairly exchange digital goods," in *Proc. of ACM CCS*, 2018.

[17] L. Eckey, S. Faust, and B. Schlosser, "Optiswap: Fast optimistic fair exchange," in *Proc. of ACM Asia CCS*, 2020.

[18] Y. Zhang, R. H. Deng, X. Liu, and D. Zheng, "Outsourcing service fair payment based on blockchain and its applications in cloud computing," *IEEE TSC*, vol. 14, no. 4, pp. 1152–1166, 2018.

[19] C. Lin, D. He, X. Huang, and K.-K. R. Choo, "OBFP: Optimized blockchain-based fair payment for outsourcing computations in cloud computing," *IEEE TIFS*, vol. 16, pp. 3241–3253, 2021.

[20] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-enabled federated learning: A survey," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–35, 2022.

[21] "Solidity programming language," https://solidity.readthedocs.io/en/develop/, [Accessed 07-11-2023].

[22] L. Breidenbach, C. Cachin, B. Chan, A. Coventry, S. Ellis, A. Juels, F. Koushanfar, A. Miller, B. Magauran, D. Moroz *et al.*, "Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks," *Chainlink Labs*, 2021.

[23] H. Pagnia, F. C. Gärtner *et al.*, "On the impossibility of fair exchange without a trusted third party," Citeseer, Tech. Rep., 1999.

[24] A. C.-C. Yao, "How to generate and exchange secrets," in *Proc. of IEEE FOCS*, 1986.

[25] C. Chenli, W. Tang, and T. Jung, "Fairtrade: Efficient atomic exchange-based fair exchange protocol for digital data trading," in *Proc. of IEEE Blockchain*, 2021.

[26] Y. Li, X. Feng, J. Xie, H. Feng, Z. Guan, and Q. Wu, "A decentralized and secure blockchain platform for open fair data trading," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 7, p. e5578, 2020.

[27] S. Delgado-Segura, C. Pérez-Solà, G. Navarro-Arribas, and J. Herrera-Joancomartí, "A fair protocol for data trading based on bitcoin transactions," *FGCS*, vol. 107, pp. 832–840, 2020.

[28] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek, "Fair two-party computations via bitcoin deposits," in *Proc. of FC*, 2014.

[29] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.

[30] "Zero knowledge contingent payment," https://en.bitcoin.it/wiki/ZeroKnowledgeContingentPayment, [Accessed 07-11-2023].

[31] Y. Li, C. Ye, Y. Hu, I. Morpheus, Y. Guo, C. Zhang, Y. Zhang, Z. Sun, Y. Lu, and H. Wang, "ZKCPlus: Optimized fair-exchange protocol supporting practical and flexible data exchange," in *Proc. of ACM CCS*, 2021.

[32] L. Zhang, H. Zhang, J. Yu, and H. Xian, "Blockchain-based two-party fair contract signing scheme," *Information Sciences*, vol. 535, pp. 142–155, 2020.

[33] H. K. Alper and A. Küpçü, "Optimally efficient multi-party fair exchange and fair secure multi-party computation," *ACM Transactions on Privacy and Security*, vol. 25, no. 1, pp. 1–34, 2021.

[34] Z. Wan, R. H. Deng, and D. Lee, "Electronic contract signing without using trusted third party," in *Proc. of NSS*, 2015.

[35] H. Tian, J. He, and L. Fu, "Contract coin: Toward practical contract signing on blockchain," in *Proc. of Spring ISPEC*, 2017.

[36] J. Weng, J. Weng, C. Cai, H. Huang, and C. Wang, "Golden Grain: Building a Secure and Decentralized Model Marketplace for MLaaS," *IEEE TDSC*, vol. 19, no. 5, pp. 3149–3167, 2022.

[37] H. Xu, P. Nanda, J. Liang, and X. He, "FCH, an incentive framework for data-owner dominated federated learning," *JISA*, vol. 76, p. 103521, 2023.

[38] Z. Zhou, X. Cao, J. Liu, B. Zhang, and K. Ren, "Zero knowledge contingent payments for trained neural networks," in *Proc. of ESORICS*, 2021.

[39] B. An, M. Xiao, A. Liu, Y. Xu, X. Zhang, and Q. Li, "Secure crowdsensed data trading based on blockchain," *IEEE TMC*, vol. 22, no. 3, pp. 1763–1778, 2021.

[40] J. An, Z. Wang, X. He, X. Gui, J. Cheng, and R. Gui, "PPQC: A blockchain-based privacy-preserving quality control mechanism in crowdsensing applications," *IEEE/ACM TON*, vol. 30, no. 3, pp. 1352–1367, 2022.

[41] Y. Lu, Q. Tang, and G. Wang, "Zebralancer: Private and anonymous crowdsourcing system atop open blockchain," in *Proc. of IEEE ICDCS*, 2018.

[42] Y. Lu, Q. Tang, and G. Wang, "Dragoon: Private decentralized hits made practical," in *Proc. of IEEE ICDCS*, 2020.

[43] Y. Zhao, Y. Yu, Y. Li, G. Han, and X. Du, "Machine learning based privacy-preserving fair data trading in big data market," *Information Sciences*, vol. 478, pp. 449–460, 2019.

[44] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE IOT*, vol. 6, no. 3, pp. 4660–4670, 2018.

[45] C. Lin, D. He, S. Zeadally, X. Huang, and Z. Liu, "Blockchain-based data sharing system for sensing-as-a-service in smart cities," *ACM TOIT*, vol. 21, no. 2, pp. 1–21, 2021.

[46] C. Li, S. Liang, J. Zhang, Q.-e. Wang, and Y. Luo, "Blockchain-based data trading in edge-cloud computing environment," *Information Processing & Management*, vol. 59, no. 1, p. 102786, 2022.

[47] M. Herlihy, "Atomic cross-chain swaps," in *Proc. of ACM PODC*, 2018.

[48] H. Tian, K. Xue, X. Luo, S. Li, J. Xu, J. Liu, J. Zhao, and D. S. L. Wei, "Enabling cross-chain transactions: A decentralized cryptocurrency exchange protocol," *IEEE TIFS*, vol. 16, pp. 3928–3941, 2021.

[49] H. Xie and Z. Yan, "Spcex: Secure and privacy-preserving cryptocurrency exchange," *IEEE TDSC*, vol. 21, no. 5, pp. 4404–4417, 2024.

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

[51] H. Cui, Z. Wan, X. Wei, S. Nepal, and X. Yi, "Pay as you decrypt: Decryption outsourcing for functional encryption using blockchain," *IEEE TIFS*, vol. 15, pp. 3227–3238, 2020.

[52] S. Avizheh, M. Nabi, and R. Safavi-Naini, "Refereed delegation of computation using smart contracts," *IEEE TDSC*, vol. 21, no. 6, pp. 5208–5227, 2024.

[53] W. Sun, J. Liu, Y. Yue, and P. Wang, "Joint resource allocation and incentive design for blockchain-based mobile edge computing," *IEEE TWC*, vol. 19, no. 9, pp. 6050–6064, 2020.

[54] Y. Du, Z. Wang, J. Li, L. Shi, D. N. K. Jayakody, Q. Chen, W. Chen, and Z. Han, "Blockchain-aided edge computing market: Smart contract and consensus mechanisms," *IEEE TMC*, vol. 22, no. 6, pp. 3193–3208, 2023.

[55] M. Dai, S. Guo, S. Guo, S. Shao, and X. Qiu, "Trusted sharing of computing power resources: Benefit-driven heterogeneous network service provision mechanism," *IEEE TSC*, vol. 17, no. 3, pp. 1265–1278, 2024.

[56] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE TII*, vol. 13, no. 6, pp. 3154–3164, 2017.

[57] Z. Wan, T. Zhang, W. Liu, M. Wang, and L. Zhu, "Decentralized privacy-preserving fair exchange scheme for V2G based on blockchain," *IEEE TDSC*, vol. 19, no. 4, pp. 2442–2456, 2021.

[58] Z. Wan, W.-T. Zhu, and G. Wang, "Prac: Efficient privacy protection for vehicle-to-grid communications in the smart grid," *Computers & security*, vol. 62, pp. 246–256, 2016.

[59] P. Goyal, R. Netravali, M. Alizadeh, and H. Balakrishnan, "Secure incentivization for decentralized content delivery," in *Proc. of 2nd USENIX Workshop on HotEdge*, 2019.

[60] S. He, Y. Lu, Q. Tang, G. Wang, and C. Q. Wu, "Fair peer-to-peer content delivery via blockchain," in *Proc. of Springer ESORICS*, 2021.

[61] S. He, Y. Lu, Q. Tang, G. Wang, and C. Q. Wu, "Blockchain-based p2p content delivery with monetary incentivization and fairness guarantee," *IEEE TPDS*, vol. 34, no. 2, pp. 746–765, 2023.

[62] V. H. Lakhani, L. Jehl, R. Hendriksen, and V. Estrada-Galiñanes, "Fair incentivization of bandwidth sharing in decentralized storage networks," in *Proc. of IEEE ICDCS Workshops*, 2022.

[63] Y. Gao and J. WU, "Efficient multi-party fair contract signing protocol based on blockchains," *Journal of Cryptologic Research*, vol. 5, no. 5, pp. 556–567, 2018.

[64] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[65] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Proc. of EUROCRYPT*, 1986.

[66] G. Fuchsbauer, "Commuting signatures and verifiable encryption," in *Proc. of EUROCRYPT*, 2011, pp. 224–245.

[67] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE TDSC*, vol. 18, no. 5, pp. 2438–2455, 2019.

[68] C. Cai, Y. Zheng, and C. Wang, "Leveraging crowdsensed data streams to discover and sell knowledge: A secure and efficient realization," in *Proc. of IEEE ICDCS*, 2018.

[69] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, and R. H. Deng, "Crowdbc: A blockchain-based decentralized framework for crowdsourcing," *IEEE TPDS*, vol. 30, no. 6, pp. 1251–1266, 2018.

[70] L. Zhang, Y. Li, X. Xiao, X.-Y. Li, J. Wang, A. Zhou, and Q. Li, "Crowdbuy: Privacy-friendly image dataset purchasing via crowdsourcing," in *Proc. of IEEE INFOCOM*, 2018.

[71] C. Huang, J. Wang, H. Chen, S. Si, Z. Huang, and J. Xiao, "ZkMLaaS: a Verifiable Scheme for Machine Learning as a Service," in *Proc. of IEEE GLOBECOM*, 2022.

[72] L. Zhao, Q. Wang, C. Wang, Q. Li, C. Shen, and B. Feng, "Veriml: Enabling integrity assurances and fair payments for machine learning as a service," *IEEE TPDS*, vol. 32, no. 10, pp. 2524–2540, 2021.

[73] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Providing Sound Foundations for Cryptography*, 2019, pp. 329–349.

[74] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: what it is, and what it is not," in *IEEE Trustcom/BigDataSE/Ispa*, vol. 1, 2015, pp. 57–64.

[75] F. Tramer, F. Zhang, H. Lin, J.-P. Hubaux, A. Juels, and E. Shi, "Sealed-glass proofs: Using transparent enclaves to prove and sell knowledge," in *Proc. of IEEE EuroS&P*, 2017.

[76] C. Cai, Y. Zheng, A. Zhou, and C. Wang, "Building a secure knowledge marketplace over crowdsensed data streams," *IEEE TDSC*, vol. 18, no. 6, pp. 2601–2616, 2019.

[77] S. He, D.-H. Shin, J. Zhang, J. Chen, and P. Lin, "An exchange market approach to mobile crowdsensing: pricing, task allocation, and walrasian equilibrium," *IEEE JSAC*, vol. 35, no. 4, pp. 921–934, 2017.

[78] M. Xiao, K. Ma, A. Liu, H. Zhao, Z. Li, K. Zheng, and X. Zhou, "Sra: Secure reverse auction for task assignment in spatial crowdsourcing," *IEEE TKDE*, vol. 32, no. 4, pp. 782–796, 2019.

[79] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," in *Proc. of ACM SIGKDD*, 2007.

[80] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proc. of ACM SIGKDD*, 2014.

[81] Y. Li, L. Li, Y. Zhao, N. Guizani, Y. Yu, and X. Du, "Toward decentralized fair data trading based on blockchain," *IEEE Network*, vol. 35, no. 1, pp. 304–310, 2020.

[82] C. Ying, H. Jin, J. Li, X. Si, and Y. Luo, "Incentive mechanism design via smart contract in blockchain-based edge-assisted crowdsensing," *Frontiers of Computer Science*, vol. 19, no. 3, p. 193802, 2025.

[83] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," in *Proc. of IEEE S&P*, 2013.

[84] Z. Li, Z. Yang, and S. Xie, "Computing resource trading for edge-cloud-assisted internet of things," *IEEE TII*, vol. 15, no. 6, pp. 3661–3669, 2019.

[85] R. L. Rivest and A. Shamir, "Payword and MicroMint: Two simple micropayment schemes," in *International workshop on security protocols*, 1996.

[86] S. Bag, F. Hao, S. F. Shahandashti, and I. G. Ray, "Seal: Sealed-bid auction without auctioneers," *IEEE TIFS*, vol. 15, pp. 2042–2052, 2020.

[87] S. Zheng, L. Pan, D. Hu, M. Li, and Y. Fan, "A blockchain-based trading platform for big data," in *Proc. of IEEE INFOCOM Workshops*, 2020.

[88] D. Hu, Y. Li, L. Pan, M. Li, and S. Zheng, "A blockchain-based trading system for big data," *Computer Networks*, vol. 191, p. 107994, 2021.

[89] C. Cai, Y. Zheng, Y. Du, Z. Qin, and C. Wang, "Towards private, robust, and verifiable crowdsensing systems via public blockchains," *IEEE TDSC*, vol. 18, no. 4, pp. 1893–1907, 2019.

[90] Y. Liang, Y. Li, and B.-S. Shin, "Decentralized crowdsourcing for human intelligence tasks with efficient on-chain cost," *Proc. of VLDB Endowment*, vol. 15, no. 9, p. 1875–1888, 2022.

[91] E. Wang, J. Cai, Y. Yang, W. Liu, H. Wang, B. Yang, and J. Wu, "Trustworthy and efficient crowdsensed data trading on sharding blockchain," *IEEE JSAC*, vol. 40, no. 12, pp. 3547–3561, 2022.

[92] D. Liu, C. Huang, J. Ni, X. Lin, and X. S. Shen, "Blockchain-cloud transparent data marketing: Consortium management and fairness," *IEEE Transactions on Computers*, vol. 71, no. 12, pp. 3322–3335, 2022.

[93] Y. Dai, D. Xu, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks," *IEEE TVT*, vol. 69, no. 4, pp. 4312–4324, 2020.

[94] S. Guo, X. Hu, S. Guo, X. Qiu, and F. Qi, "Blockchain meets edge computing: A distributed and trusted authentication system," *IEEE TII*, vol. 16, no. 3, pp. 1972–1983, 2019.

[95] L. Yin, J. Xu, and Q. Tang, "Sidechains with fast cross-chain transfers," *IEEE TDSC*, vol. 19, no. 6, pp. 3925–3940, 2022.

[96] X. Luo, K. Xue, Q. Sun, and J. Lu, "Crosschannel: Efficient and scalable cross-chain transactions through cross-and-off-blockchain micropayment channel," *IEEE TDSC*, 2024.

[97] J. Liu, P. Li, F. Zhang, and K. Ren, "monoCash: A channel-free payment network via trusted monotonic counters," *IEEE TDSC*, vol. 21, no. 5, pp. 4770–4783, 2024.

[98] Y. Guan, H. Zheng, J. Shao, R. Lu, and G. Wei, "Fair outsourcing polynomial computation based on the blockchain," *IEEE TSC*, vol. 15, no. 5, pp. 2795–2808, 2021.

[99] C. Dong, Y. Wang, A. Aldweesh, P. McCorry, and A. Van Moorsel, "Betrayal, distrust, and rationality: Smart counter-collusion contracts for verifiable cloud computing," in *Proc. of ACM CCS*, 2017.

[100] H. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, "Arbitrum: Scalable, private smart contracts," in *Proc. of USENIX SECURITY*, 2018.

[101] X. Xiao, Y. Zhang, X. Dong, L. Wang, Y. Xiang, and X. Cao, "Fair outsourcing paid in fiat money using blockchain," *IEEE TSC*, 2022.

[102] Z. Yang, K. Liu, Y. Chen, W. Chen, and M. Tang, "Two-level stackelberg game for iot computational resource trading mechanism: A smart contract approach," *IEEE TSC*, vol. 15, no. 4, pp. 1883–1895, 2022.

[103] Z. Xie, R. Wu, M. Hu, and H. Tian, "Blockchain-enabled computing resource trading: A deep reinforcement learning approach," in *Proc. of IEEE WCNC*, 2020.

[104] X. Ren, M. Xu, D. Niyato, J. Kang, C. Qiu, and X. Wang, "Paramart: Parallel resource allocation based on blockchain sharding for edge-cloud services," *IEEE TSC*, vol. 17, no. 4, pp. 1655–1669, 2024.

[105] H. Wang, Q. Wang, D. He, Q. Li, and Z. Liu, "BBARS: Blockchain-based anonymous rewarding scheme for V2G networks," *IEEE IOT*, vol. 6, no. 2, pp. 3676–3687, 2019.

[106] N. V. Keizer, O. Ascigil, I. Psaras, and G. Pavlou, "Rewarding relays for decentralised nat traversal using smart contracts," in *Proc. of ACM/IEEE MobiHoc*, 2020.

[107] J. Liu, Y. Xue, Z. Peng, C. Lin, and X. Huang, "Fairrelay: Fair and cost-efficient peer-to-peer content delivery through payment channel networks," *arXiv preprint arXiv:2405.02973*, 2024.

[108] Z. Hong, S. Guo, P. Li, and W. Chen, "Pyramid: A layered sharding blockchain system," in *Proc. of IEEE INFOCOM*, 2021.

[109] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proc. of ACM CCS*, 2018.

[110] W. Wang, D. Niyato, P. Wang, and A. Leshem, "Decentralized caching for content delivery based on blockchain: A game theoretic perspective," in *Proc. of IEEE ICC*, 2018.

[111] I. Homoliak, S. Venugopalan, D. Reijsbergen, Q. Hum, R. Schumi, and P. Szalachowski, "The security reference architecture for blockchains: Toward a standardized model for studying vulnerabilities, threats, and defenses," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 341–390, 2020.

[112] Z.-G. Wan, R. H. Deng, D. Lee, and Y. Li, "Microbtc: Efficient, flexible and fair micropayment for bitcoin using hash chains," *JCST*, vol. 34, pp. 403–415, 2019.

[113] R. Song, S. Gao, Y. Song, and B. Xiao, "ZKDET: A Traceable and Privacy-Preserving Data Exchange Scheme based on Non-Fungible Token and Zero-Knowledge," in *Proc. of IEEE ICDCS*, 2022, pp. 224–234.

[114] Z. Wang, H. Jin, W. Dai, K.-K. R. Choo, and D. Zou, "Ethereum smart contract security research: survey and future research opportunities," *Frontiers of Computer Science*, vol. 15, no. 2, p. 152802, 2021.

[115] Y. Zheng, H. Duan, X. Yuan, and C. Wang, "Privacy-aware and efficient mobile crowdsensing with truth discovery," *IEEE TDSC*, vol. 17, no. 1, pp. 121–133, 2017.

[116] S. Wang, L. Shi, Q. Hu, J. Zhang, X. Cheng, and J. Yu, "Privacy-aware data trading," *IEEE TIFS*, vol. 16, pp. 3916–3927, 2021.

[117] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on ethereum systems security: Vulnerabilities, attacks, and defenses," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–43, 2020.

[118] L. Zhou, X. Xiong, J. Ernstberger, S. Chaliasos, Z. Wang, Y. Wang, K. Qin, R. Wattenhofer, D. Song, and A. Gervais, "Sok: Decentralized finance (defi) attacks," in *Proc. of IEEE S&P*, 2023.

[119] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proc. of ACM STOC*, 1988.

[120] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Proc. of CRYPTO*, 2012.

[121] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proc. of ACM STOC*, 1987.

[122] E. M. Songhori, S. U. Hussain, A.-R. Sadeghi, T. Schneider, and F. Koushanfar, "Tinygarble: Highly compressed and scalable sequential garbled circuits," in *Proc. of IEEE S&P*, 2015.

[123] H. Kılınç Alper and A. Küpçü, "Coin-based multi-party fair exchange," in *Proc. of ACNS*, 2021.

[124] R. Xu, J. Joshi, and C. Li, "Nn-emd: Efficiently training neural networks using encrypted multi-sourced datasets," *IEEE TDSC*, vol. 19, no. 4, pp. 2807–2820, 2022.

[125] T. Lu, B. Zhang, and K. Ren, "Privdata network: A privacy-preserving on-chain data factory and trading market," *IEEE TDSC*, vol. 21, no. 3, pp. 1424–1436, 2024.

[126] W. Dai, C. Dai, K.-K. R. Choo, C. Cui, D. Zou, and H. Jin, "SDTE: A secure blockchain-based data trading ecosystem," *IEEE TIFS*, vol. 15, pp. 725–737, 2019.

[127] X. Liu, B. Qin, R. H. Deng, and Y. Li, "An efficient privacy-preserving outsourced computation over public data," *IEEE TSC*, vol. 10, no. 5, pp. 756–770, 2015.

[128] J. Liu, X. He, R. Sun, X. Du, and M. Guizani, "Privacy-Preserving Data Sharing Scheme with FL via MPC in Financial Permissioned Blockchain," in *Proc. of IEEE ICC*, 2021.

[129] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct non-interactive zero knowledge for a von neumann architecture," in *Proc. of USENIX SECURITY*, 2014.

[130] J. Groth, "On the size of pairing-based non-interactive arguments," in *Proc. of EUROCRYPT*, 2016.

[131] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *Proc. of IEEE S&P*, 2014.

[132] S. Ames, C. Hazay, Y. Ishai, and M. Venkitasubramaniam, "Ligero: Lightweight sublinear arguments without a trusted setup," in *Proc. of ACM CCS*, 2017.

[133] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *Proc. of IEEE S&P*, 2018.

[134] A. Chiesa, Y. Hu, M. Maller, P. Mishra, N. Vesely, and N. Ward, "Marlin: Preprocessing zksnarks with universal and updatable srs," in *Proc. of EUROCRYPT*, 2020.

[135] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2018.

[136] V. Costan and S. Devadas, "Intel SGX explained," *Cryptology ePrint Archive*, 2016.

[137] C. Cai, L. Xu, A. Zhou, and C. Wang, "Toward a secure, rich, and fair query service for light clients on public blockchains," *IEEE TDSC*, vol. 19, no. 6, pp. 3640–3655, 2021.

[138] S. Lee, M.-W. Shih, P. Gera, T. Kim, H. Kim, and M. Peinado, "Inferring fine-grained control flow inside SGX enclaves with branch shadowing," in *Proc. of USENIX SECURITY*, 2017.

[139] S. Shinde, Z. L. Chua, V. Narayanan, and P. Saxena, "Preventing page faults from telling your secrets," in *Proc. of ACM ASIA CCS*, 2016.

[140] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proc. of ACM CCS*, 2012.

[141] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE TDSC*, vol. 15, no. 5, pp. 840–852, 2016.

[142] R. McLaughlin, C. Kruegel, and G. Vigna, "A large scale study of the ethereum arbitrage ecosystem," in *Proc. of USENIX SECURITY*, 2023.

[143] C. Lin, D. He, X. Huang, M. K. Khan, and K.-K. R. Choo, "Dcap: A secure and efficient decentralized conditional anonymous payment system based on blockchain," *IEEE TIFS*, vol. 15, pp. 2440–2452, 2020.

[144] Z. Guan, Z. Wan, Y. Yang, Y. Zhou, and B. Huang, "Blockmaze: An efficient privacy-preserving account-model blockchain based on zk-snarks," *IEEE TDSC*, vol. 19, no. 3, pp. 1446–1463, 2020.

[145] A. Ahmad, K. Kim, M. I. Sarfaraz, and B. Lee, "OBLIVIATE: A Data Oblivious Filesystem for Intel SGX," in *Proc. of NDSS*, 2018.

[146] A. Rane, C. Lin, and M. Tiwari, "Raccoon: Closing Digital Side-Channels through Obfuscated Execution," in *Proc. of USENIX SECURITY*, 2015.

[147] G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate, and M. Maffei, "Anonymous multi-hop locks for blockchain scalability and interoperability," in *Proc. of NDSS*, 2018.

[148] L. Aumayr, P. Moreno-Sanchez, A. Kate, and M. Maffei, "Blitz: Secure multi-hop payments without two-phase commits," in *Proc. of USENIX SECURITY*, 2021.

[149] S. Wadhwa, L. Zanolini, A. Asgaonkar, F. D'Amato, C. Fang, F. Zhang, and K. Nayak, "Data independent order policy enforcement: Limitations and solutions," in *Proc. of ACM CCS*, 2024.

[150] M. Saad, A. Anwar, S. Ravi, and D. Mohaisen, "Revisiting nakamoto consensus in asynchronous networks: A comprehensive analysis of bitcoin safety and chainquality," in *Proc. of ACM CCS*, 2021.

[151] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of bft protocols," in *Proc. of ACM CCS*, 2016.

[152] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *Proc. of IEEE S&P*, 2017.

[153] D. Lu, T. Yurek, S. Kulshreshtha, R. Govind, A. Kate, and A. Miller, "Honeybadgermpc and asynchromix: Practical asynchronous mpc and its application to anonymous communication," in *Proc. of ACM CCS*, 2019.

[154] C. Cai, J. Weng, X. Yuan, and C. Wang, "Enabling reliable keyword search in encrypted decentralized storage with fairness," *IEEE TDSC*, vol. 18, no. 1, pp. 131–144, 2018.

[155] H. Duan, Y. Du, L. Zheng, C. Wang, M. H. Au, and Q. Wang, "Towards practical auditing of dynamic data in decentralized storage," *IEEE TDSC*, vol. 20, no. 1, pp. 708–723, 2022.

[156] B. Zhang, H. Cui, X. Liu, Y. Chen, Z. Yu, and B. Guo, "Decentralized and Secure Deduplication with Dynamic Ownership in MLaaS," *JISA*, vol. 76, p. 103524, 2023.

[157] B. Zhang, H. Cui, Y. Chen, X. Liu, Z. Yu, and B. Guo, "Enabling secure deduplication in encrypted decentralized storage," in *Proc. of NSS*, 2022.

[158] X. Ren, C. Qiu, Z. Chen, X. Wang, D. Niyato, and W. Wang, "CompCube: A Space-Time-Request Resource Trading Framework for Edge-Cloud Service Market," *IEEE TSC*, vol. 16, no. 5, pp. 3252–3264, 2023.

Hao Zeng received his B.S. degree in computer science and technology from Northwestern Polytechnical University, Xi'an, China in 2023. He is currently working toward the M.S. degree with the Northwestern Polytechnical University. His research interests include blockchain, data security, and cloud security.

Helei Cui received the Ph.D. degree in Computer Science from the City University of Hong Kong, in 2018. He is currently a Professor with the School of Computer Science, Northwestern Polytechnical University, China. His research interests include Trustworthy Crowd Computing, Privacy-Preserving Computing, Edge Intelligence, Decentralized Cloud Storage, etc.

Man Li received her bachelor's degree in computer science and technology from Yunnan University in 2022. She is currently working toward a PhD degree with the School of Computer at Northwestern Polytechnical University. Her research interests include secure computing and edge computing.

Bo Zhang received his B.S. degree in computer science and technology from Northwestern Polytechnical University, Xi'an, China in 2023. He is currently working toward the M.S. degree with the Northwestern Polytechnical University. His research interests include data security, blockchain, cloud computing, etc.

Chengjun Cai is currently an Associate Professor in City University of Hong Kong (Dongguan). He obtained his Ph.D. degree in Computer Science from City University of Hong Kong in June 2021, and Bachelor degree in Computer Science and Technology from Jinan University, Guangzhou in July 2016. His research interests include decentralized data-driven applications, blockchain security, and applied cryptography.

Zhiwen Yu received the M.E. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China in 2003 and 2005, respectively. He is a Professor with Northwestern Polytechnical University and Harbin Engineering University. His current research interests include pervasive computing, mobile crowdsensing, internet of things, and intelligent information technology.

Bin Guo received the Ph.D. degree in computer science from Keio University, Minato, Japan, in 2009, He was a Postdoctoral Researcher with the Institut TELECOM SudParis, Essonne, France. He is currently a Professor with Northwestern Polytechnical University, Xi'an, China. His research interests include ubiquitous computing, mobile crowd sensing, and HCI.