IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 February 2024, accepted 27 February 2024, date of publication 29 February 2024, date of current version 8 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3371859

RESEARCH ARTICLE

Fast Payments Across Heterogeneous Blockchains for Internet of Things

ZHIKAI LIN, KEXING WANG, YONGDONG WU, AND DINGCHENG LI
College of Cyber Security, Jinan University, Guangzhou 510632, China

Corresponding author: Kexing Wang (jnukxwang@foxmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61932011, Grant 62102165, Grant U2001205, and Grant 62032025, in part by the Guangdong Key Laboratory of Data Security and Privacy Preserving under Grant 2023B1212060036, in part by the Guangdong-Hong Kong Joint Laboratory for Data Security and Privacy Preserving under Grant 2023B1212120007, and in part by the Science and Technology Major Project of Tibetan Autonomous Region of China Grants XZ202201ZD0006G01.

ABSTRACT Internet of Things data exchange services facilitate the connection and flexible exchange of data among distributed IoT data sources. Traditional IoT data exchange services typically involve direct data uploads to the cloud, allowing service providers to generate significant profits through IoT data analysis. However, this approach exposes data owners to privacy risks while not benefiting them from the service providers' gains. Furthermore, the use of different blockchain architectures by various IoT networks presents challenges for achieving cross-chain interoperability. Although blockchain technology has ensured the security and integrity of IoT data transactions, its efficiency remains a concern. To address these issues, we propose a real-time cross-chain transaction scheme for IoT data exchange in an IoT environment. Our scheme aims to address the real-time security challenges associated with cross-chain data interactions. First, we established a secure and fair settlement scheme by employing a smart contract model based on collateral and challenge mechanisms. In addition, cryptographic tools are utilized to enhance the expressive capabilities of weakly secure service providers, thereby reducing reliance on third-party security assumptions. Finally, through experimentation, we demonstrate the effectiveness of our smart contract model in reducing the time required for on-chain transactions.

INDEX TERMS Atomic cross-chain swap, Internet of Things, fast payments, blockchain.

I. INTRODUCTION

The Internet of Things (IoT) is revolutionizing industries by connecting billions of devices worldwide, projected to reach 20.4 billion by 2025, streamlining operations and enhancing productivity across various sectors including manufacturing, environmental monitoring, and healthcare [1], [2], [3]. Despite its growth, IoT faces critical challenges such as data privacy, centralization vulnerabilities, and the inefficiency of large-scale data management. Blockchain technology emerges as a pivotal solution, offering decentralization, enhanced security, and reliable data exchange without centralized authority, addressing IoT's scalability and trust issues [4]. Blockchain technology is being applied across various IoT domains, such as data outsourcing on a single chain, distributed storage of firefighting IoT data, and IoT data sharing across multiple chains [5], [6], [7], [8]. Numerous endeavors have been made to enhance and adapt business workflows to maximize the utilization of IoT data, with IoT data exchange emerging as a prominent scheme [9], [10], [11]. IoT data exchange platforms have emerged to connect diverse and distributed data sources, facilitating the exchange of IoT data among data owners [12]. However, when data exchange scenarios involve different IoT network devices, a new challenge arises the cross-chain challenge encountered when different IoT networks adopt varying blockchain architectures. This challenge encompasses real-time security concerns about cross-chain data interactions within the IoT environment. This backdrop underscores the urgent need for innovative frameworks that can bridge these technological gaps, ensuring seamless, secure, and efficient IoT data exchanges in a rapidly evolving digital ecosystem [13], [14], [15], [16].

The IoT blockchain transaction methodology comprises distinct phases: transaction preparation, transaction submission, transaction confirmation, and data acquisition. In contrast to prevalent payment procedures, blockchain exhibits an extended duration for transaction confirmation.

The associate editor coordinating the review of this manuscript and approving it for publication was Renato Ferrero.

This delay is attributed to the protracted consensus mechanism inherent in blockchain technology, resulting in diminished throughput. As the size of blockchain network nodes escalates, this latency becomes more pronounced. IoT-centric applications are designed to deliver expeditious, high-quality services to ensure optimal user satisfaction. Particularly in scenarios necessitating instantaneous interaction, such as immediate payments and intricate transactions involving data-dependent services, end-users must be able to access real-time data exchange services without incurring any losses [17].

In contemporary IoT blockchain frameworks, augmented throughput is presently achieved through enhancements to the underlying Proof-of-Work (POW) mechanism, entailing a reduction in block size and an augmentation of the rate at which new blocks are incorporated into the blockchain. Nonetheless, the resultant improvements fall short of addressing the imperative for real-time switching capabilities. For instance, in POW-consensus-based blockchains, such as the Bitcoin blockchain, a meager provision of merely 7 transactions per second is available. IoT blockchain scheme endeavors to bolster throughput by modifying block capacity. Even so, the endeavor to augment throughput is exemplified by a reduction in block size of up to 4MB, facilitating the confirmation of twenty-seven transactions within a single second. Moreover, the IoT blockchain scheme employing the Lightning Network or Two-Phase Commit protocol having low throughput fails to satisfy the requisites of the majority of IoT use cases. Therefore, enhancing the transaction speed of blockchains has both theoretical and practical significance. Cross-chain technology has emerged as a promising scheme to address blockchain scalability concerns. In our proposed scheme, we introduce blockchain oracles to enable real-time data interaction, propose a pledge-based cross-chain mechanism, and establish an off-chain service provider that leverages BLS threshold signatures to ensure the security of data exchange among IoT devices. This holistic approach aims to address the aforementioned challenges and promote efficient and secure data interactions within IoT ecosystems.

Previous studies proposed a scheme that utilizes off-chain payments for IoT data transactions, which deviates from the initial intention of on-chain payments. These studies primarily focused on single-chain structures and did not consider their applicability to multi-chain cross-chain transactions. In light of these limitations, to address security and traceability concerns in IoT data exchange, we propose the utilization of atomic swaps combined with blockchain for real-time data exchange within the IoT environment. Several studies have explored the integration of blockchain and IoT for data payment [7], [8], [18], [19]. However, these studies had certain limitations and made strong assumptions. Specifically, they require users to recharge their off-chain channels, even though a user only needs to make a single payment. Essentially, this approach relies on an off-chain payment method, which contradicts the original intention of the on-chain payment design. Off-chain channel payment involves two transactions: a deposit transaction and a redemption transaction. However, in the blockchain, a user's normal payment transaction is represented by a single on-chain transaction. Additionally, most atomic swaps rely on a trusted third party, which poses challenges in practical applications where maintaining complete trust is difficult to ensure.

In this paper, we aim to design an on-chain cross-chain transaction scheme that supports IoT-enabled real-time data exchange where the core challenge is to design a secure off-chain service provider model. Then, we design a fair and fast cross-chain atomic exchange scheme. In a nutshell, our contributions can be depicted as follows:

- A cross-chain trading scheme has been designed that can support real-time data exchange in the IoT environment, which achieves fast autonomous data exchange between data consumers and data providers.
- A secure service provider model in the scheme has been designed that reduce the security strength required by the service provider.
- A pledge-challenge model has been utilized between data providers and data consumers so that the two parties do not need to wait for transaction confirmation to achieve the purpose of fast transactions and use punishment mechanisms to solve the problem of distrust between participants.

The remainder of this paper is organized as follows. In Section II, we introduce related studies. In Section III, we introduce preliminary knowledge of the cryptography and blockchain used in this study. In Section IV, we propose a fast cross-chain scheme that supports real-time data exchange in the IoT. In Section V, we describe the proposed cross-chain scheme in detail. In Section VI, the security of the proposed scheme is analyzed. In Section VII, we validate the feasibility of the proposed detection scheme through experiments conducted using heterogeneous chains. Finally, in Section VIII, we present our conclusions.

II. RELATED WORK

A. IOT CROSS-CHAIN APPLICATION

After the integration of blockchain and IoT, the IoT ecosystem has experienced significant improvements in terms of interoperability, security, traceability, and reliability [20]. Ou et al. [21] have presented work that underscores the significance of cross-chain technology in establishing an Internet of Blockchains and facilitating blockchain interoperability. With the continuous evolution and increasing diversity of the blockchain ecosystem, there is a growing demand for cross-chain technology to adapt and ensure high efficiency and security in cross-chain operations. However, the implementation of cross-chain technology poses various challenges, including efficiency concerns, security issues, and the robustness of connections between cross-chain networks.

To address the aforementioned issues, contemporary research primarily encompasses two distinct categories: the proposal of novel cross-chain communication protocols and enhancements to consensus mechanisms.

1) THE PROPOSAL OF NOVEL CROSS-CHAIN COMMUNICATION PROTOCOLS

Robert et al. [7] analyzed the current state of off-chain payment technologies applied in IoT scenarios. They presented a framework for the integration of the Lightning Network into the IoT ecosystem, along with a novel Lightning Network channel optimization algorithm. In a separate study, Meijers et al. [8] introduced a trustless data trading system, devising secure protocols for off-chain data transactions between buyers and sellers. Their approach aims to minimize on-chain operations for both parties, thereby reducing expenses. However, both of these schemes leverage off-chain payment technologies, such as the Lightning Network, which inherently diverges from the original intent of blockchain technology. Additionally, the delayed transaction confirmation resulting from consensus mechanism delays renders these schemes unsuitable for immediate payment requirements in the IoT transaction system.

2) ENHANCEMENTS TO CONSENSUS MECHANISMS

Wang et al. [22] introduced a credit-based incentive approach, aiming to enhance the consensus-building process by incorporating a new reputation module. This module allows each participant to share a global view of reputation, illustrating the potential of reputation as an incentive mechanism within a consensus protocol. The approach demonstrates commendable efficiency and safety. However, the absence of consensus introduces delays in transaction confirmation, rendering it unsuitable for most real-time IoT systems that demand instantaneous confirmation. And Huang et al. [23] proposed a supervisable consensus scheme based on an enhanced version of DPOS-PBFT (Delegated Proof of Stake-Practical Byzantine Fault Tolerance). While this improved scheme holds promise for the IoT blockchain consensus mechanism, it is important to note that machine storage requirements escalate with an increasing number of nodes, and the delay in final block confirmation extends, consequently elongating transaction confirmation times.

B. CROSS-CHAIN INTEROPERABILITY AND DELAY

Blockchain interoperability encompasses the capacity of different blockchain systems to assets exchange and share information, data, or assets. Given its pivotal role in the contemporary decentralized economy, numerous frameworks have been developed to facilitate these services.

Sonkamble et al. [24] introduced a blockchain-driven Electronic Health Record (EHR) framework named MyBlock-EHR. This framework utilizes the partitioning of EHR into on-chain and off-chain storages to ensure performance guarantees, allowing for the retrieval of valid off-chain data. Hei et al. [25] presented an all-encompassing cross-chain exchange system known as Practical AgentChain. This innovative cross-chain system is grounded in smart contracts and trusted computing techniques. Practical AgentChain facilitates the mapping of various coins to corresponding tokens, enabling seamless trading transactions.

Anyswap[1] as a cross-chain token swapping service designed to facilitate the exchange and migration of tokens across 51 different blockchains. The platform employs a distributed network comprising secure multi-party computation (SMPC) nodes to oversee the minting and burning processes of wrapped tokens. Consensus on verification is achieved by these SMPC nodes using a distributed threshold signature algorithm. WeCross[2] stands as an open-source, high-efficiency blockchain interoperability platform with the primary goal of supporting widely-used permissioned blockchains, including but not limited to Hyperledger Fabric [26] and FISCO BCOS.[3] Within WeCross, two cross-chain transaction protocols are implemented: Two-Phase Commit protocol and hash time lock contract.

Mazumdar [27] introduced Quick Swap, an enhanced HTLC-based atomic swap protocol incorporating innovative concepts to mitigate latency and fairness issues associated with traditional atomic swaps. By integrating a griefing penalty mechanism, Quick Swap aims to encourage timely settlements by penalizing delays and offering an option to cancel swaps, addressing some flexibility and fairness challenges of conventional HTLC exchanges while still leveraging blockchain's inherent features like smart contracts and cryptographic proofs for secure and enforceable exchanges. Imoto et al. [28] developed an improved atomic cross-chain swap protocol that supports smart contracts, aimed at reducing the time and space costs associated with exchanges across multiple blockchains. This protocol introduces a novel approach that eliminates the need for storing swap topology in contracts, utilizing timelocks based on the number of signatures a contract receives instead. This method promises immediate activation of entering arc contracts upon the triggering of any leaving arc contract, designed to enhance the efficiency and simplicity of multichain transactions.

Nevertheless, these schemes exhibit certain limitations concerning cross-chain interoperability and transaction speed. Both the MyBlockEHR [24] and Practical AgentChain [25] schemes exclusively executed homogenous cross-chain functionality on the Ethereum platform, lacking support for heterogeneous cross-chain operations. Furthermore, Anyswap, Wecross, [27] and [28] are susceptible to consensus mechanism constraints, resulting in significant transaction delays. Among these, Anyswap and Wecross exhibit a relatively restricted range of supported blockchain networks.

III. BACKGROUND

A. SHAMIR SECRET SHARING

The Shamir Secret Sharing Scheme [29] allows the sharing of a secret by dividing it into pieces and giving each

[1] https://multichain.org
[2] https://wecross.readthedocs.io/_/downloads/zh-cn/latest/pdf/
[3] https://fisco-bcos-documentation.readthedocs.io/en/latest/

participant their own unique part, where some, or all, the parts are needed to reconstruct the secret. Requiring all the participants to combine the secret might be impractical, and therefore sometimes the threshold scheme is used where any k of the parts is sufficient to reconstruct the original secret. Mathematical Definition of the Shamir Secret Sharing Scheme:

- Knowledge of any k or more $\mathcal{D}_i$ pieces makes $\mathcal{D}$ easily computable
- Knowledge of any k-1 or fewer $\mathcal{D}_i$ pieces leaves $\mathcal{D}$ completely undetermined (all possible values are equally likely)

The process of reconstructing the secret is performed using polynomial interpolation, where given k points in the 2-dimensional plane $(x,y), \ldots, (x_k, y_k)$, with distinct x_i, there is one polynomial q(x_i) $= y_i$ for all i. Efficient methods exist for performing the reconstruction process.

B. BLS SIGNATURE

The aggregated signature scheme of the BLS algorithm based on bilinear mapping can generate short signatures, that is, the length of the aggregated signature is the same as that of a single signature. Several blockchain-related improvement schemes have adopted bilinear map-based aggregate signature technology [30]. The BLS signature protocol was proposed by Boneh in 2001 [31]. The BLS signature structure is as follows.

- Key-Gen: The private key sk is a secret integer and the public key is $pk = sk * G$. G in the formula is a q-order cyclic group.
- Sign: Message m, it is mapped to a point on the curve, recorded as F, then the signature $Sig = sk * F$.
- Verification: Based on the discrete logarithm problem, the BLS signature defines a bilinear mapping function e in three q-order cyclic groups (G_1, G_2, G_T): $G_1 * G_2 \rightarrow G_T$. Using the characteristics of the elliptic curve bilinear pairing function, the signature verification process can be deduced as:

$$e(pk, F) = e(sk * G, F) = e(G, sk * F) = e(G, Sig) \tag{1}$$

The BLS signature can be readily converted into threshold versions through Shamir secret sharing, owing to its suitability for signature aggregation [32]. In the proposed scheme, we establish a trustworthy third party within the pledge-challenge model by employing BLS threshold signatures. Moreover, we provide a comprehensive explanation of the verification process for BLS threshold signatures within the proposed scheme.

C. BLOCKCHAIN AND SMART CONTRACT

Satoshi Nakamoto launched the first cryptocurrency using distributed ledger technology, known as blockchain [13]. Blockchain is a decentralized ledger that secures, verifies, and records all peer-to-peer transactions quickly, securely, and transparently. This technology can solve many problems that traditional technologies cannot solve. Blockchain has attracted extensive attention from academia owing to its commercial and research value. A review of the main features of blockchain is as follows.

- Complete Decentralization: This is based on a distributed P2P network in which many untrusted nodes can achieve fair data exchange without relying on a central party.
- Correct Execution: Blockchain is a global computer in which each blockchain node can trace and verify the correctness of the data computation.
- Tamper-resistance: The data (i.e., blocks and transactions) are tamper-resistant because they are organized in a special data structure (Merkle tree and hash chain).

Smart contracts are designed to construct a decentralized application (DApp) [33] that facilitates the process of executing an application automatically and verifiably. People can participate in one DApp by providing valid inputs through on-chain transactions to call a function in a smart contract.

D. BLOCKCHAIN ORACLE

The blockchain oracle serves as a vital intermediary among real-world events physical occurrences and blockchain-based smart contracts. Its primary function is to retrieve, validate, and transmit information to smart contracts for execution. Oracle servers play a crucial role in responding to various queries, such as the exchange rate between Ethereum and the USD. Oracles can consult multiple sources or a singular source to obtain necessary information, subsequently relaying it back to smart contracts. These data feed services can also function as computation oracles, as exemplified by Truebit [34], wherein they perform computation-intensive tasks designated by users off-chain. By supplying computational power to blockchains, these oracles facilitate the establishment of a decentralized token economy [35]. In our case, the blockchain oracle was employed as a data verification oracle, utilizing the pull-based inbound oracle pattern to direct the flow of information [36]. This pattern allows for the transmission of external world data into blocks, effectively notifying the external world of the requirement to supply essential information to the network.

E. CROSS-CHAIN COMMUNICATION AND NOTARY TECHNOLOGY

The informal definition of cross-chain communication can be concisely articulated as follows: a process P writes Tx$_P$ to X and Q writes Tx$_Q$ to Y if and only if the descriptions of Tx$_P$ and Tx$_Q$ satisfy desc(Tx$_P$) $= d_Q \wedge$ desc(Tx$_Q$) $= d_P$, where d_P and d_Q represent the constraints of P on Tx$_P$ and Q on Tx$_Q$ [37]. The protocol, composed of the distributed ledgers X and Y, participants P and Q, and transactions Tx$_P$ and Tx$_Q$, initiates with the establishment of application-specific parameters during the setup stage, which is typically negotiated off-chain. Following successful initialization, the commitment stage involves X publishing

TABLE 1. The notations of explanation.

Notation	Explanation
$[n]$	The set of natural numbers from 1 to n.
sp$_i$	Notary member in SP, $i \in [n]$.
(pk$_{sp_i}$,sk$_{sp_i}$)	A Key pair held by sp_i is designed to ensure the atomicity of transactions.
status$_i$	Off-chain verification status.
Gen$_{bls}$(.)	Generating public-private key pair according to Shamir Secret Sharing and BLS signature.
f_i(.)	The polynomial function is selected by sp_i.
calculate(.)	Computing the results after undergoing magnification processing.
getTransaction(.)	Obtaining the cross-chain transactions corresponding to the participating parties.
getDeposits(.)	Obtaining the balance of the mortgage account.
getBalances(.)	Obtaining the balance of the transaction.
call(.)	Calling the target contract.
check(.)	Calling the Oracle contract to verify whether the deposit is sufficient off-chain.
H(.)	A function to calculate hashes on the blockchain.
L_i(.)	The Lagrangian basis function.
challenge(.)	Participant submits query requests for a cross-chain transaction.
checkValidity(.)	Verifying that the transaction exists on the standard chain.
sendTransaction(.)	Broadcasting a transaction to the blockchain.

a verifiable commitment to adhere to the protocol, which Q subsequently verifies. After verifying Q, a similar commitment is made on Y. The protocol either progresses to a termination stage upon verification failure or inaction by Q, rolling back any changes to revert to the pre-protocol state, not a literal rollback but a figurative "restoration".

In cross-chain communication, cross-chain verification technologies ensure the secure execution of cross-chain communication protocols. Notary mechanisms, widely applied in heterogeneous network cross-chain verifications, are categorized into single-signature notaries, multi-signature notaries, and distributed signature notaries. Single-signature notaries operate in a centralized manner, offering high processing efficiency but risking a single point of failure. Multi-signature notaries require consensus among several notaries, each signing on their respective ledgers to complete transactions, mitigating centralization issues but necessitating multi-signature support across all involved chains. Distributed signature notaries employ secure multi-party computation to reconstruct signatures with sub-signatures, reducing reliance on third parties. Security increases from single-signature to distributed notaries, as does implementation complexity.

IV. OVERVIEW

In this section, we present a cross-chain model that facilitates real-time data exchange in an IoT context. The functions performed by various entities in the IoT data exchange model are detailed in Section IV-A. Section IV-B outlines the threat model associated with the cross-chain model, while Section IV-C delineates the design goals pursued by the cross-chain transaction model, the primary objective of which is to expedite cross-chain transactions for both data consumers and data providers, eliminating the need to wait for transaction confirmation on the blockchain. Subsequently, upon completion of a cross-chain transaction, any participant within the blockchain network can initiate a challenge, whereby the involved parties failing to fulfill their commitments will be subject to mandatory punitive measures.

A. SYSTEM MODEL

Figure 1 illustrates the cross-chain model that supports real-time data exchange in the IoT. This model includes the following three roles.

1) **Data consumer** denoted as C, assumes the role of a transaction participant engaged in data procurement. To obtain the necessary IoT data, C initiates a payment transaction in the blockchain.
2) **Data provider** denoted as P, fulfills the role of a data provider wherein IoT devices transmit data to the cloud infrastructure, which is under the management of P. In our proposed model, P assumes the role of a transaction participant, offering data provision services. P publishes details regarding the IoT data sought by C.
3) **Service Provider** denoted as SP, comprises a collection of notaries. Each notary possesses a key generated by the key generation algorithm of the BLS threshold signature scheme, which is utilized to authenticate cross-chain registration requests and penalty transactions.

The model also includes the following six components. The relationship between the Oracle contract, the user contract, and the service provider is shown in Figure 3.

1) **User Contract** denoted as C_u, is a smart contract deployed on the blockchain that facilitates the implementation of data exchange functions during cross-chain transactions. During this process, the data consumer initiates a monetary transfer to the data provider. Subsequently, the data provider submits a data identification to the data consumer within the blockchain environment.
2) **Oracle contract** denoted as C_o, refers to a smart contract invoked by C_u during its execution. Subsequently, C_o engages in interactions with SP to obtain necessary authentication data from the off-chain domain.
3) **Service Provider Contract** denoted as C_{sp}, serves multiple purposes in facilitating transaction verification and ensuring the integrity of the challenged transaction. First, it aids the notary in verifying the status of the contested transaction by centrally validating the Merkle path associated with the transaction within the prescribed time frame. Moreover, it enables the contract to designate the transaction's legal status. Additionally, the contract is employed to secure and deduct the deposits of the participants engaged in atomic exchange, utilizing the aggregated public key signed by the BLS threshold as a means to lock the participants' deposits. In the event of malicious behavior

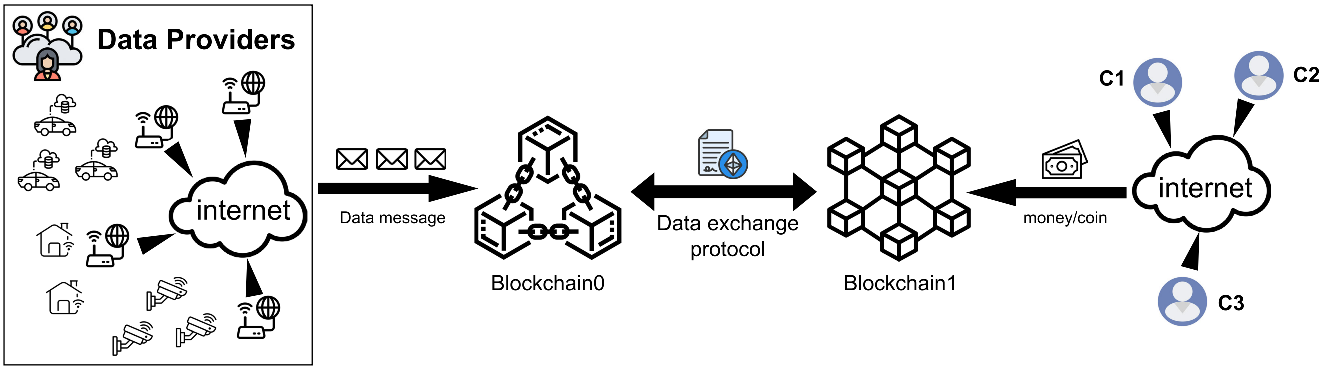


FIGURE 1. The system model.

by a participant, the penalty transaction, as determined by the notary and implemented through the aggregated signature, is executed within the contract to effectuate deposit deductions.

4) **Blockchains** were heterogeneous in the proposed scheme. We use B_u and B_d to identify the user and data blockchains, which are designed to ensure atomicity, traceability, and security of IoT data transactions.
5) **Distributed Data Storage** denoted as DDS (e.g., IPFS), was employed to record the initialization parameters of the notary.
6) **Data exchange system** denoted as Sys, was designed to facilitate query operations on IoT cloud data. Its input is the data identifier submitted by P to the blockchain, which is a ciphertext encrypted by the user identity and the corresponding data index. Its output is data corresponding to the data index. When C submits the data identification bound to its identity, Sys verifies whether the identity in data identification is consistent with C. If the result is consistent, Sys distributes the data to C. Otherwise, the request is refused. When negotiating with P, C must confirm that the data already exists in Sys. The input value for Sys verification is the hash value of the data identifier that P must transmit to the blockchain.

The system procedure and interplay between various roles and components are illustrated in Figure 2. During the initial step, P and C engaged in negotiations about the deposits required for subsequent transactions. As depicted in the second step of the figure, SP assigns a public key to the notary set via the application of the Shamir key generation algorithm, subsequently making it accessible to the DDS. This step is crucial for initializing a secure SP. In the subsequent steps (third and fourth), P and C lock the deposit with deposit accounts created by the aggregated public key and negotiate the data index and transaction amount, among other information. The first step and these two steps collectively serve as the security collateral assurance for the entire data exchange. Furthermore, these two steps, in conjunction with the subsequent fifth step, serve to prevent premature data disclosure, ensuring the security of the data. Subsequently, C verifies the accuracy of the data index using Sys in the fifth step. Subsequently, in the sixth step, C invokes C_u to execute the fund transfer to P, while concurrently, P invokes C_u to provide the data consumer with a data index that is bound to their identity. Finally, in the seventh step, C supplies the index to the Sys to acquire the desired data. The sixth and seventh steps represent the transactions that need to be conducted during the data exchange process.

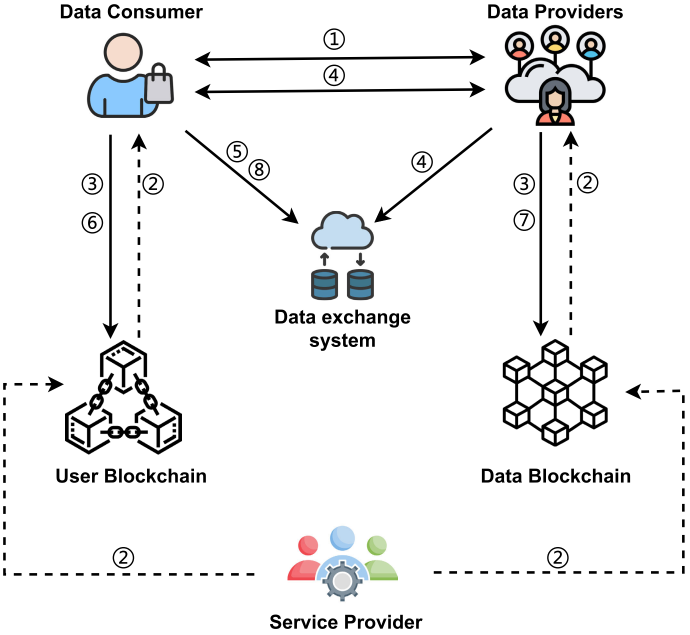
B. THREAT MODEL

We establish the security assumptions and threat model that will be utilized to assess the susceptibility of the scheme to attacks. Based on general security assumptions, we make the following security assumptions.

- Both parties involved in a transaction can be dishonest.
- Off-chain service providers are semi-honest, and without loss of generality, we assume that malicious attackers cannot control more than one-third of the members in SP.
- In a short period, the exchange rate of coins on blockchain will not drop sharply, which is used to prevent honest participants from losing out when the challenge stage is performed. In reality, a sharp decline in exchange rates is rare.

We present an analysis of the potential threats that our schemes may encounter, encompassing attack actions from both malicious participants and untrustworthy SP.

- **Malicious participants.** As participants engaged in transactions within the blockchain model, both C and P faced potential security issues, including the risk of engaging in malicious activity. To achieve secure and efficient data atomic exchange in the cross-chain process, a primary concern is to counteract fraudulent cross-chain transactions, such as double-spending attacks, initiated by malicious users. One intuitive approach involves the participant invoking a pre-deployed smart contract during the transaction request stage, enabling the smart contract to enforce the agreed-upon data exchange between the involved parties. However, in the case of asset atomic swaps where transaction confirmation is not required, the participant can deplete the account balance before the



① Negotiate the transaction deposit.
② Provide members' public key sets:{pk_1,...,pk_n}.
③ Create a deposit account.
④ Negotiate data, transaction amount, store data, and generate data index and index hash.
⑤ Check data by index hash.
⑥ Call the contract to transfer assets.
⑦ Discover the transaction (contract transaction) and transfer data index and index hash.
⑧ Use the data index to redeem the data.

FIGURE 2. The system procedure.

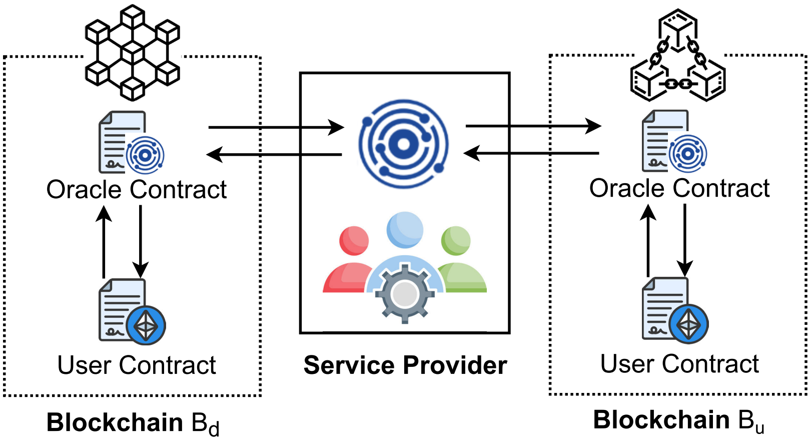


FIGURE 3. The cross-chain model.

contract is triggered. Consequently, during the official initiation of smart contract transfer, the account balance reflects negative assets, thus facilitating the success of the attack. Therefore, the foremost challenge lies in devising mechanisms to thwart the aforementioned attacks instigated by malicious participants, while safeguarding the property of the other participant.

- **Malicious service providers.** SP can be considered analogous to a full node, whereas its members function as light nodes. SP is typically regarded as a fully trusted entity in most schemes, this assumption does not hold in reality. The main concern is the potential presence of malicious members within the SP. Specifically, SP maintains an off-chain transaction pool that may be susceptible to the influence of malicious members. However, the emergence of malicious members will affect SP, and what SP expresses to the C_u is not the originally expected result of C_o, but a wrong verification result. Therefore, building a reliable service provider is the second challenge.

C. DESIGN GOALS

The program design goals were delineated based on the aforementioned system and threat model.

- Efficiency: The scheme should efficiently reduce the on-chain execution time of cross-chain transactions to satisfy the instantaneous data exchange demands of IoT devices. Furthermore, we present an evaluation of the experimental outcomes by providing experimental data.
- Malicious Behavior Resistance: The proposed scheme should possess the capability to effectively counter the risks outlined in Section IV-B.

V. THE PROPOSED SCHEME

In this section, we describe the complete cross-chain transaction process, which includes the setup, cross-chain transaction, and challenge and punishment stage. These stages are illustrated in Figure 4.

A. SETUP STAGE

Before initiating the transaction, certain preparatory measures must be undertaken:

1) **Target datasets and time-limit negotiation.** C shares the selected data with P, who stores the data in Sys. P further applies a one-way function calculation to the data identity to generate the corresponding hash value. P utilizes the hash value to associate the uploaded data within Sys and subsequently notifies C, allowing C to verify the presence of the data within Sys. Additionally, C and P engage in negotiations to determine the effective period, denoted as T, for publishing the transaction. Subsequently, C and P send the time flag T to SP, which then configures T within DDS. Specifically, C successfully publishes a cross-chain transaction tx_q on B_u. However, if P fails to publish the transaction tx_{1-q} to B_d within the specified time period T, SP will impose penalties on P.

2) **Create deposit accounts.** In the setup stage depicted in Figure 3, each member $sp_i \in$ SP, $i \in n$ of the notary selects polynomials f_i for B_u and B_d according to the following expression:

$$f_i(x) = \sum_{j=0}^{t-1} a_{i,j} \cdot x^j \bmod p \tag{2}$$

Here, t represents the threshold value and $a_{i,j}$ denotes a random number. Subsequently, each member sp_i of the notary broadcasts values $A_{i,k} = g_2 * a_{i,k} \bmod p$ for $k \in [t-1]$. In this context, p signifies the order of the BLS cyclic group G_2, Q_2 represents the generator, and $sk_{sp_i} = a_{i,0}$ and $pk_{sp_i} = A_{i,0}$ correspond to the private and public keys of sp_i respectively. These steps constitute the first stage, denoted as $Gen_{bls}(.)$ in Figure 3. Then, sp_i calculates the fragment $s_{i,j} = f_i(j) \bmod p, j \in [n]$, and transmits $s_{i,j}$ to sp_j via the Transport Layer Security protocol. Subsequently, member sp_j verifies the values of $A_{i,k}$ and $s_{i,j}$ using the verification formula described in Equation 3.

$$g_2 \cdot s_{i,j} = \sum_{k=0}^{t-1} A_{i,k} \cdot j^k \tag{3}$$

Member sp_j reconstructs secret key sk_{sp_j} using Equations 4 and 5 to obtain aggregated public key PK. Subsequently, PK is published on the blockchain and DDS, whereas threshold t is established within DDS. Finally, C and P utilize PK to generate the deposit accounts $addr_{DC}$ and $addr_{DP}$. These actions represent the second step of the setup stage, as illustrated in Figure 3.

$$sk_{sp_j} = \sum_{i=1}^{n} s_{i,j} \bmod p \tag{4}$$

$$PK = \sum_{i=1}^{n} pk_{sp_i} \tag{5}$$

3) **Lock up the deposits.** C and P jointly sign the transfer transaction amount to the deposit account on the blockchains as the deposit, constituting the third step of the setup stage, as illustrated in Figure 3. Once the deposit is locked by C and P, they inform SP of the completion of the lock and record the successful lock status in DDS. Subsequently, SP generates a corresponding Contract C_{sp} on the blockchain, which is responsible for verifying the correctness of the submitted transaction signature and inquiring about merkle-path proof associated with the transaction on the blockchain.

B. CROSS-CHAIN TRANSACTION STAGE

C verifies the completion of the setup stage by checking the status in DDS. Subsequently, C and P engaged in negotiations regarding information about data exchange and the transaction amount for a single transaction. C validates the accuracy of the data index provided by P, which is associated with C's identity information, against Sys. Upon successful verification, the cross-chain transaction stage between C and P commences.

C initiates cross-chain transactions with P by calling C_u. C_u initiates the computation of the deposit for the cross-chain transaction via the "calculate" function. Herein, the parameter "coef" denotes the multiplier employed in the deposit calculation process. Subsequently, C_u invokes C_o to access the public interface of SP and verify whether the deposit is adequate. Upon receiving a positive response, denoted by "true", C_u proceeds with a transfer operation. This initial step represents the first step in the cross-chain swap, as illustrated in Figure 3. Subsequently, P interacts with the relevant C_u on B_d to transmit the data index. This exchange constitutes the second step in the cross-chain swap depicted in Figure 3. The structure and function of C_u are described in Algorithm 1.

Algorithm 1 UserContract

```
Input: amount, address
Output: The result of transfer execution
  deposit=calculate(amount, coef)
  if OracleContract(deposit, msg.sender) then
      address.transfer()
      return true
  end if
  return false
```

Before officially each cross-chain transaction, it is imperative to verify the balance of the associated user's security deposit account who initiated the transaction. Hence, it is essential to design a global verification security deposit program. In blockchain systems, miners select transactions from the transaction pool to be included in blocks during

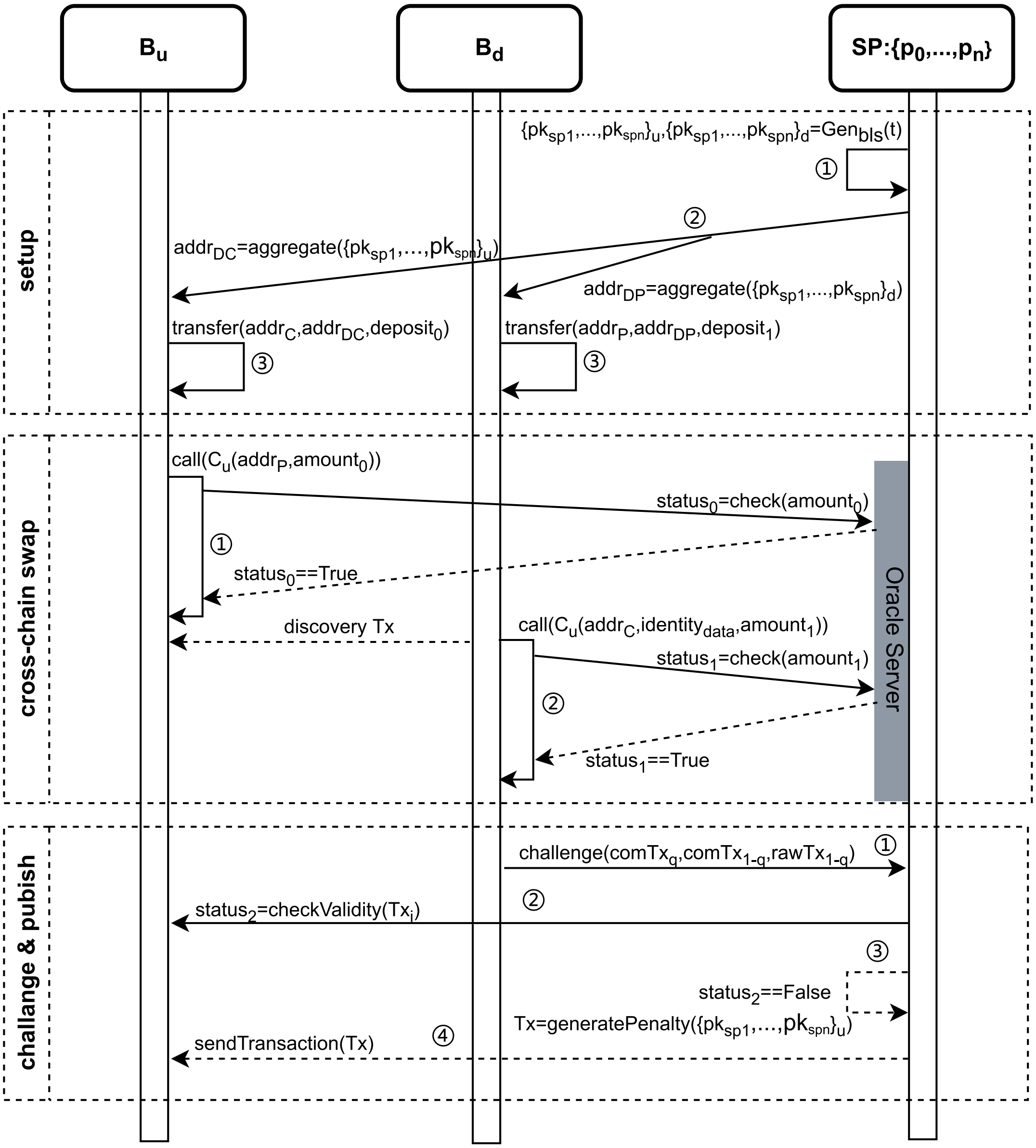

B_u
B_d
SP:$\{p_0,\ldots,p_n\}$
setup
cross-chain swap
challange & pubish
$\{pk_{sp1},\ldots,pk_{spn}\}_u,\{pk_{sp1},\ldots,pk_{spn}\}_d=Gen_{bls}(t)$
①
②
$addr_{DC}$=aggregate($\{pk_{sp1},\ldots,pk_{spn}\}_u$)
$addr_{DP}$=aggregate($\{pk_{sp1},\ldots,pk_{spn}\}_d$)
transfer($addr_C$,$addr_{DC}$,$deposit_0$)
transfer($addr_P$,$addr_{DP}$,$deposit_1$)
③
③
call(C_u($addr_P$,$amount_0$))
$status_0$=check($amount_0$)
①
$status_0$==True
discovery Tx
call(C_u($addr_C$,$identity_{data}$,$amount_1$))
$status_1$=check($amount_1$)
②
$status_1$==True
Oracle Server
challenge($comTx_q$,$comTx_{1-q}$,$rawTx_{1-q}$)
①
②
$status_2$=checkValidity(Tx_i)
③
$status_2$==False
Tx=generatePenalty($\{pk_{sp1},\ldots,pk_{spn}\}_u$)
④
sendTransaction(Tx)

FIGURE 4. Setup stage,cross-chain transaction stage and challenge stage.

the mining process. The transactions remain in the pool until they are incorporated into the longest valid chain. Leveraging this characteristic, we can query the deposits of $addr_{DC}$ and $addr_{DP}$. Nonetheless, it is undesirable to perform extensive backtracking of the blockchain for deposit verification. For instance, if a user executes a cross-chain transaction in a block situated near the genesis block, locating that specific block incurs substantial costs. Consequently, it is necessary to design a stateless program. Stateless programs can be implemented using smart contracts, however, traversing transactions within a transaction pool poses challenges. When the transaction volume is substantial, the resulting loop executions become exceedingly large, and smart contracts are subject to an upper limit on gas fees. Therefore, on-chain implementation of the program is not feasible.

Algorithm 2 Transaction Pool Query Interface

```
Input: $addr_C$, $addr_P$, amount
Output: The confirmation result of query
  transactions=getTransaction(from:$addr_C$,to:$addr_P$)
  balance=getDeposits($addr_{DC}$)
  if balance > sum(getBalances(transactions))+amount then
      address.transfer()
      return true
  end if
  return false
```

Based on the aforementioned conclusions, an open program was designed in SP to verify the legality of C_u requests from the blockchain. As depicted in Algorithm 2, the program operates as follows: The SP synchronously maintains the transaction pool on the blockchain and exposes this program to be invoked by the blockchain oracle, which returns the results corresponding to the transaction pool in the relevant state. Specifically, the program retrieves the cross-chain transactions associated with the identities of C and P from the transaction pool. It calculates the existing deposit amount by incorporating the deposit amount in the blockchain, resulting in a variable *sum*. This value is then compared with the balance in the security deposit account, denoted as *balance*, using the inequality $balance - sum > 0$. If the outcome of the interface is evaluated to be true, the transfer operation is executed and the contract execution result is returned as *true*, otherwise, the execution of C_u is terminated.

C. CHALLENGE AND PUNISHMENT STAGE

If any party submits an invalid transaction or fails to submit the transaction within the designated timeframe, the corresponding cross-chain transaction becomes inaccessible to the longest valid chain after the confirmation time T. Subsequently, the challenger initiates the challenge stage by transmitting $(comTx_b, comTx_{1-b}, rawTx_{1-b})_{b\in\{0,1\}}$ to SP for verification. Where $comTx_b$ represents the transaction signature of the challenger, $comTx_{1-b}$ denotes the transaction signature of the challenged party, and $rawTx_{1-b}$ refers to the original transaction of the challenged party. This represents the first step in the challenge stage, as illustrated in Figure 3.

Member sp_i initiates the retrieval of the Merkle path corresponding to $rawTx_{1-b}$ from the blockchain. If $rawTx_{1-b}$ is not found within the canonical chain, sp_i is unable to inquire conclusive evidence regarding the validity of $rawTx_{1-b}$ within the given time frame T. Consequently, SP proceeds with the generation of a penalty transaction. The subsequent procedures described represent the remaining stages of the challenge stage, as illustrated in Figure 3.

Member sp_i generates a sub-signature by signing the original penalty transaction rawTx_p and disseminates the signature to each node. The aggregate signature is then computed by SP using Equation 6, where H(.) represents the blockchain's hash function of the blockchain, and $L_i(.)$ denotes the Lagrangian basis function. To validate the aggregate signature, member sp_i verifies $\mathrm{Sig}_{all} = \mathrm{Sig}(0)$ using Equation 7. Ultimately, SP assigns the signature of rawTx_p as $\mathrm{Sig}_{all} = \mathrm{Sig}(0)$ and broadcasts it to the blockchain network.

$$Sig(x) = \sum_{i=1}^{n} sk_{sp_i} \cdot H(rawTx_p) \cdot L_i(x) \qquad (6)$$

$$e(Sig_{all}, Q_2) = e(H(rawTx_p), PK) \qquad (7)$$

VI. SECURITY ANALYSIS

In this section, we present a comprehensive security analysis of the proposed scheme. The scheme leverages smart contracts and cryptographic tools as defensive measures against the threats outlined in Section IV-B.

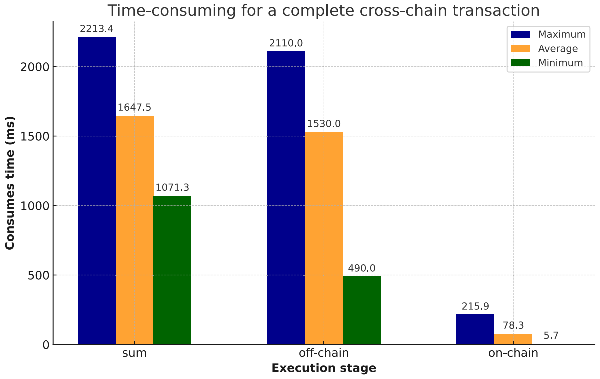


FIGURE 5. On-chain,off-chain and a complete cross-chain transaction time-consuming.

- **Malicious participants.** Both data consumers and providers can act maliciously. When data consumers intend to submit cross-chain transactions, they may employ double-spending attacks or negative assets to disrupt the proper execution of smart contracts. To mitigate these risks, we adopted an account-based approach that offers protection against double-spending attacks and facilitates transaction challenges. The challenge stage, which involves the execution process, is performed by an off-chain service provider. This design effectively protects against external interference. Furthermore, the challenge stage is open to all participants in the blockchain, enabling any user to submit transaction data and trace committed transactions. Failure to uphold the commitments of any consumer participant results in punitive measures. Similarly, data providers are obligated to fulfill their commitment to provide data to consumers. During the transaction challenge stage, data consumers can challenge the service provider, who will subsequently trace the submitted transaction. Failure to fulfill their commitment will undoubtedly lead to negative consequences.
- **Malicious service providers.** A service provider comprises numerous members, and the presence of malicious members within the service provider can result in erroneous outcomes from user-submitted transaction challenges. To counter this threat, we employ a combination of BLS signature and Shamir secret sharing. Drawing on the assumptions detailed in Section IV-B, it is presumed that the malicious attacker's control over the service provider's members does not exceed one-third. During the key-generation stage, members hold subkeys, whereas, during the challenge stage, they validate the submitted transaction. In the event of proven malicious behavior, a member generates a penalty transaction, signs it with a subkey to acquire a partial signature, and subsequently, all members successfully verify the aggregate signature and publish the transaction. This algorithm remains secure unless less than one-third of the total membership of the service provider is subject to manipulation.

TABLE 2. Cost of cross-chain transactions.

Number of users	Total cost per round (USD)	Average cost per round (USD)
10	1.105	0.110
200	21.305	0.107
500	54.803	0.109
1000	106.077	0.106
2000	213.091	0.106

VII. EVALUATION RESULTS

In this section, the effectiveness of our proposed scheme is evaluated based on the following three aspects: the efficiency of the complete cross-chain stage (including on-chain time and off-chain execution time), the cost of cross-chain operations, and the scalability of transactions that can be maintained by this program. Within the service provider environment, we utilize the HTTP interface to establish the interface outlined in Section V as the blockchain Oracle server. To demonstrate the viability of our scheme, we deployed a local Ethereum testnet on our server (CentOS 7.9, Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz) and a separate server (CentOS 7.9, Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz) hosting a local HyperLeger Fabric 1.4 testnet. Cross-chain transaction programs were developed using the Go programming language.

We evaluated the effectiveness of our methodology throughout the entire cross-chain data exchange process. Our experimental configuration included two transaction participants and a consortium of off-chain service providers. Considering the real-time nature of IoT data interactions, we conducted an in-depth analysis of the time needed for each cross-chain transaction and examined communication costs.

As shown in Figure 5, we analyzed the time consumption associated with on-chain transactions within the local construction network. The time consumption for each transaction involved in performing the quality assessment was recorded and averaged. The comprehensive time cost was measured from the moment the data consumer triggered the smart contract until the data consumer provided data identity. The average duration required for effective utilization of such cross-chain transactions ranged from 1071.3 to 1491.8 ms. The figure illustrates the comprehensive documentation of the time involved in the cross-chain transaction process. By leveraging the blockchain oracle [36], we outsourced the verification process to an off-chain service provider, thereby effectively reducing the on-chain execution time. Specifically, the average duration of cross-chain on-chain transactions ranged from 5.7 to 215.9 ms, while the average duration of cross-chain off-chain transactions ranged from 490 to 2110 ms. Consequently, the average overall cross-chain time fell within the range of 1071.3-2377.4 ms.

A. TRANSACTION FEES

In Table 2, the size of a cross-chain transaction exhibits a linear growth pattern concerning the number of participating users. As of June 13, 2023, the market value of one ETH amounted to $1,742.65. With a gas price set at 1 Gwei, the incurred cross-chain cost for an individual user is approximately $0.106 denominated in Ethereum. It is noteworthy that during the initial stage, wherein a sole pledging operation suffices to participate in multiple cross-chain exchanges, the option to extend the confirmation time can be considered as a trade-off to achieve reduced cross-chain costs.

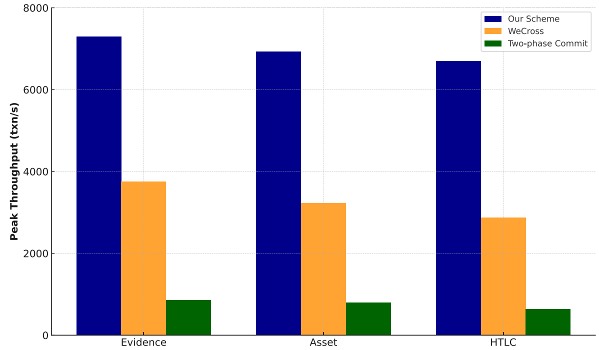


FIGURE 6. The peak throughput of three cross-chain contracts.

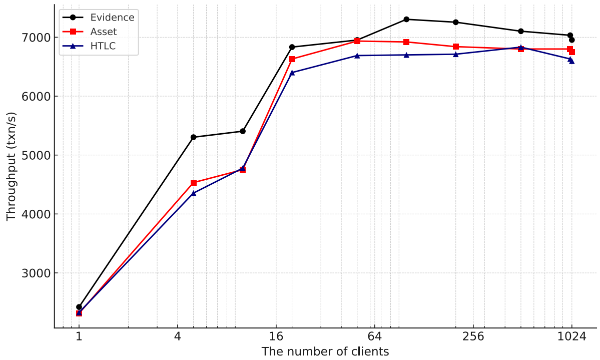


FIGURE 7. On-chain,off-chain and a complete cross-chain transaction time-consuming.

We compare our proposed scheme with WeCross from various perspectives, including scalability, execution latency, scheme cost, security assumptions, implementation complexity, and interoperability. For this evaluation, we instantiated the HyperLedger Fabric 1.4 testnet and WeCross latest platform on a server(CentOS 7.9, Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz), maintaining consistent configurations as per the default settings outlined in the WeCross white paper. Simultaneously, the Fisco 2.7 testnet was deployed on a separate server(CentOS 7.9, Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz). We compose and deploy smart contracts for three distinct scenarios within the blockchain network, including cross-chain asset exchange (Asset), data attestation (Evidence), and the execution of complex transaction hash time lock (HTLC) contracts.

1) **Evidence.** Electronic data attestation serves as a means of documenting the entire process of 'user authentication - data creation - storage - transmission.' When executing an evidence contract, results in the generation of two distinct transactions, involving the processes of writing and submitting transactions.
2) **Asset.** Asset exchange represents one of the most prevalent forms of transactions, forming the foundation

for numerous scenarios. When executing an asset transfer contract, a write transaction with a quantity of one is generated, and it includes a validation operation to check the sufficiency of account assets.

3) **HTLC.** HTLC is a complex contract commonly employed in cross-chain asset exchanges. In such exchanges, users employ this contract to facilitate the initiation of transaction processes by establishing locks within the contract and subsequently revealing keys to their respective counterparties. The execution of this contract within the Fisco Bcos blockchain entails a series of transactions, including "newProposal," "setNewProposalTxInfo," "setSecret," "lock," "setCounterpartyLockState," "unlock," "setCounterpartyUnlockState," and "deleteProposalID." Similarly, within the Hyperledger Fabric blockchain, the associated transactions involve "newProposal," "setNewProposalTxInfo," "lock," "setCounterpartyUnlockState," "unlock," and "deleteProposalID." These transactions are executed across both blockchain platforms in a cross-functional manner.

B. SCALABILITY

We employed Jmeter-5.4 to conduct performance testing and comparative analysis of the cross-chain execution stage, with a particular focus on evaluating the scalability effects of the proposed scheme. Throughput results serve as a key indicator of the scheme's performance. As part of the experimental procedure, we incrementally increased the user concurrency levels, with users sending transaction requests to SP over a local network. We start the timer when the participant sends a request and end when a contract outputs the final result. We disregard the first and last 10% of transactions and evaluate the stable performance. The prototype of our scheme successfully accommodated concurrent users numbering in the thousands, with the highest observed throughput stabilizing at a rate of over 7,000 requests per second. (Refer to Figure 6 for a visual representation of the achieved peak throughput.) According to Little's law [38], the throughput exhibited a linear increase during the initial stage, wherein the proposed scheme demonstrated its capability to support a minimum of 100 concurrent users. Subsequently, with the escalation in the number of concurrencies, the throughput of the proposed scheme reached an inflection point, leading to a decline. Throughout both periods, the throughput of the proposed scheme remained stable, accommodating up to approximately 1,000 concurrent users. These results exemplify the high-performance characteristics of our scheme, with peak throughput rates for the Evidence, Asset, and HTLC contracts reaching 7,304 txn/s, 6,920 txn/s, and 6,712 txn/s, respectively. Such performance levels effectively meet the data exchange demands of IoT platforms integrated with blockchain technology [20], [39]. Further escalation in the number of concurrencies would result in system overload, entering the oversaturation zone. The throughput of three contracts changes with the increasing amount of participants (cf. Figure 7).

TABLE 3. Cross-chain execution time in different scenarios.

Scenarios	WeCross(millisecond)	Our scheme(millisecond)
Evidence	4505	1303
Transmission	476	1303
Confirmation	4029	-
Asset exchange	3438	1718
Transmission	950	1718
Confirmation	2488	-
HTLC contract	32817	5323
Transmission	9286	5323
Confirmation	23531	-

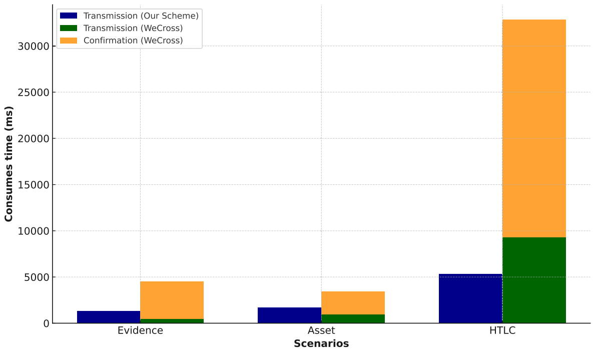


FIGURE 8. Cross-chain execution time in different scenarios.

Furthermore, our proposed scheme was subjected to comparative performance analysis in the same environment, pitted against WeCross and the Two-Phase Commit protocol. The conventional Two-Phase Commit protocol is widely employed in numerous frameworks, allowing for the non-blocking submission and execution of all selected transactions in the first phase, followed by result verification in the second phase [40]. In the case of WeCross and the Two-Phase Commit protocol, transaction confirmation in the first phase is contingent upon the consensus mechanisms of the underlying blockchains. Unfortunately, in scenarios involving complex transactions such as Evidence and HTLC, longer confirmation times are incurred when different transactions generated by contracts are included in separate blocks. For instance, in the case of the Evidence scenario, contract execution necessitates the generation of two transactions - the initiation transaction and the submission transaction - resulting in distinct confirmation times. When these two transactions are confirmed in separate blocks, the overall confirmation time is extended. In contrast, our proposed scheme shifts this security guarantee to the challenge and punishment stage, thereby reducing the time required for confirmation. In terms of throughput performance, our scheme outperforms the aforementioned alternatives.

C. EXECUTION LATENCY

The proposed scheme necessitates a conventional blockchain transaction, namely a deposit set up, during the setup stage, incurring time costs. Notably, during this stage, participants are required to engage in a single deposit action, allowing them to partake in multiple cross-chain exchanges. This approach significantly enhances efficiency in terms of both time and cost. During the challenge-punishment stage, when the proposed scheme successfully triggers and generates a penalty transaction, it incurs the time cost of a standard blockchain transaction. However, the incurred penalties are borne by the discredited party. Given that adversaries are rational, the frequency of triggering penalty transactions is anticipated to be low.

In Table 3 and 8, we present the aggregate transaction confirmation durations for the two schemes across three distinct scenarios. This includes the time it takes for a user to submit a transaction for execution on the blockchain, as well as the duration required for the final transaction to be confirmed by the blockchain.

1) **Evidence.** In the Evidence scenario, the execution of Evidence smart contracts results in two transactions: ‘‘write’’ and ‘‘submit.’’ The proposed scheme allows for the negotiation and verification of the total transaction amount during the setup stage, with blockchain Oracle logs providing direct access to verification outcomes, thus enabling batch transaction amount verification. Although the proposed scheme's transmission delay slightly exceeds that of WeCross transactions, the presence of two transactions in different blocks within the Evidence scenario doubles the confirmation delay for WeCross. According to the Evidence column in Table 3, WeCross transactions require an average of 4505 ms, including an average transmission delay of 476 ms (with ‘‘write’’ and ‘‘submit’’ operations averaging 271 ms and 205 ms, respectively), and an average blockchain transaction confirmation delay of 4029 ms. The proposed scheme completes transactions in an average of 1303 ms, encompassing both transmission delay and amount verification delay, with an average transmission delay of 424 ms (including ‘‘write’’ and ‘‘submit’’ operations averaging 283 ms and 141 ms, respectively), and an average deposit verification delay of 879 ms.
2) **Asset.** In the asset exchange process facilitated by smart contracts, a single ‘‘asset transfer’’ transaction is generated. The proposed scheme, during its execution, necessitates an amount verification operation, resulting in a transmission delay slightly exceeding that of the WeCross transmission delay. According to Table 3 under the Asset exchange column, WeCross transactions take an average of 3438 ms to complete, with an average transmission delay of 950 ms and an average blockchain transaction confirmation delay of 2488 ms. The proposed scheme averages 1718 ms to complete the smart contract, including transmission and amount verification delays, with an average transmission delay of 733 ms and an average deposit verification delay of 985 ms.
3) **HTLC.** In the HTLC scenario, the execution of HTLC smart contracts generates a series of transactions, divided into ‘‘set lock’’ operations (including ‘‘newProposal,’’ ‘‘setNewProposalTxInfo,’’ ‘‘setSecret,’’ ‘‘lock,’’ ‘‘setCounterpartyLockState’’) and ‘‘release lock’’ operations (including ‘‘unlock,’’ ‘‘setCounterpartyUnlockState,’’ ‘‘deleteProposalID’’), with specific triggering conditions. The proposed scheme negotiates transaction triggers and deposit verification at the setup stage, resulting in shorter transmission delays compared to sequential WeCross activations. WeCross faces extended confirmation delays in complex transactions involving multiple exchanges. According to the HTLC contract section in Table 3, WeCross transactions average 32817 ms, with a transmission delay of 9286 ms (6037 ms for ‘‘set lock’’ and 3249 ms for ‘‘release lock’’), and a blockchain transaction confirmation delay of 23531 ms. The proposed scheme averages 5323 ms, with a transmission delay of 3837 ms (2516 ms for ‘‘set lock’’ and 1321 ms for ‘‘release lock’’), and a pledge amount verification delay of 1486 ms.

From the execution delay analysis of the three transaction scenarios, the main source of WeCross time consumption is transaction confirmation. In contrast, the time costs for the proposed scheme are attributed to the necessary execution of smart contracts and deposit verification. Importantly, these costs are significantly lower than those incurred from transaction confirmations, illustrating the effectiveness of the proposed scheme in reducing operational delays.

D. SCHEME COSTS

The transaction costs in our proposed scheme originate from the collateralization process during the setup stage, which necessitates users stake an amount equal to or greater than the assets corresponding to the intended transaction. This mechanism provides security assurance for user payments. It is noteworthy that the collateralization process for a single cross-chain transaction can support multiple cross-chain operations by the participating parties. WeCross employs hash time lock technology to enhance security when handling cross-chain asset exchanges. This technology requires the locking of assets involved in a transaction, and for each contract deployment, relevant configurations within the contract are established through transaction submissions.

E. SECURITY ASSUMPTIONS AND IMPLEMENTATION COMPLEXITY

The security of this scheme is based on a service provider constructed using the threshold BLS signature scheme. In situations where an attacker cannot control several members exceeding the threshold, the protocol ensures that honest parties are safeguarded from economic losses through the process of challenge-response [41]. The distributed notary mechanism, which relies on this security feature, does not necessitate consideration of the underlying technical details of the blockchain. It can be universally applied across various homogeneous and heterogeneous chains, making its implementation relatively straightforward.

WeCross employs a two-phase commitment mechanism that relies on the consensus mechanisms of the underlying blockchains. In the first phase, the cross-chain routing of Blockchain 1 invokes the preparation of cross-chain assets on Blockchain 1, and the interaction between the cross-chain routers of Blockchain 1 and Blockchain 2 ensures the completion of the preparations on Blockchain 2. In the second phase, the cross-chain router first submits transaction requests to Blockchain 1, verifies the transaction confirmation from Blockchain 1, and then submits transaction requests to Blockchain 2. Moreover, WeCross abandons the traditional cross-chain process based on sidechain SPV proofs and resets the underlying routing. When executing cross-chain transactions, WeCross utilizes a two-phase commitment protocol, which divides the transaction into a voting phase and a submission phase, encompassing interfaces for preparation, submission, and rollback. The design of cross-chain network routing in WeCross, along with the necessity to design contracts according to the two-phase commitment protocol, leads to increased complexity, directly related to factors such as the applicable chains and the scale of the project. In comparison to the implementation and deployment challenges of our proposed scheme, WeCross presents a higher level of complexity.

F. INTEROPERABILITY

Interoperability in blockchain primarily addresses whether the semantics of specific applications can transcend different blockchains. Our proposed scheme is compatible with all blockchains that support smart contracts. Achieving compatibility between WeCross and native blockchains requires the development of adapters. Currently, the WeCross official support for adapters is limited to Fisco Bcos and HyperLedger Fabric.

VIII. CONCLUSION

In this paper, we present an efficient and robust cross-chain transaction scheme that facilitates real-time data exchange for IoT applications on a blockchain. We've implemented a distributed notary for IoT heterogeneous chain networks' data exchange services using threshold BLS signatures, reducing reliance on third parties. Furthermore, we employed a blockchain oracle machine to transfer the on-chain verification processes to off-chain execution. In addition, we propose a pledge-based cross-chain atomic exchange scheme to enhance the security and immediacy of IoT data exchange. Experiments show transactions complete within 1071.3-2377.4 milliseconds, with throughput peaking at over 7000 requests per second across various transaction scenarios, importantly without consensus mechanism-induced delays during execution.

This paper primarily addresses the performance issues of IoT schemes integrated with blockchain concerning the impact of consensus mechanisms. The proposed scheme aims to mitigate these issues. The next steps of our work are as follows: In our scheme, we introduce distributed notaries. In certain IoT scenarios, such as finance, data security, and outsourced computation, the parties involved in transactions on the blockchain may not wish to disclose their transaction information to third-party notaries. Therefore, our next endeavor is to enhance the privacy aspects of our scheme by incorporating cryptographic tools. This enhancement will encompass features such as transaction unlinkability, transaction confidentiality, and the assurance of both immediacy and security in transactions.

REFERENCES

[1] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by Internet of Things," *Trans. Emerg. Telecommun. Technol.*, vol. 25, no. 1, pp. 81–93, Jan. 2014.

[2] T. N. Qureshi, N. Javaid, A. Almogren, A. U. Khan, H. Almajed, and I. Mohiuddin, "An adaptive enhanced differential evolution strategies for topology robustness in Internet of Things," *Int. J. Web Grid Services*, vol. 18, no. 1, pp. 1–33, 2022.

[3] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town crier: An authenticated data feed for smart contracts," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 270–282.

[4] N. K. Tran, M. Ali Babar, and J. Boan, "Integrating blockchain and Internet of Things systems: A systematic review on objectives and designs," *J. Netw. Comput. Appl.*, vol. 173, Jan. 2021, Art. no. 102844.

[5] L. Li, T. Zhang, G. Sun, D. Jin, and N. Li, "A fair, verifiable and privacy-protecting data outsourcing transaction scheme based on smart contracts," *IEEE Access*, vol. 10, pp. 106873–106885, 2022.

[6] L. Li, D. Jin, T. Zhang, and N. Li, "A secure, reliable and low-cost distributed storage scheme based on blockchain and IPFS for firefighting IoT data," *IEEE Access*, vol. 11, pp. 97318–97330, 2023.

[7] J. Robert, S. Kubler, and S. Ghatpande, "Enhanced lightning network (off-chain)-based micropayment in IoT ecosystems," *Future Gener. Comput. Syst.*, vol. 112, pp. 283–296, Nov. 2020.

[8] J. Meijers, G. D. Putra, G. Kotsialou, S. S. Kanhere, and A. Veneris, "Cost-effective blockchain-based IoT data marketplaces with a credit invariant," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, 2021, pp. 1–9.

[9] S. Wan and S. Goudos, "Faster R-CNN for multi-class fruit detection using a robotic vision system," *Comput. Netw.*, vol. 168, Feb. 2020, Art. no. 107036.

[10] S. Wan, Z. Gu, and Q. Ni, "Cognitive computing and wireless communications on the edge for healthcare service robots," *Comput. Commun.*, vol. 149, pp. 99–106, Jan. 2020.

[11] L. Qi, X. Zhang, S. Li, S. Wan, Y. Wen, and W. Gong, "Spatial–temporal data-driven service recommendation with privacy-preservation," *Inf. Sci.*, vol. 515, pp. 91–102, Apr. 2020.

[12] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified locality-sensitive hashing-based recommender systems with privacy protection," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 14, p. e5681, Jun. 2022.

[13] N. S. Bitcoin, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[14] W. Lin et al., "A blockchain-enabled decentralized settlement model for IoT data exchange services," *Wireless Netw.*, 2020, doi: 10.1007/s11276-020-02345-9.

[15] R. G. Brown, "The corda platform: An introduction," *Retrieved*, vol. 27, p. 2018, May 2018.

[16] L. Moreau, "Stability of continuous-time distributed consensus algorithms," in *Proc. 43rd IEEE Conf. Decis. Control (CDC)*, vol. 4, 2004, pp. 3998–4003.

[17] V. Gugueoth, S. Safavat, S. Shetty, and D. Rawat, "A review of IoT security and privacy using decentralized blockchain techniques," *Comput. Sci. Rev.*, vol. 50, Nov. 2023, Art. no. 100585.

[18] Z. Abubaker, A. U. Khan, A. Almogren, S. Abbas, A. Javaid, A. Radwan, and N. Javaid, ‘‘Trustful data trading through monetizing IoT data using BlockChain based review system,’’ *Concurrency Comput., Pract. Exper.*, vol. 34, no. 5, Feb. 2022, Art. no. e6739.

[19] S. Shao, F. Chen, X. Xiao, W. Gu, Y. Lu, S. Wang, W. Tang, S. Liu, F. Wu, J. He, Y. Ji, K. Zhang, and F. Mei, ‘‘IBE-BCIOT: An IBE based cross-chain communication mechanism of blockchain in IoT,’’ *World Wide Web*, vol. 24, no. 5, pp. 1665–1690, Sep. 2021.

[20] H.-N. Dai, Z. Zheng, and Y. Zhang, ‘‘Blockchain for Internet of Things: A survey,’’ *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.

[21] W. Ou, S. Huang, J. Zheng, Q. Zhang, G. Zeng, and W. Han, ‘‘An overview on cross-chain: Mechanism, platforms, challenges and advances,’’ *Comput. Netw.*, vol. 218, Dec. 2022, Art. no. 109378.

[22] E. K. Wang, Z. Liang, C.-M. Chen, S. Kumari, and M. K. Khan, ‘‘PoRX: A reputation incentive scheme for blockchain consensus of IIoT,’’ *Future Gener. Comput. Syst.*, vol. 102, pp. 140–151, Jan. 2020.

[23] R. Huang, X. Yang, and P. Ajay, ‘‘Consensus mechanism for software-defined blockchain in Internet of Things,’’ *Internet Things Cyber-Phys. Syst.*, vol. 3, pp. 52–60, Jun. 2023.

[24] R. G. Sonkamble, S. P. Phansalkar, V. M. Potdar, and A. M. Bongale, ‘‘Survey of interoperability in electronic health records management and proposed blockchain based framework: MyBlockEHR,’’ *IEEE Access*, vol. 9, pp. 158367–158401, 2021.

[25] Y. Hei, D. Li, C. Zhang, J. Liu, Y. Liu, and Q. Wu, ‘‘Practical AgentChain: A compatible cross-chain exchange system,’’ *Future Gener. Comput. Syst.*, vol. 130, pp. 207–218, May 2022.

[26] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, and Y. Manevich, ‘‘Hyperledger fabric: A distributed operating system for permissioned blockchains,’’ in *Proc. 13th EuroSys Conf.*, 2018, pp. 1–15.

[27] S. Mazumdar, ‘‘Towards faster settlement in HTLC-based cross-chain atomic swaps,’’ in *Proc. IEEE 4th Int. Conf. Trust, Privacy Secur. Intell. Syst., Appl. (TPS-ISA)*, Dec. 2022, pp. 295–304.

[28] S. Imoto, Y. Sudo, H. Kakugawa, and T. Masuzawa, ‘‘Atomic cross-chain swaps with improved space, time and local time complexities,’’ *Inf. Comput.*, vol. 292, Jun. 2023, Art. no. 105039.

[29] A. Shamir, ‘‘How to share a secret,’’ *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[30] C. Yuan, M. Xu, and X. Si, ‘‘Optimization scheme of consensus algorithm based on aggregation signature,’’ *Comput. Sci.*, vol. 45, no. 2, pp. 53–56, 2018.

[31] D. Boneh, B. Lynn, and H. Shacham, ‘‘Short signatures from the Weil pairing,’’ in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Nov. 2001, pp. 514–532.

[32] R. Bacho and J. Loss, ‘‘On the adaptive security of the threshold BLS signature scheme,’’ in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2022, pp. 193–207.

[33] G. Wood, ‘‘Ethereum: A secure decentralised generalised transaction ledger,’’ *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.

[34] J. Teutsch and C. Reitwießner, ‘‘Truebit: A scalable verification solution for blockchains,’’ 2018.

[35] A. Pasdar, Y. C. Lee, and Z. Dong, ‘‘Connect API with blockchain: A survey on blockchain Oracle implementation,’’ *ACM Comput. Surveys*, vol. 55, no. 10, pp. 1–39, Oct. 2023.

[36] R. Mühlberger, S. Bachhofner, E. C. Ferrer, C. Di Ciccio, I. Weber, M. Wöhrer, and U. Zdun, ‘‘Foundational Oracle patterns: Connecting blockchain to the off-chain world,’’ in *Business Process Management: Blockchain and Robotic Process Automation Forum: BPM 2020 Blockchain and RPA Forum*. Seville, Spain: Springer, 2020, pp. 35–51.

[37] A. Zamyatin, M. Al-Bassam, D. Zindros, E. Kokoris-Kogias, P. Moreno-Sanchez, A. Kiayias, and W. J. Knottenbelt, ‘‘Sok: Communication across distributed ledgers,’’ in *Proc. Financial Cryptography Data Security: 25th Int. Conf.* Berlin, Germany: Springer, Mar. 2021, pp. 3–36.

[38] J. D. Little and S. C. Graves, ‘‘Little’s law,’’ in *Building Intuition: Insights From Basic Operations Management Models and Principles*. Springer, 2008, pp. 81–100.

[39] S. Zafar, K. Bhatti, M. Shabbir, F. Hashmat, and A. H. Akbar, ‘‘Integration of blockchain and Internet of Things: Challenges and solutions,’’ *Ann. Telecommun.*, vol. 77, no. 4, pp. 1–20, 2022.

[40] T. Dickerson, P. Gazzillo, M. Herlihy, and E. Koskinen, ‘‘Adding concurrency to smart contracts,’’ in *Proc. ACM Symp. Princ. Distrib. Comput.*, Jul. 2017, pp. 303–312.

[41] K. Wang, Z. Zhang, and H. S. Kim, ‘‘ReviewChain: Smart contract based review system with multi-blockchain gateway,’’ in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2018, pp. 1521–1526.

ZHIKAI LIN was born in Guangdong, China, in 2000. He is currently pursuing the master’s degree in cyber security with Jinan University, Guangdong. His research interests include blockchain and the IoT security.

KEXING WANG received the M.S. and Ph.D. degrees from the College of Information Science and Technology, Jinan University. His research interests include blockchain, network security, and applied cryptography.

YONGDONG WU received the B.Eng. and M.S. degrees from Beihang University, the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, and the master’s degree in management of technology from the National University of Singapore. He is currently a Senior Scientist with the Infocomm Security Department, Institute of Infocomm Research, Agency for Science Technology, and Research (A*STAR), Singapore. He is also an Adjunct Associate Professor with Singapore Management University. He is with the National Joint Engineering Research Center for Network Security Detection and Protection Technology, Guangdong Key Laboratory of Data Security and Privacy Preserving, Guangdong-Hong Kong Joint Laboratory for Data Security and Privacy Preserving, Jinan University Guangzhou, China. He has published over 100 articles and seven patents. His research interests include multimedia security, cyber-physical system security, the IoT security, and network security. His research results and proposals was incorporated in the ISO/IEC JPEG 2000 Security Standard 15444-8, in 2007. He was a recipient of the Best Paper Award of the IFIP Conference on Communications and Multimedia Security 2012.

DINGCHENG LI received the M.S. degree from Jinan University. His research interests include blockchain security and zero-knowledge proof.