EquiLink Bridge: A Semi-Custodial Approach to Cross-Chain Transactions via TEE

Tingda Shen, Yebo Feng, Jin Dong, Konglin Zhu, Lei Jiao, and Lin Zhang

***Abstract*—The rising demand for blockchain interoperability is accelerating advancements in cross-chain bridge technologies, which are crucial for a seamless information transfer in multi-blockchain ecosystems. Existing blockchain bridges are typically classified into two categories: custodial and non-custodial. Custodial bridges use a trusted third party for easier and faster transactions but depend on custodian trust, while non-custodial bridges enhance transparency and control with smart contracts but increased complexity and latency. Currently, no bridge design successfully combines the benefits of both while avoiding their drawbacks. This paper presents EquiLink, a semi-custodial bridge that combines the benefits of both custodial and non-custodial methods. EquiLink employs a smart contract, known as the EquiLink Service, to initiate cross-chain transfers. It then uses the EquiLink Network, a system composed of remote-attested Trusted Execution Environments (TEEs), to verify and issue these transfers between two blockchains. Any eligible participants validated through remote attestation can join the EquiLink Network and contribute to the bridge's functionality. Additionally, participants are regulated by an economic model, providing an extra layer of security through economic incentives. This semi-custodial bridge enhances transparency and control for users. Meanwhile, it mitigates the risks associated with centralized custody and decentralization. In the evaluation, EquiLink is resilient against both replay and physical attacks. Additionally, it operates efficiently, reducing transaction costs by 14.1% and latency by 18.9%**

***Index Terms*—Blockchain, cross-chain bridge, hash time-lock contract, semi-custodial model, trusted execution environment.**

Received 19 November 2025; revised 31 March 2026; accepted 5 May 2026. Date of publication 11 May 2026; date of current version 12 June 2026. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFB2704500, in part by the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, and in part by the Beijing Key Laboratory of Multimodal Data Intelligent Perception and Governance. *(Corresponding authors: Yebo Feng; Konglin Zhu.)*

Tingda Shen and Lin Zhang are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: shentingda@bupt.edu.cn; zhanglin@bupt.edu.cn).

Yebo Feng is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: yebo.feng@ntu.edu.sg).

Jin Dong is with the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beijing 100086, China (e-mail: dongjin@baec.org.cn).

Konglin Zhu is with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Beijing Key Laboratory of Multimodal Data Intelligent Perception and Governance, Beijing 100876, China (e-mail: klzhu@bupt.edu.cn).

Lei Jiao is with the Center for Cyber Security and Privacy, University of Oregon, Eugene, OR 97403 USA (e-mail: ljiao2@uoregon.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSC.2026.3691985, provided by the authors.

Digital Object Identifier 10.1109/TSC.2026.3691985

I. Introduction

As BLOCKCHAIN ecosystems undergo rapid growth, CoinGecko data indicates that 242 blockchain networks surpassed $86.9 billion in Total Value Locked (TVL) by April 2025 [1]. This expansion has led to the emergence of heterogeneous blockchain platforms with distinct characteristics and functionalities, forming a multi-chain ecosystem [2]. Consequently, the need for interoperability and seamless value transfer has become increasingly critical, making cross-chain bridges more crucial than ever [3]. Cross-chain bridges establish connections between independent blockchains, enabling decentralized applications (dApps) to operate across heterogeneous networks by facilitating both asset and data transfers [4], [5]. As a fundamental technology for blockchain interoperability, cross-chain bridges serve as the backbone of the multi-chain ecosystem, driving the global digital economy.

Cross-chain bridges are typically classified into custodial and non-custodial types [6]. Custodial bridges rely on a trusted third party to manage user assets and facilitate transfers, which usually improves efficiency and simplifies transaction coordination, but also introduces dependence on the custodian and potential single points of failure [7]. In contrast, non-custodial bridges eliminate trusted intermediaries and instead rely on consensus protocols, smart contracts, and cryptographic techniques to enable trustless cross-chain transactions [7]. While such designs improve transparency and user control, their reliance on decentralized verification and coordination often increases system complexity and reduces transfer efficiency [3].

The two prevailing types of cross-chain bridges exhibit inherent limitations. Custodial bridges prioritize easier and faster transactions, yet they suffer from inherent transparency limitations in asset custody and remain exposed to single points of failure, while non-custodial bridges offer enhanced transparency and user control [6], [8], [9], yet they often suffer from increased system complexity and reduced efficiency. Current solutions generally favor different trade-offs, making it difficult to design a bridge that simultaneously meets the requirements of efficiency, decentralization, and security. To address this challenge, semi-custodial bridges have emerged as a promising architectural paradigm that combines on-chain asset custody with trusted off-chain execution or verification components. In such designs, assets remain locked in smart contracts on the blockchain, while certain coordination or verification tasks are delegated to trusted entities or specialized execution environments. This hybrid trust model enables semi-custodial bridges to balance the trade-offs between efficiency, security, and decentralization.

This paper aims to balance efficiency, decentralization, and security by proposing a novel semi-custodial cross-chain bridge scheme called the EquiLink Bridge. EquiLink leverages trusted

1939-1374 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies.
Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

hardware to secure off-chain execution, thereby reducing the high cost of on-chain operations. Concretely, EquiLink combines lightweight on-chain verification with decentralized governance of public participants equipped with Trusted Execution Environments (TEEs) to enhance both efficiency and decentralization. TEEs are isolated environments designed to ensure the confidentiality and integrity of workloads [10]. When combined with blockchains, TEEs enable efficient and confidential off-chain computation while the blockchain maintains consensus and verifiable state on-chain. Public participants equipped with TEEs can undergo remote attestation to join the network as *trusted participants*, who form the core of EquiLink's decentralized verification network. This network is responsible for securely executing and validating cross-chain transactions within their respective TEEs.

Specifically, EquiLink includes three main components: (i) the EquiLink Service ES, (ii) the Light Server LS, and (iii) the EquiLink Network EN. The ES, composed of smart contracts and InterPlanetary File System (IPFS), provides decentralized functionality for node routing, fund staking, and code distribution, enabling governance over cross-chain transactions and participants. The LS is responsible for authenticating TEE-based participants via remote attestation to establish a root of trust, enabling public participants to join the EquiLink Network. The EN, composed of trusted participants, forms a decentralized verification network in which participants may serve as validators for transaction verification or as watchtowers for detecting and penalizing misbehavior, thereby enabling non-custodial cross-chain transactions. For each transaction, a subset of trusted participants is randomly selected from the network to execute and verify the cross-chain operation. By combining hardware-backed trust with decentralized coordination, EquiLink realizes a semi-custodial cross-chain bridge.

Experimental evaluations illustrate that the proposed solution significantly improves transaction costs, transaction latency, and overall resource consumption, and confirms the effectiveness of its economic model and attack resistance. EquiLink reduces average transaction costs by 14.1%, achieving 289 k gas (optimistic) and 302 k gas (pessimistic), while decreasing transaction latency by 18.9% on average and 70% in high-congestion scenarios. The system's overall overhead illustrates that the entry and exit costs for participants in EquiLink can be amortized after validating only a small number of transactions. Furthermore, off-chain validation of 1,000 transactions is completed within 55 seconds on a 4-core, 16 GB machine, indicating that EquiLink imposes low hardware requirements and reduces the entry barriers for participants. In addition, a game-theoretic analysis confirms that the economic model can reach a stable equilibrium. Finally, we reproduced replay and physical attacks to verify the system's strong resilience against adversarial behaviors.

The primary contributions of this paper are as follows.

- *Semi-Custodial Bridge Solution:* We propose a semi-custodial cross-chain bridge that leverages a TEE-enabled validator network to support cost-efficient transfers across heterogeneous blockchains. By combining the strengths of custodial and non-custodial designs, the bridge reduces costs and latency through off-chain trusted execution, preserves transparency via decentralized services, and enhances security through a dual-hash locking mechanism coordinated by TEEs and watchtowers.
- *Efficient and Reliable Interoperability:* We design a decentralized EquiLink Network that distributes validation and monitoring across publicly accessible trusted participants, eliminating single points of failure and maintaining robustness under failures or adversarial conditions. A tailored protocol shifts resource-intensive verification to secure off-chain execution, thereby improving both efficiency and reliability of interoperability.
- *Sustainable Economic Model:* The system ensures sustainability through three mechanisms: *monetary incentives* rewarding validators via gas fees, *staking* penalizing misbehavior through stake confiscation, and a *watchtower mechanism* providing secondary verification, timeout recovery, and compensation.
- *Code release:* The code is openly available at [https://github.com/Stdwill/EquiLink-bridge], providing transparency and enabling further development and collaboration.

II. Related Work

This section introduces TEE-blockchain systems, classifies cross-chain mechanisms, reviews representative cross-chain bridge designs, and discusses the inherent limitations of both custodial and non-custodial approaches.

A. TEE-Blockchain Systems

TEE-blockchain systems refer to hybrid architectures that integrate blockchains with trusted execution environments, where blockchains provide consensus and persistent state while TEEs execute computations over private data off-chain. This design leverages the complementary properties of both technologies, enabling confidential and efficient execution while maintaining verifiable state on-chain. Ekiden [11] is the first system to propose a hybridized TEE-blockchain architecture, where computation is separated from consensus for smart contract execution. However, contracts in Ekiden are largely isolated and lack strong interoperability capabilities. Phala Network [12] extends this idea by building a distributed confidential computing network based on TEE workers to support cross-contract and cross-chain interactions. Liu et al. [8] propose a TEE-enabled framework that leverages trusted sensors and consistency protocols to securely upload off-chain data to the blockchain.

Although TEE–blockchain systems can combine the advantages of both technologies, such hybrid architectures may also introduce new security challenges compared with single-architecture designs. For instance, the execution of TEEs still relies on the underlying host system, and a malicious host may manipulate scheduling or even induce physical faults [10], potentially interrupting execution or causing inconsistent states. Moreover, the lack of a reliable trusted timer within TEEs makes it difficult to determine the freshness of blockchain states [11], thereby introducing additional challenges for state synchronization and secure verification.

B. Custodial Bridges

Custodial bridges rely on a trusted authority to facilitate asset transfers across blockchain networks. In such systems, users must trust a centralized operator to lock, mint, and distribute assets during the cross-chain transfer process [13]. Representative centralized bridges include cBridge [14], Across Protocol [15], Rhino.fi [16], Multichain [17], Binance Bridge [18], and Polybridge [19]. For example, Binance Bridge supports transfers between Binance and Ethereum by leveraging Binance's infrastructure to manage the process. Similarly, Rhino.fi

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

stores assets on different chains under the control of a central operator, enabling users to transfer assets by interacting with the operator's account during cross-chain transactions [20]. The xDAI Bridge [21] employs a lock-and-mint mechanism, using smart contracts to facilitate transfers from Ethereum to the Gnosis chain. In this system, assets are held by TokenBridgeDAO members, while xDAI tokens are minted and distributed through a multi-signature approval scheme.

While custodial bridges offer higher efficiency and lower latency, they also entail significant risks. A major concern is the inherent single point of failure, where centralized nodes can lead to catastrophic asset loss during hacks or regulatory actions [22]. Moreover, centralized operators hold the power to approve or reject transactions at their discretion, resulting in limited transparency. For example, the disappearance of Multichain's founder caused operational paralysis, highlighting the vulnerabilities of relying on a central authority for critical infrastructure [17], [22]. Furthermore, these bridges may face stricter regulatory scrutiny due to their resemblance to traditional financial institutions [23]. In contrast, EquiLink replaces the centralized operator in custodial bridges with a decentralized validator network, thereby avoiding the single point of failure inherent in custodial designs. This decentralized architecture ensures that transaction validation is no longer controlled by a single trusted entity, improving transparency and verifiability in cross-chain operations.

C. Non-Custodial Bridges

In contrast to custodial solutions, non-custodial bridges eliminate the need for trusted intermediaries by distributing trust across multiple independent nodes. They can be classified as Multi-Party Computation (MPC), Zero-knowledge proofs (ZKPs), payment-channel-based, or liquidity-pool-based approaches. Representative decentralized bridges include Rainbow Bridge [24], zkCross [25], zkBridge [26], XClaim [27], Stargate [28], Chainlink [29], and ALBA Cross-Chain Bridge [30]. For example, Multichain (formerly AnySwap) uses an MPC-based Threshold Signature Scheme (TSS) to enable transfers within 3–30 minutes [31], while zkBridge employs ZKPs to secure transactions, generating 128 signatures in 18 seconds with sub-2-minute delays [26]. Similarly, zkCross offers a two-layer solution to address cross-chain privacy and auditing challenges [25]. Horizon adopts a gas-efficient approach for trustless transactions without compromising decentralization [32], while ALBA Cross-Chain Bridge introduces Pay2Chain bridges that leverage off-chain payment channels [30].

Despite their advantages, non-custodial bridges typically incur higher communication costs and performance overheads due to the need for consensus among multiple nodes, increasing transaction complexity and latency [33]. Studies modeling cross-chain processes, such as queueing theory applications in the Cosmos network, provide insights into the trade-offs between system throughput and latency, further illustrating the challenges in optimizing non-custodial bridge performance [34]. Additionally, the complexity of non-custodial bridges also makes them more susceptible to vulnerabilities [6]. Several bridge-related exploits have been attributed to issues such as poor private key management and flaws in smart contract logic [6]. In contrast, EquiLink establishes trust for newly joined public participants through remote attestation and leverages TEE-based trusted participants during transaction validation. By doing so, it reduces reliance on gas-intensive on-chain consensus and verification. In this sense, EquiLink introduces TEEs into cross-chain verification as a design choice well suited to the verification bottleneck, thereby reducing on-chain overhead and improving efficiency and latency compared with existing approaches.

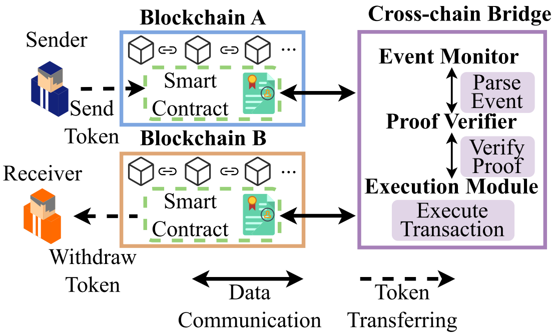


Fig. 1. The system framework of EquiLink.

III. System and Security Model

A. System Model

In a cross-chain bridge system, we consider several key roles and components. Fig. 1 illustrates the framework of a general cross-chain bridge. The bridge involves a set of blockchains, denoted as BC_s and BC_r, which are responsible for maintaining independent ledgers in cross-chain transactions. Users, acting as senders on BC_s and receivers on BC_r, initiate cross-chain transactions. A generic cross-chain bridge must incorporate three components to enable secure and reliable cross-chain transactions: an Event Monitor, a Proof Verifier, and an Execution Module. The Event Monitor continuously observes on-chain smart contracts to capture transaction-related events. The Proof Verifier authenticates the associated data to ensure that the reported transaction has indeed occurred on the source blockchain. Upon successful verification, the Execution Module performs the corresponding cross-chain operation on the destination blockchain, thereby completing the asset transfer.

EquiLink instantiates these three components using trusted participants equipped with TEEs. These participants are assigned to two distinct roles in each cross-chain transaction: validators v that capture transaction-related events and attest to the correctness of cross-chain transactions and watchtowers wt_i that monitor protocol execution and detect potential misbehavior. By leveraging TEEs, these participating nodes can provide integrity guarantees for critical operations. To enable interoperability, the system includes an on-chain coordination service with smart contracts that manages transaction requests and state synchronization between heterogeneous blockchains. Additionally, other trusted components, such as a Light Server, are leveraged to provide remote attestation to verify the correctness and trustworthiness of participants. The key notations used throughout the paper are summarized in Table III in Appendix, available online.

B. Threat Model

The objective of a cross-chain bridge is to correctly and securely complete each transaction from BC_s to BC_r. We

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

define the adversary $\mathcal{A}$ as an entity aiming to disrupt cross-chain transactions for financial gain or to degrade system availability. We assume that remote attestation is carried out by a set of trusted replicated attesters in the Light Server. A participant is admitted only when more than half of the attesters approve the attestation result, thereby proving the identity and code integrity of TEE-based participants. The EquiLink Service is assumed to provide the pre-defined cross-chain transaction services; however, it may be subject to exploitation by the adversary $\mathcal{A}$ due to potential smart contracts vulnerabilities.

We further assume that all cryptographic primitives used in the system are secure and that the underlying blockchains achieve consensus finality after transaction confirmations. We assume that the remote attestation mechanism can correctly verify the identity and integrity of honest TEEs, and that honest TEEs execute the intended code faithfully. However, we do not assume that all TEEs remain uncompromised. In practice, an adversary may compromise a subset of TEE-based participants or attempt to exploit side-channel leakage.

The adversary may include malicious senders or external attackers. Malicious senders may attempt to initiate double-spending transactions to obtain unauthorized assets. External attackers may disrupt the network or compromise a subset of TEE-based participants (including validators and watchtowers), gaining control over message delivery and potentially modifying, delaying, or dropping messages. They may also attempt to extract sensitive information through side-channel attacks, exploit vulnerabilities in smart contracts, or launch physical attacks against participating nodes. We assume that the adversary may compromise a subset of TEE-based participants but cannot compromise all TEEs involved in a given transaction simultaneously. In particular, the protocol remains secure as long as at least one participant in the decentralized verification group (DVG) executes the protocol correctly. Secure communication channels are assumed between honest participants. Based on the attacker's capabilities and goals, we consider the following attacks:

1) *Double-Spending Attacks:* A malicious sender issues two conflicting transactions t_1, t_2 on BC_s that spend the same input/nonce, attempting to bypass asset locking and mint duplicate assets on BC_r, violating value conservation.
2) *Replay Attacks:* A valid transaction t on BC_s is intercepted and resent to BC_r, triggering unauthorized asset issuance and breaking transaction uniqueness.
3) *Compromised Participants:* A corrupted or malicious participant p_{mal} may forge, drop, or equivocate on transactions, producing a tampered one $\tilde{t} = \mathcal{A}(t)$ or selectively relaying t, causing ledger divergence or false confirmations.
4) *Side-Channel Attacks:* An adversary may attempt to extract sensitive information from TEEs through side-channel attacks (e.g., cache timing or speculative execution leakage), potentially revealing secrets used in cross-chain transaction validation.
5) *Physical Faults:* Node crashes or disconnections interrupt transaction processing ($t_{\text{status}} = 1$), raising fund-recovery concerns for sender A_s.

C. Design Goals

An ideal cross-chain system should satisfy the following security properties:

- *Accurate and Secure:* The system must correctly and securely process cross-chain transactions, ensuring that assets are transferred with the exact amount to the intended recipient while preventing any unauthorized manipulation. It guarantees that each transaction executes atomically—either fully succeeds or fully fails—within a bounded time frame to avoid indefinite asset locking.
- *Incentive and Penalty Mechanism:* The system should align rewards and penalties to encourage honest participants and penalize malicious ones, so that following the protocol is always the most economically rational choice.
- *Efficient:* The system should minimize transaction latency and computational overhead by using TEEs to execute cross-chain logic off-chain while requiring only lightweight on-chain verification.
- *Low Cost:* The design should reduce on-chain gas consumption by shifting expensive computation and validation tasks to TEE-based off-chain participants. By relying on TEEs to guarantee correctness without full on-chain consensus, the system significantly lowers transaction fees, making cross-chain transfers economically viable.

IV. System Design

In this section, we present the detailed system description, including the overall architecture and the functionality of each component.

A. Overview

EquiLink achieves efficient, secure, and decentralized cross-chain interactions through off-chain computation with on-chain lightweight verification, TEE-based trust establishment, and the inclusion of public participants.

As presented in Fig. 2, EquiLink consists of three modules: EquiLink Service (ES), Light Server (LS), and EquiLink Network (EN). Specifically, (i) ES provides the on-chain support for cross-chain transfer through three core contracts: the router contract C_{router}, the staking contract C_{staking}, and the cross-chain transaction contract C_{tx}; (ii) LS provides third-party support for remote attestation and maintains a trusted router list stored on C_{router}; and (iii) EN is a set of trusted participants p forming a secure decentralized enclave network. This section presents the framework of EquiLink, while the protocol design and transaction workflow are detailed in Section V.

B. EquiLink Service

The EquiLink Service is the on-chain component of EquiLink, responsible for transaction routing, staking, and cross-chain transaction execution. It is implemented through three core smart contracts: C_{router}, C_{staking}, and C_{tx}. As a blockchain-based component, it provides a decentralized and transparent foundation for cross-chain services without relying on a single trusted operator.

1) Router Contract (C_{router}): This contract handles transaction routing, participant status management, and stores the IPFS address of the EquiLink source code for public verification. It provides the following functions:

- `router()`: Executes the routing process for transaction t using a Keccak-256-based random function, $DVG(t) \leftarrow \text{router}(A_s, A_r, t_v, \text{Nonce})$, where A_s, A_r, and

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

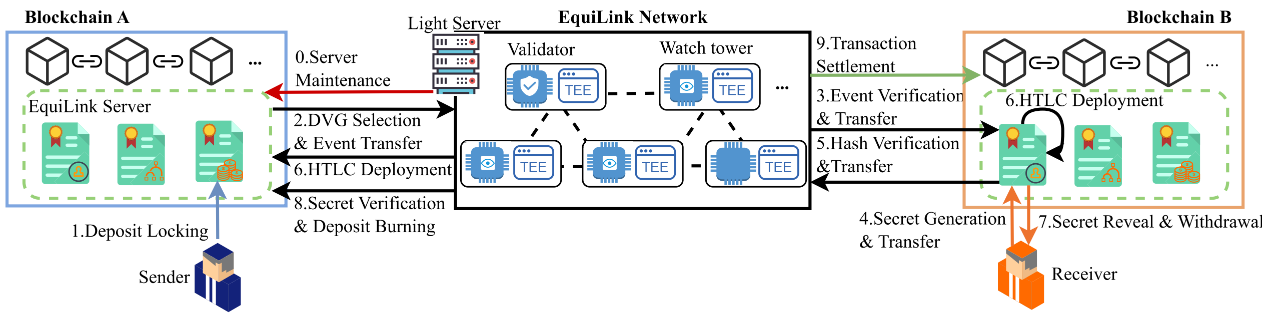


Fig. 2. The framework of EquiLink and the workflow of a cross-chain transaction, showing how the sender transfers tokens from Blockchain A to B.

t_v are transaction parameters, and Nonce represents uncontrollable randomness. The function outputs four active and eligible participants in EN to form $DVG(t) = \{v, wt_i, wt_j, wt_k\} \subseteq EN$. An event is emitted to record the generated DVG.

- `updateStatus()`: Updates the participant status $p_{\text{status}} \in \{0, 1\}$, ensuring that only active participants ($p_{\text{status}} = 0$) remain eligible for subsequent validation rounds.

2) Staking Contract ($C_{staking}$): This contract manages participants' staking balances p_{staking}. To participate in transaction validation, a participant must maintain sufficient stake such that $p_{\text{staking}} > t_{\text{fee}}$; only then can it join the transaction's DVG. The contract provides the following functions:

- `staking()`: Deposits t_{stake} into C_{staking}, increasing p_{staking} and making the participant eligible to join a DVG. An event is emitted to record the deposit.
- `withdraw()`: Withdraws t_{withdraw} from p_{staking}, provided no pending transactions restrict the withdrawal. The balance is updated accordingly, and an event is emitted to record the withdrawal.

3) Cross-Chain Transaction Contract (C_{tx}): This contract facilitates cross-chain transactions and provides the following functions:

- `transfer()`: Transfers tokens t_v into the contract, invokes `router()` in C_{router} to generate the DVG, and emits an event for cross-chain communication.
- `lock()`: Locks t_v in C_{tx} until the required conditions are satisfied.
- `mint()`: Mints t_v on BC_r after successful verification, allowing A_r to withdraw with the correct secret S.
- `burn()`: Burns t_v after finalization to maintain token-supply consistency and prevent double-spending.
- `createHTLC()`: Creates an HTLC for secure asset transfer.
- `checkTimeout()`: Checks whether transaction t has timed out and triggers rollback if necessary.

C. Light Server

The Light Server, as a third-party coordination service, with a set of trusted replicated attesters, provides a lightweight off-chain service to support the secure onboarding of public participants in the EquiLink bridge. Here, replicated attesters refer to a committee of attestation nodes that independently verify TEE attestation evidence during onboarding. They perform remote attestation of TEE-based participants and collectively maintain the trusted router list used by the router contract C_{router}. This mechanism ensures that only verified trusted participants can join the EquiLink Service, thereby maintaining the integrity of trusted participants through rigorous verification while preserving the decentralized nature of EquiLink.

To transform public participants into trusted participants, the system leverages a remote attestation mechanism. Each participant's TEE generates a verifiable attestation report (or "quote") that includes the measured hash of its execution environment, thereby demonstrating its integrity and authenticity. In addition, the process involves secure certificate management, endorsement verification, and compliance with predefined security policies. These features ensure that only participants with unaltered and properly configured TEE environments are admitted as trusted nodes in the network.

The attestation process begins when a participant launches its TEE-based application, which produces a local attestation report encapsulating the enclave measurement. The participant then sends this report to a remote attestation service (e.g., the Intel Attestation Service) for verification. Upon successful validation by the attestation authority, the signed report is submitted to the replicated attesters, which independently check it against the predefined security policies. A participant is enrolled as a trusted node in the EquiLink Network only when a majority of the replicated attesters approve the attestation result and jointly authorize its inclusion in the trusted router list through signatures to the router contract C_{router}. This multi-step process ensures that only genuine and uncompromised TEEs participate, thereby enhancing overall network security.

D. EquiLink Network

The EquiLink Network EN, composed of trusted participants, is designed to provide a secure decentralized network for cross-chain validation. To join the network, a participant must complete TEE remote attestation via the Light Server and stake tokens p_{staking}, thereby becoming a trusted participant $p \in EN$. This ensures that only verified and economically bonded participants contribute to cross-chain execution.

As shown in Fig. 3, trusted participants adopt a lightweight and stateless TEE architecture after joining the EquiLink Network. In this design, the enclave is only responsible for cross-chain transaction verification, while all system states and asset records remain stored on-chain. This ensures that TEEs do not replace blockchain consensus or state management, but only offload computationally expensive verification tasks.

Specifically, participants execute secure operations inside the **Trusted Execution Environment (TEE)**, including smart contract interactions and transaction verification. In the **Rich**

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

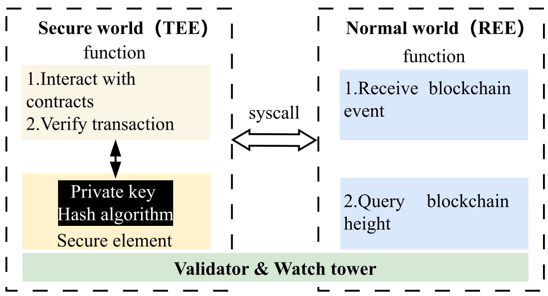


Fig. 3. The computing architecture of a trusted participant (built on a TEE).

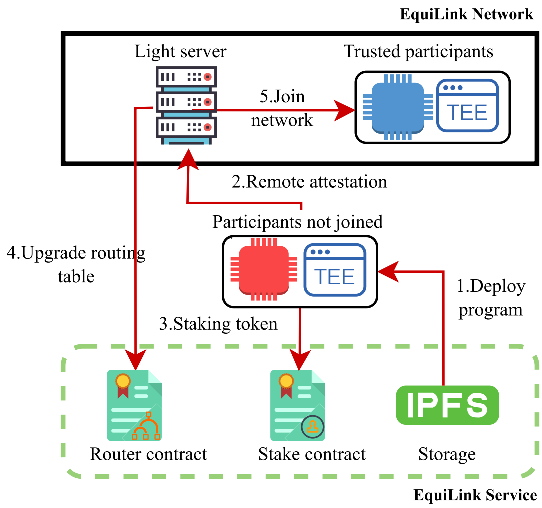


Fig. 4. The workflow of the initialization phase.

Execution Environment (REE), they monitor blockchain events and retrieve the current blockchain height for validation. This design preserves the blockchain trust model while improving efficiency.

When a transaction is initiated, the router contract C_{router} randomly selects four TEE-based participants from EN to form a DVG. Within each DVG, the validator v and watchtowers wt play different roles:

- *Validator* (v)*:* verifies the correctness of cross-chain transactions inside the TEE and submits the results via smart contracts. Formally, $v : t_{\text{status}} \rightarrow \{0, 1\}$, where t_{status} denotes the transaction status.
- *Watchtower* (wt)*:* monitors the transaction state t_{status}, detects anomalies, and enforces penalties via smart contracts. Formally, $wt : t_{\text{status}} \rightarrow \{0, 1\}$, where *Failure* indicates a timeout or irregular behavior.

V. Protocol Design

Section IV introduces the components and workflow of the EquiLink bridge. This section explains the EquiLink protocol using a representative cross-chain transaction example, as shown in Fig. 2. To ensure atomic and verifiable cross-chain transactions while minimizing trust assumptions, EquiLink adopts an HTLC-based protocol anchored by TEEs and DVG. The protocol consists of following three phases: (i) **Initialization Phase** (Fig. 4), where the replicated attesters in the Light Server authenticates TEE-based participants via remote attestation to establish trust and add the verified trusted participants to the trusted router list; (ii) **Transaction Phase** (Fig. 5): DVG, based on TEEs, coordinates secure and atomic token transfers across chains and applies rewards and penalties to participants according to the economic model. This phase consists of three key processes: locking, transaction, and settlement, supported by EquiLink Service; and (iii) **Exit Phase** (Fig. 6), where participants retrieve their stake, claim rewards, and safely exit the system.

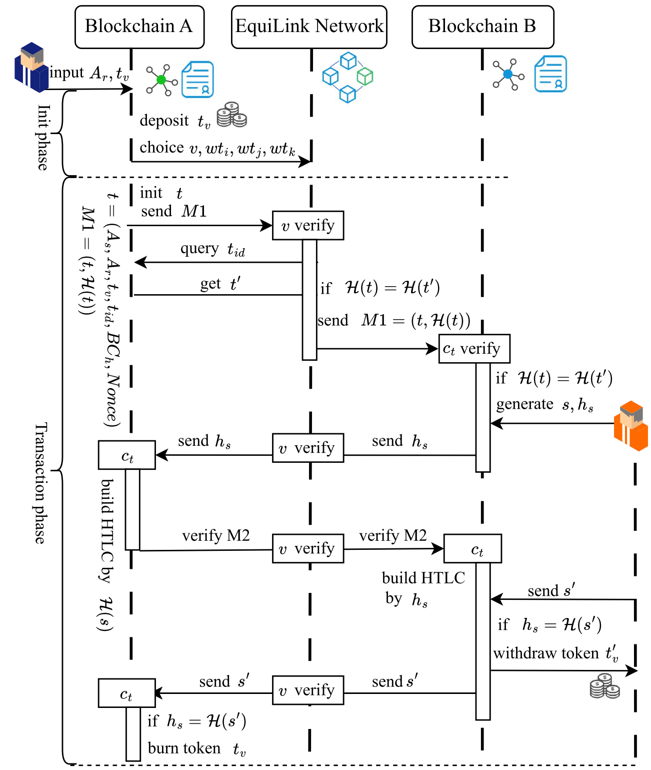


Fig. 5. The workflow of the transaction phase.

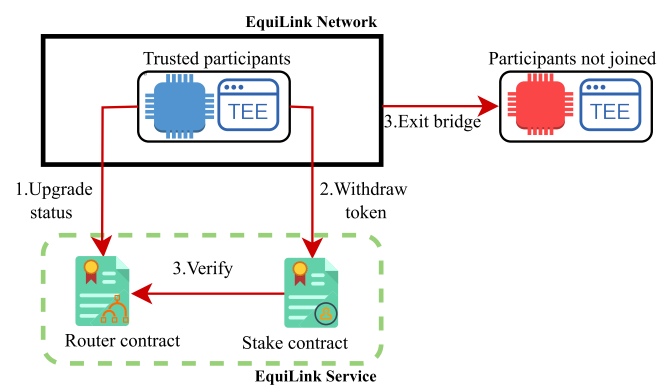


Fig. 6. The workflow of the exit phase.

A. Initialization Phase

In the proposed system, any participant built on a TEE can join the EquiLink Network for reward, thereby enhancing both participation and transparency. The initialization workflow is illustrated in Fig. 4. The process, which is jointly managed by the Light Server and EquiLink Service, consists of the following steps:

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

1) An untrusted participant retrieves and deploys the verification program from IPFS.
2) The participant submits a remote attestation request to the Light Server for verification. Once more than half of the replicated attesters approve the attestation result on-chain, the attestation is considered successful.
3) Upon successful attestation, the participant stakes tokens in the staking contract C_{staking}, ensuring that $p_{\text{staking}} > t_{\text{fee}}$ before proceeding.
4) The Light Server updates the router list in the router contract C_{router}, incorporating the new participant into the EquiLink Network.
The participant is officially registered in the system and can now be randomly assigned to a DVG through the `router()` function, $DVG(t) \leftarrow$ router $(A_s, A_r, t_v, \text{Nonce})$.

B. Transaction Phase

The transaction phase forms the core of the EquiLink bridge. To facilitate secure and atomic token transfers across chains, we design an efficient HTLC-based protocol enhanced with TEEs. This phase consists of three main processes: the *locking process*, the *transaction process*, and the *settlement process*. The protocol flow is illustrated in Fig. 5.

The overall process is as follows.

1) *Locking Process:*
   1) In this process, the user first authorizes the cross-chain transaction contract C_{tx} by calling approve $(A_s, C_{\text{tx}}, t_v)$, where A_s is the sender's address and t_v is the token amount.
   2) The user then initiates a cross-chain transaction by invoking the `transfer()` function of C_{tx} with parameters $\texttt{transfer}(A_r, BC_r, t_v)$, and the contract subsequently executes $\texttt{lock}(t_v)$, where A_r is the receiver's address, BC_r is the target blockchain. Within the `transfer()` function, C_{tx} calls the `router()` function of the routing contract C_{router} to randomly select a DVG $\{v, wt_i, wt_j, wt_k\}$.
2) *Transaction Process:*
   1) `transfer()` triggers transaction:
   $$t = (A_s, A_r, t_v, t_{id}, L, Nonce),$$
   and generates event $M1 = (t, \mathcal{H}(t))$.
   2) The validator v receives $M1$ and queries the source blockchain BC_s using the transaction identifier t_{id} to retrieve the corresponding on-chain transaction record t'.
   3) After verifying that $\mathcal{H}(t) = \mathcal{H}(t')$, the validator forwards $M1$ to Blockchain BC_r.
   4) The contract C_{tx} on BC_r verifies whether $\mathcal{H}(t) \stackrel{?}{=} \mathcal{H}(t')$. If so, the receiver generates a secret S, computes $h_s = \mathcal{H}(S)$, and submits h_s to BC_s through the validator.
   5) The validator builds $\texttt{createHTLC}(t_v, h_s)$ in C_{tx}. This function $\texttt{createHTLC}(t_v, h_s)$ emits an event: $M2 = \mathcal{H}(h_s, A_r, t_v, BC_r)$.
   6) Upon receiving the `createHTLC` event $M2$, the validator verifies it by querying $M2$ on BC_s. If the verification fails, C_{tx} triggers a refund to the sender.
   7) If verification succeeds, the validator sends M_2 to C_{tx} on BC_r as a request for token minting.
   8) On BC_r, C_{tx} verifies:
   $$M2 \stackrel{?}{=} \mathcal{H}(h_s, A_r, t_v, BC_r).$$
   If verification is successful, the token is minted $\texttt{mint}(t'_v)$ and made available to receiver A_r upon providing the secret S'; otherwise, C_{tx} triggers a failure event, and the sender can execute $\texttt{withdraw}(t_v)$ to refund tokens t_v.
   9) After the mint event, the validator forwards S' to BC_s, where C_{tx} verifies whether $h_s \stackrel{?}{=} \mathcal{H}(S')$. Upon successful verification, the token $\texttt{burn}(t_v)$ is executed on BC_s, and the validator is rewarded with t_{gas}.
3) *Settlement Process:*
The settlement process determines the distribution of transaction fees t_{fee} and penalties t_{punish} to corresponding roles, based on the rules defined by the economic model and the confirmation deadline ΔL. Each watchtower $wt \in \{wt_i, wt_j, wt_k\}$ starts a timer. If no transaction event is confirmed within ΔL, the watchtowers query BC_r for the secret S and invoke `checkTimeout()` to determine whether transaction t has timed out, thereby triggering a rollback if necessary.
   4) If the validator v submits the transaction result within $\Delta L/2$, it receives t_{fee}.
   5) Watchtowers (wt) are allowed to submit results between $\Delta L/2$ and ΔL. The first wt to submit S successfully before ΔL receives t_{fee}.
   6) If no valid result is submitted by ΔL, two cases are considered:
      7) *Case 1: No secret S exists on BC_r:* The transaction is deemed unsuccessful due to external factors. The sender is refunded $t_v - t_{\text{gas}}$.
      8) *Case 2: Secret S exists on BC_r:* The failure is attributed to the validator v. Its staking balance is reduced to $p'_{\text{staking}} = p_{\text{staking}} - t_{\text{punish}}$, and the sender is compensated with $t'_v = t_v - t_{\text{gas}} + t_{\text{punish}}$.

This reflects a fail-safe design: when S has appeared on BC_r but the validator does not respond within ΔL, the protocol conservatively attributes the failure to the validator in order to protect asset safety.

C. Exit Phase

When a participant $p \in EN$ decides to settle rewards and exit the system, the following steps, illustrated in Fig. 6, are executed:

1) The participant updates their status to $p_{\text{status}} = 1$ on the routing contract C_{router}, indicating the intention to exit.
2) After confirming that no transactions are in progress, the participant invokes the `withdraw()` function on the staking contract C_{staking} to cancel their stake and withdraw the entire staking balance p_{staking}.

EquiLink builds upon Hashed Time-Locked Contracts (HTLCs) but addresses their inherent limitations. In traditional HTLCs, timeouts often trigger rigid transaction failures and prolonged notification delays, especially if a node becomes unresponsive after receiving secret S. To mitigate these issues, EquiLink's dual-hash locking mechanism introduces the following advancements:

1) *Automated timeout arbitration:* In HTLC, the timeout arbitration step is initiated by the user. In EquiLink, this step is automated and initiated by the watchtower.

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

2) *Secondary Confirmation:* While HTLCs irreversibly abort upon signature failure, EquiLink allows a transaction to proceed if the minting event on BC_r is finalized and S is uploaded; otherwise, it executes a secure rollback.
3) *Simplified Routing Path:* EquiLink eliminates the multi-hop routing and cumulative time-lock redundancies of standard HTLCs. By deploying smart contracts directly on BC_s and BC_r, the protocol streamlines interactions between the sender, validator, and receiver.

D. Economic Model

To ensure the sustainability and fairness of the EquiLink bridge, we design an economic model that incentivizes honest participation and penalizes malicious behavior. Specifically, the model consists of a reward mechanism, which distributes the bridge fee t_{fee} among honest participants p_{honest}, and a penalty mechanism, which deducts the staking amount p_{staking} from participants responsible for transaction failures in order to compensate affected users.

The execution of this model is governed by the smart contracts within the EquiLink Service. The formal representation of the economic model is as follows:

- *Fee Allocation:*

$$\text{Fee}(t) = \begin{cases} v \mapsto t_{\text{fee}}, & \text{if } S \text{ is retrieved by } v, \\ wt \mapsto t_{\text{fee}}, & \text{if } S \text{ is retrieved by } wt, \\ A_s \mapsto t_v - t_{\text{gas}}, & \text{otherwise (refund).} \end{cases}$$

- *Penalty:*

$$\text{If } p \text{ is malicious}, \quad p'_{\text{staking}} = p_{\text{staking}} - t_{\text{punish}}, \quad (t_{\text{punish}} \geq t_{\text{fee}}),$$

where any unilateral deviation strictly reduces the deviator's expected payoff by at least a fixed positive margin t_{punish}. "deviator" denotes the party that chooses defection and thereby incurs the penalty. The execution of the economic model occurs at the final stage of the transaction phase, as detailed in Section 3, where we discuss the specific implementation methods.

VI. Security Analysis

In this section, we first define the security properties that the system achieved. We then provide a formal analysis demonstrating that the proposed design satisfies these properties, followed by theoretical arguments showing that the system can withstand the representative attacks outlined earlier.

A. Security Properties

Based on the proposed threat model in Section III-B, we now define the fundamental security properties that the system achieved. These properties serve as the foundation for the subsequent system security analysis.

Definition 1. (Time-bounded Atomic Settlement): A cross-chain transaction t satisfies *time-bounded settlement* if, once initiated on the source chain BC_s at block height L_{start}, it must be completed within a bounded block interval ΔL. Specifically, by block height $L_{\text{end}} \leq L_{\text{start}} + \Delta L$, either the intended token value t_v is successfully transferred to the receiver A_r on the destination chain BC_r, or refunded to the sender A_s on the source chain BC_s. Formally,

$$\exists\, L_{\text{end}} \leq L_{\text{start}} + \Delta L : [\text{Deliver}(t, A_r) \ \vee \ \text{Refund}(t, A_s)] .$$

Intuitively, this property ensures that every cross-chain transaction terminates within ΔL blocks, achieving atomic completion (either delivery or refund).

Definition 2. (Secret-preserving Security): A cross-chain transaction t satisfies *secret-preserving security* if the unlocking secret S remains computationally hidden from any adversary $\mathcal{A}$ that does not complete the transaction successfully. Formally,

$$\Pr\left[\mathcal{A}(t, h_s) \to S \,\middle|\, t_{\text{status}} = 1\right] \leq \text{negl}(\lambda),$$

where h_s is the hash commitment of S, and negl(λ) denotes a negligible function in λ.

Intuitively, this property ensures that the secret S cannot be derived from on-chain information or any protocol transcript unless the transaction is successfully settled.

Definition 3. (Exclusive Participants Authority): A cross-chain transaction t satisfies *exclusive participants authority* if only the designated participants assigned to t are authorized to access or operate on the locked asset. Let $DVG(t)$ denote the set of participants assigned to t. Formally, for any requesting participant p, access is granted if and only if

$$p \in DVG(t) \quad \Longleftrightarrow \quad \text{Authorized}(p, t) = \text{true}.$$

Intuitively, this property ensures that no unauthorized participant can interfere with or claim the asset locked in the cross-chain transaction, providing strict access control for transaction execution.

Definition 4. (Incentive Effectiveness): Let $DVG(t)$ denote the designated participants for transaction t. Each $p \in DVG(t)$ chooses a strategy from $\{\textsf{Cooperation}, \textsf{Defection}\}$. Let $t_{\text{fee}} > 0$ be the per-transaction reward, $t_{\text{gas}} \geq 0$ denotes the operating cost (e.g., gas, computation), $t_{\text{punish}} \geq 0$ denotes any potential illicit gain from a successful deviation.

The per-transaction payoff of p is

$$R_p = \begin{cases} t_{\text{fee}} - t_{\text{gas}}, & \text{if Cooperation}, \\ t_{\text{fee}} - t_{\text{gas}} - t_{\text{punish}}, & \text{if Defection}, \end{cases}$$

where R_p represents the transaction fee-related gain (or loss) of participant p, $t_{\text{punish}} > 0$ is the on-chain penalty (deducted from the staked deposit p_{staking}).

B. Security Proof

We formally analyze whether the proposed system satisfies the security properties defined in Section VI-A. To establish the overall security of the protocol, it is sufficient to prove Theorems 1–4.

Theorem 1. (Time-bounded Atomic Settlement): If at least one honest participant p in DVG, each cross-chain transaction t achieves atomic completion within ΔL blocks, ensuring either successful delivery or full refund.

Proof: See Appendix B, (available online) for the proof. □

Theorem 2. (Secret-preserving Guarantee):If at least one honest participant ($p_{honest} > 0$) exists in DVG, malicious actors cannot forge the secret S, thereby preventing asset theft.

Proof: See Appendix C, (available online) for the proof. □

Theorem 3. (Exclusive Participants Authority Guarantee): For any cross-chain transaction t, only participants included in $DVG(t)$ are able to initiate, approve, or execute state transitions over the assets locked by t. Any participant $p \notin DVG(t)$ cannot influence the token flow of t with non-negligible probability.

Proof: See Appendix D, (available online) for the proof. □

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

Theorem 4. (Incentive Effectiveness Guarantee): For any designated participant and under other participants' strategies, choosing Honest yields a strictly higher expected payoff than Defection by a positive margin; consequently, the all-honest strategies forms a strict Nash equilibrium.

Proof: See Appendix E, (available online) for the proof. □

C. Attack Resistance Analysis

Consistent with Section III-B, we assume an adversary that can corrupt a subset of participants (validators, watchtowers, and users), arbitrarily control the network, and compromise a subset of TEE-enabled participants (including potential side-channel leakage). However, the adversary cannot reverse finalized blockchain states, and the protocol remains secure as long as at least one honest participant exists in the selected DVG for each transaction. We analyze the following attack types, leveraging **Theorems 1–4** and **Definitions 1–4**.

- *Double-Spending Attacks:*
  Analysis: By **Definition 1** (and **Theorem 1**), every cross-chain transaction t must either deliver to A_r on BC_r or refund to A_s on BC_s within ΔL, with no admissible intermediate state that both mints on BC_r and retains locked value on BC_s. Hence conflicting spends cannot simultaneously succeed without violating atomic settlement.
- *Replay Attacks:*
  Analysis: Redeeming on BC_r requires the preimage S of h_s; by **Definition 2**, S is computationally hidden unless the transaction legitimately completes. By **Definition 1**, expired HTLCs refund on BC_s after ΔL, so re-sending old messages cannot trigger a second valid redemption. Thus replay attempts fail except with negligible probability.
- *Compromised Participants:*
  Analysis: By **Theorem 3**, only designated participants $DVG(t)$ can authorize state transitions for t, thereby preventing unauthorized forgery or equivocation. Moreover, by **Theorem 4**, any deviation from the protocol results in strictly lower expected payoff due to on-chain slashing, making honest behavior incentive-compatible. Therefore, the randomized DVG selection mechanism provides strong robustness against partial participant compromise, ensuring correct progress as long as at least one honest participant is included in the selected DVG. The corresponding probability analysis is given in Appendix F, available online.
- *Side-Channel Attacks:*
  Analysis: Suppose an adversary extracts information from a subset of TEEs through side-channel attacks. The leaked information may include the transaction identifier t, the verification messages M_1, M_2, and the secret value S. However, according to **Theorem 3**, each TEE only verifies its assigned transactions and cannot influence unrelated transactions across the network. Moreover, t and M_1, M_2 are already publicly available on the blockchain for transaction validation, so their disclosure does not introduce additional security risks. Even if S is exposed, the smart contract's access-control logic only allows it to determine whether the transaction is redeemed or refunded, and it cannot be used to steal assets arbitrarily. Any attempt to bypass the contract logic would violate the blockchain's consensus and finality guarantees. Therefore, side-channel leakage does not enable unauthorized asset creation or theft.
- *Physical Faults:*
  Analysis: Upon physical failure $t_{\text{status}} = 1$, watchtowers are part of $DVG(t)$ and, by **Definition 3**, are authorized to take over upon validator crashes/disconnections. Combined with **Definition 1** (and **Theorem 1**), they can either submit S for delivery or trigger refund at timeout, ensuring that t terminates within ΔL despite node failures.

VII. Experimental Evaluation

This section presents the experimental setting and outlines the key metrics used to evaluate the performance of cross-chain bridges, including transaction cost, latency, economic model analysis, overall consumption, and security assessment. These metrics are critical for evaluating the effectiveness of cross-chain transactions and identifying key design challenges. The subsequent subsections provide a detailed elaboration of each metric.

A. Experimental Setting

Six representative bridges were selected for evaluation based on their versatility and degree of decentralization, including two centralized bridges (Across Protocol, cBridge [14]) and four decentralized bridges (zkCross, ALBA, zkBridge, and XClaim). To improve the representativeness of the evaluation, we further included the cross-chain bridge aggregator LI.FI [35], which provides the best available routing option among multiple cross-chain bridges at the time of execution.

Experiments were conducted in two Ethereum-based environments. Across Protocol, cBridge, and LI.FI were evaluated in their real online environments using WETH-to-ETH transfers between Ethereum Mainnet and Arbitrum One. To ensure fair comparison, transactions on these platforms were initiated at the same time under the same network conditions, so that the quoted prices and execution environments were as consistent as possible. In contrast, zkCross, ALBA, zkBridge,and XClaim do not provide publicly accessible official platforms or complete deployable code for direct online testing. Therefore, after re-implementing these systems, we evaluated them on a local Ganache blockchain forked from Ethereum Mainnet under a unified experimental setup. The block generation time was fixed at 12 seconds.

B. Transaction Cost

Transaction cost refers to the total cost of a cross-chain transaction, defined as $C = t_{\text{gas}} + t_{\text{fee}}$. It is a key factor that determines the potential for widespread user adoption of a cross-chain bridge. In this subsection, we take the cost of a single transaction as the baseline. Based on the experimental settings in Section VII-A, gas metrics are retrieved from live mainnet logs for online bridges and from execution traces for our re-implemented local baselines.

Fig. 7 shows that EquiLink achieves lower gas costs than existing solutions, even under pessimistic execution conditions. Specifically, its gas consumption under pessimistic scenario is 14.1% lower than the optimistic scenario of the zkCross, demonstrating robustness across dispute scenarios. Figures 8 and 9 illustrate that EquiLink maintains stable gas consumption through a mint-and-burn design, with over 98% of transactions

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

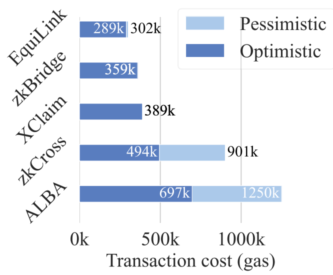


Fig. 7. Comparison of gas consumption between non-custodial bridges and EquiLink.

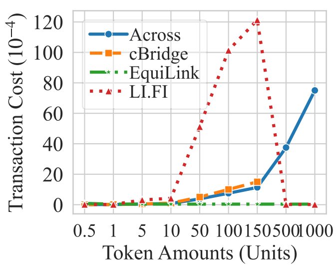


Fig. 8. Comparison of gas consumption between custodial bridges and EquiLink.

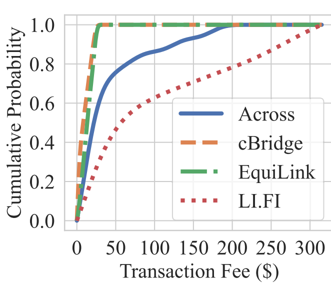


Fig. 9. Comparison of transaction fee CDF between custodial bridges and EquiLink.

costing less than $25, demonstrating cost-effectiveness, particularly for high-value transfers.

C. Transaction Latency

Transaction latency refers to the time elapsed from when the sender A_s invokes the smart contract on chain BC_s to when the receiver A_r on chain BC_r successfully receives the token v. It is a critical metric for evaluating cross-chain bridges, as it has a direct impact on user experience and system security during transaction settlement periods. To ensure the accuracy of our latency evaluation, we conducted 10 independent experiments for each token amount ranging from 0.1 ETH to 500 ETH, and averaged the results to minimize environmental noise.

Fig. 10 illustrates the cumulative distribution function (CDF) of transaction confirmation times. As shown, EquiLink significantly outperforms the other three, with most transactions confirmed within 36 seconds, while the others all exceed 240 seconds. Fig. 12 compares transaction latencies across different token amounts, where EquiLink consistently achieves the lowest latency and maintains stable performance between 28 and 37 seconds regardless of token amounts. Across and LI.FI exhibit increasing latency with token amounts, reaching up to 900 and 540 seconds, respectively, for large transfers. Fig. 11

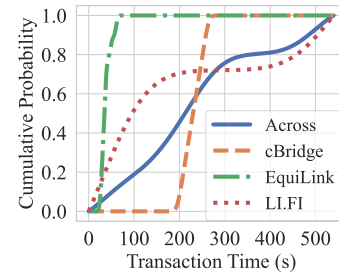


Fig. 10. Comparison of transaction time CDF between custodial bridges and EquiLink.

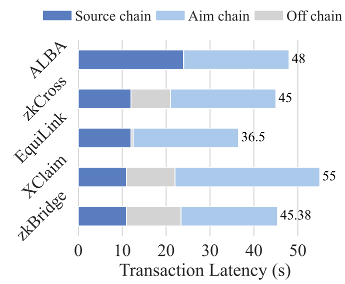


Fig. 11. Comparison of transaction latency between non-custodial bridges and EquiLink.

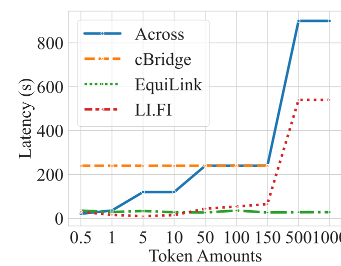


Fig. 12. Comparison of transaction latency between custodial bridges and EquiLink.

further breaks down the end-to-end latency of EquiLink into approximately 12 seconds on the source chain, 0.5 seconds for off-chain processing, and 24 seconds on the aim chain, reduces the average cross-chain transaction latency by 18.9%. These findings demonstrate that EquiLink provides more predictable and scalable latency performance, making it particularly suitable for real-time or high-frequency cross-chain interactions.

D. Economic Model Analysis

To analyze the economic model of the cross-chain bridge, we employ a game-theoretic approach to understand the strategic interactions among various participants.

1) Model Setup: We define the gain model in the cross-chain bridge as a constant-sum game, because the participants, namely the validator and watchtowers, compete over the allocation of the fixed total fee t_{fee}. Each participant aims to maximize its own gain while minimizing the gains of others. The game is expressed as $\sum_{p=1}^{n} R_p = t_{\text{fee}}$, where R_p represents the transaction fee-related gain (or loss) of participant p, and n is the number of players included in the game model.

The key components and players in the game are defined as follows: the players consist of a validator v and a representative

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

TABLE I
PAYOFF MATRIX BETWEEN VALIDATOR (V) AND WATCHTOWER (WT)

v \ wt	C	D
C	$(t_{\text{fee}}, 0)$	$(t_{\text{fee}}, 0)$
D	$(-t_{\text{punish}}, t_{\text{fee}})$	$(0, 0)$

watchtower wt, and each player can choose one of two strategies, namely Cooperation (C) or Defection (D).

We list the assumptions made in the model as follows: (i) the validator and watchtower have two possible behaviors, namely cooperation and defection; (ii) if the validator cooperates, the watchtower determines whether it needs to intervene in the event of validator failure or misbehavior; and (iii) in the event of a failure, the watchtower takes over to determine the cause and distribute penalties or rewards accordingly.

2) Game Representation: Let x denote the strategy profile vector of the players. The validator's and watchtower's payoffs are derived from the transaction fee t_{fee}, which is provided by the sender. The corresponding payoff matrix is shown in Table I.

3) Equilibrium Analysis: For the validator, cooperating yields a higher payoff than defecting, as $t_{\text{fee}} > t_{\text{punish}}$. For the Watchtower, if the validator defects, cooperating yields a payoff of t_{fee}, which is strictly greater than the zero payoff from defection, i.e., $t_{\text{fee}} > 0$. Therefore, the strategy profile (C, C) forms a Nash equilibrium because neither the validator nor the watchtower can improve their payoff by unilaterally changing their strategy. Any unilateral deviation strictly reduces the deviator's expected payoff by at least a fixed positive margin t_{punish}. Cooperation ensures the most efficient and secure operation of the cross-chain bridge, aligning with the incentives for all participants and achieving system sustainable economic model.

4) Economic Attack Analysis: EquiLink provides both protocol-level and incentive-level protection against economic attacks such as stake griefing, bribery, and fee manipulation. First, as discussed in Section V-D, fee allocation is separated from punishment, so transaction failure alone does not imply validator loss, while slashing applies only to malicious behavior. Second, the game model in Table I shows that honest cooperation yields higher payoff than malicious deviation when the punishment is sufficiently high. Third, the randomized DVG selection mechanism increases the difficulty of bribery and collusion, while the fixed smart-contract fee rule limits the room for fee manipulation. A more detailed discussion is provided in Appendix G, available online.

E. Overall Consumption

Overall consumption measures the total resource cost and performance of participants in the EquiLink Network. A lower entry barrier, reflected in reduced consumption and performance requirements, promotes broader participation and enhances the system's scalability and decentralization. To comprehensively evaluate the system's overall consumption, we evaluate the resource cost associated with *System Initialization*, *Node Joining*, and *Node Exiting*, as well as the performance impact during *Transaction Validation*, including latency and CPU utilization.

1) System Initialization: The total cost for initializing the system includes deploying all the decentralized components, with gas costs detailed in Table II. While the initialization cost is approximately 3.6 million gas, it is a one-time expense that provides decentralized services for all users, making it a justified and cost-efficient investment.

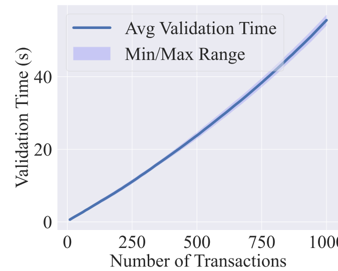


Fig. 13. Validation time for varying numbers of transactions.

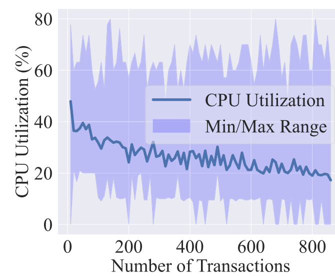


Fig. 14. CPU Utilization for varying numbers of transactions.

2) Node Joining Costs: When a new node joins the system, it incurs gas costs associated with staking and adding itself to the router list, as outlined in Table II. The total gas cost of 158,746 for node entry is equivalent to around 7-8 typical Ethereum transactions, which is quickly recouped through rewards earned from validating cross-chain transactions.

3) Node Exiting Costs: Exiting the system involves updating the routing contract to mark the node as inactive and withdrawing the staked amount. The gas costs for these actions are summarized in Table II. The total exit cost is comparable to approximately 3 Ethereum transactions, which can be typically offset by the node's earnings from participating in the network.

4) Performance of Transaction Validation: Validation time and CPU utilization reflect device performance requirements and thus influence the entry barriers to participation. We evaluate both metrics for off-chain transaction validation as the number of transactions increases from 1 to 1000. Experiments were conducted on an Alibaba Cloud server with 4 CPU cores and 16 GB of RAM, with the TEE environment executed on Intel SGX.

Fig. 13 shows that the average validation time increases with the number of transactions, while remaining smooth and stable without abrupt jumps or performance collapse. This suggests that the TEE-based validation process maintains steady processing behavior in the evaluated range, without approaching the EPC-limit-induced bottlenecks in our experiments. Fig. 14 shows that CPU utilization remains at a moderate level during continuous execution. Overall, the average CPU load stays relatively low and does not exhibit sustained growth with the number of transactions, suggesting only limited continuous CPU overhead during long-running operation.

F. Security Assessment

Given that cross-chain bridges involve asset transfers, it is essential to assess the resilience of EquiLink. We conducted experiments that replicated several critical attack vectors: smart

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

TABLE II
GAS CONSUMPTION FOR SYSTEM INITIALIZATION, NODE JOINING, AND NODE EXITING

System Initialization		Node Joining		Node Exiting	
Component/Action	**Gas Consumption**	**Action**	**Gas Consumption**	**Action**	**Gas Consumption**
Routing Contract	956,682	Node Staking ETH	65,967	Updating Routing Contract	33,561
Staking Contract	783,535	Adding to Router List	92,779	Withdrawing Staked Token	31,179
HTLC Contract for Sender	471,830				
HTLC Contract for Receiver	1,390,830				
Total: 3,602,877		**Total: 158,746**		**Total: 64,740**	

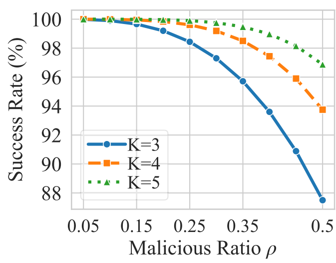


Fig. 15. Transaction success rate under key compromise for different DVG sizes K and malicious ratios ρ.

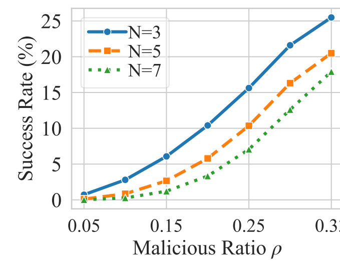


Fig. 16. Attestation compromise success rate under different attester numbers N and malicious ratios ρ.

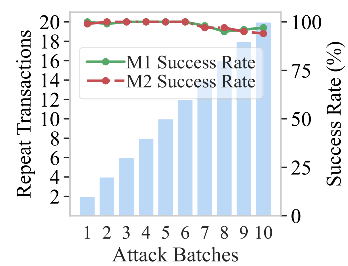


Fig. 17. Comparison of replay attack counts and success rates for M1 and M2 across different attack batches.

contract vulnerabilities, key/attestation compromise, replay attacks and physical attacks. Our security assessment framework integrates simulated adversarial environments and formal verification results.

1) Resistance to Smart Contract Vulnerabilities: We analyzed the implemented HTLC contracts using established auditing tools, including Slither and related security testing frameworks. The findings were limited to implementation-level best-practice issues, and we revised the contract code accordingly following secure coding guidelines. This suggests that, although protocol design can reduce the attack surface at the system level, smart contract vulnerabilities are not automatically eliminated by the protocol itself and still require careful attention from developers.

2) Resistance to Key/Attestation Compromise: For key compromise, we focus on the transaction phase and examine the success probability under randomized DVG selection, as shown in Fig. 15. The results show that, for a fixed malicious ratio ρ, a larger DVG size K leads to a higher transaction success rate, because the probability that the selected DVG contains at least one honest participant increases with K. However, as analyzed in Section VI-C and Appendix F, (available online), even if the keys of all selected DVG members are compromised, the impact is still bounded to degraded liveness, such as transaction failure or timeout, rather than unauthorized asset transfer. Therefore, key compromise may degrade liveness, but it does not violate asset safety.

For attestation compromise, we focus on the initialization phase and evaluate the compromise success rate under different attester numbers N, as shown in Fig. 16. The results show that increasing the number of attesters reduces the probability that a forged attestation can pass the majority-based admission rule, because an adversary must compromise more attesters to exceed the threshold. This confirms that the replicated-attester design improves robustness against attestation compromise. Moreover, even when attestation compromise occurs and a malicious participant is admitted, the impact is still mainly limited to reduced admission reliability rather than direct system failure or cross-chain transaction failure.

3) Resistance to Replay Attacks: Replay attacks occur when a malicious actor intercepts and retransmits valid cross-chain transaction data, potentially causing duplicate asset transfers. In this experiment, each transaction is repeated twice with identical events: M1 and M1' (from BC_s to BC_r) and M2 and M2' (from BC_r to BC_s). The system's response is tested by having the C_{tx} contract at BC_r store and validate the transaction ID (t_{id}), ensure only one instance (either M1 or M1') is accepted while duplicates are rejected.

Data were collected from 10 attack batches, with the number of repeat transactions per batch increasing from 2 to 20. The success rates in handling repeated transactions for M1 and M2 are plotted in Fig. 17. Results show that both M1 and M2 effectively prevent replay attacks, with M1 generally performing better. The system maintains high resilience, though success rates decrease as the transaction volume rises. This decline is not due to replay attack failures but rather contract invocation failures under high concurrency due to Ganache.

4) Resistance to Physical Faults: In addition to traditional software threats, participants $p \in EN$ are also vulnerable to physical faults like network disconnections or power outages, which can lead to participant failures during transaction execution.We validated this approach through an experiment simulating a physical attack scenario, where the validator v and Watchtowers wt_1, wt_2, wt_3 were abruptly disconnected during transaction execution. Specifically, we considered five representative cases, **Case 1:** the validator v disconnects after the

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

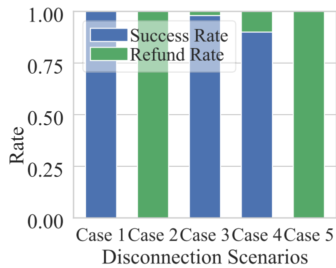


Fig. 18. Comparison of success and revoke rates across different physical faults cases.

transaction secret s has been submitted to BC_r; **Case 2:** the validator v disconnects before the secret s is submitted; **Case 3:** the validator v and one watchtower disconnect simultaneously; **Case 4:** a majority of participants disconnect, for example due to a large-scale outage; and **Case 5:** a complete network outage in the extreme case.

As shown in Fig. 18, in cases 1-4, as long as at least one participant is functional, the Watchtower can still complete the transaction. A small number of revocations occurred due to timeout $L_{\text{end}} - L_{\text{start}} > \Delta L$, demonstrating resilience to physical threats. In the extreme Case 5, where all participants disconnect, the sender can still withdraw assets, ensure no asset loss.

VIII. Discussion

A. Limitations

The EquiLink bridge addresses several challenges by mitigating asset custody risks and single points of failure in centralized bridges through a decentralized network. In addition, by performing off-chain computation with on-chain verification, it significantly reduces transaction costs and latency compared to fully decentralized bridges. However, the current design still has limitations that warrant further investigation and refinement.

The existing scheme does not provide privacy protection for transactions, as all transaction details remain fully transparent on the blockchain. Although this transparency is beneficial for auditability and verifiability, it may be undesirable in privacy-sensitive application scenarios, such as confidential business transactions, privacy-preserving financial activities, or cases where users do not wish their transaction relationships and asset flows to be publicly exposed. Second, although the design of EquiLink is in principle compatible with multiple blockchain systems, the current implementation and evaluation still focus mainly on EVM-compatible environments, and support for heterogeneous chains remains limited. Extending it to non-EVM chains and Layer-2 rollup systems still requires further protocol adaptation.

B. Future Work

One promising direction for future work is to enhance the privacy protection of EquiLink transactions. A potential solution is to integrate ZKPs to conceal transaction details while still ensuring correctness and verifiability. By using ZKPs, the bridge could enable private cross-chain transactions where neither the transaction amount nor the sender/receiver identities are publicly disclosed.

Another important direction for future work is to extend EquiLink to broader heterogeneous blockchain systems, including non-EVM chains and Layer-2 rollups. Although its architecture is in principle compatible with multiple blockchains, further protocol adaptation is still needed to support diverse execution and verification models in practice, so that EquiLink can serve as a more general underlying layer for cross-chain interoperability.

IX. Conclusion

In this paper, we presented EquiLink Bridge, a novel semi-custodial cross-chain architecture that systematically balances efficiency, decentralization, and security. By leveraging TEE-based trusted off-chain computation, EquiLink effectively reduces transaction costs by 14.1% and latency by 18.9% compared to existing solutions, achieving cost- and latency-efficient cross-chain performance. To further promote decentralization, EquiLink introduces a lightweight decentralized service and verification network that enables rapid node joining and exiting, while maintaining a minimal validation workload, thereby significantly lowering the entry barriers for participants and preserving security guarantees. Through the integration of atomic cross-chain transaction protocols and economic incentive models, EquiLink enhances resistance against replay attacks, physical attacks, and other potential threats, achieving a 100% success/revocation rate for cross-chain asset transfers. Overall, EquiLink offers a promising pathway toward achieving efficient, decentralized, and secure cross-chain interoperability.

References

[1] CoinGecko, "Top blockchains by total value locked (TVL)," 2025. Accessed: Feb. 19, 2025. [Online]. Available: https://www.coingecko.com/en/chains

[2] W. Liu, B. Cao, M. Peng, and B. Li, "Distributed and parallel blockchain: Towards a multi-chain system with enhanced security," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 1, pp. 723–739, Jan./Feb. 2025.

[3] H. Mao, T. Nie, H. Sun, D. Shen, and G. Yu, "A survey on cross-chain technology: Challenges, development, and prospect," *IEEE Access*, vol. 11, pp. 45527–45546, 2022.

[4] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *ACM Comput. Surv.*, vol. 54, no. 8, Oct. 2021, doi: 10.1145/3471140.

[5] A. Augusto, R. Belchior, J. Pfannschmidt, A. Vasconcelos, and M. Correia, "XChainWatcher: Monitoring and identifying attacks in cross-chain bridges," 2024, *arXiv:2410.02029*.

[6] S.-S. Lee, A. Murashkin, M. Derka, and J. Gorzny, "SoK: Not quite water under the bridge: Review of cross-chain bridge hacks," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency*, 2023, pp. 1–14.

[7] N. Li et al., "Blockchain cross-chain bridge security: Challenges, solutions, and future outlook," *Distrib. Ledger Technol.: Res. Pract.*, vol. 4, no. 1, pp. 1–34, 2025.

[8] C. Liu et al., "Extending on-chain trust to off-chain–trustworthy blockchain data collection using trusted execution environment (TEE)," *IEEE Trans. Comput.*, vol. 71, no. 12, pp. 3268–3280, Dec. 2022.

[9] R. Tsepeleva and V. Korkhov, "Implementation of the cross-blockchain interacting protocol," in *Proc. 21st Int. Conf. Comput. Sci. Appl.*, Cagliari, Italy, Springer, 2021, pp. 42–55.

[10] M. Li, Y. Yang, G. Chen, M. Yan, and Y. Zhang, "SoK: Understanding design choices and pitfalls of trusted execution environments," in *Proc. 19th ACM Asia Conf. Comput. Commun. Secur.*, 2024, pp. 1600–1616.

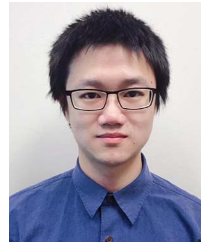
[11] R. Cheng et al., "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts," in *Proc. Eur. Symp. Secur. Privacy*, 2019, pp. 185–200.

[12] H. Yin, S. Zhou, and J. Jiang, "Phala network: A confidential smart contract network based on polkadot," *Phala Netw.*, 2019. [Online]. Available: https://crebaco.com/planner/admin/uploads/whitepapers/3580918phala-paper.pdf

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.

[13] S. P. Krishna and P. Singh, “Security challenges in building blockchains bridges and countermeasures,” *Evergreen*, vol. 10, no. 3, pp. 1558–1569, 2023.
[14] Celer Network, “The best crypto & binance bridge| cbridge,” 2025. Accessed: Mar. 23, 2026. [Online]. Available: https://cbridge.celer.network
[15] Across Protocol, “Across protocol,” 2025. Accessed: Feb. 25, 2025. [Online]. Available: https://across.to/
[16] rhino.fi, “rhino.fi,” 2025. Accessed: Feb. 25, 2025. [Online]. Available: https://app.rhino.fi/
[17] Chainalysis Team, “Multichain exploit: Possible hack or rug pull,” Jul. 2023. Accessed: Oct. 7, 2024. [Online]. Available: https://www.chainalysis.com/blog/multichain-exploit-july-2023/
[18] BNB Chain, “BNB chain bridge,” 2025. Accessed: Feb. 25, 2025. [Online]. Available: https://www.bnbchain.org/en/bnb-chain-bridge
[19] Y. Li, H. Liu, and Y. Tan, “Polybridge: A crosschain bridge for heterogeneous blockchains,” in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency*, 2022, pp. 1–2.
[20] “Rhino.fi bridge: Bridge your crypto to and from multiple chains in seconds,” 2026. Accessed: Mar. 23, 2026. [Online]. Available: https://app.rhino.fi/bridge
[21] “xDAI bridge - gnosis chain,” 2023. Accessed: Feb. 25, 2025. [Online]. Available: https://bridge.gnosischain.com
[22] P. Han, Z. Yan, W. Ding, S. Fei, and Z. Wan, “A survey on cross-chain technologies,” *Distrib. Ledger Technol.: Res. Pract.*, vol. 2, no. 2, pp. 1–30, 2023.
[23] M. Staples et al., *Risks and Opportunities for Systems Using Blockchain and Smart Contracts. Data61*. Sydney, NSW, Australia:CSIRO, 2017.
[24] A. Labs, “Rainbow bridge,” 2025. Accessed: Mar. 18, 2025. [Online]. Available: https://rainbowbridge.app/
[25] Y. Guo et al., “zkCross: A novel architecture for cross-chain privacy-preserving auditing,” in *Proc. 33rd USENIX Secur. Symp. (USENIX Secur. 24)*, Aug. 2024, pp. 6219–6235.
[26] T. Xie et al., “zkbridge: Trustless cross-chain bridges made practical,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2022, pp. 3003–3017.
[27] A. Zamyatin, D. Harz, J. Lind, P. Panayiotou, A. Gervais, and W. Knottenbelt, “XClaim: Trustless, interoperable, cryptocurrency-backed assets,” in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 193–210.
[28] C. Huang, T. Yan, and C. J. Tessone, “Seamlessly transferring assets through layer-0 bridges: An empirical analysis of stargate bridge’s architecture and dynamics,” in *Proc. Companion Proc. ACM Web Conf.*, 2024, pp. 1776–1784.
[29] C. Labs, “Cross-chain interoperability protocol (CCIP),” 2025. Accessed: Mar. 18, 2025. [Online]. Available: https://chain.link/cross-chain/
[30] G. Scaffino, L. Aumayr, M. Bastankhah, Z. Avarikioti, and M. Maffei, “Alba: The dawn of scalable bridges for blockchains,” in *Proc. Network Distrib. Syst. Secur. Symp.*, 2025, pp. 1–20.
[31] S. Ismail, H. Reza, H. K. Zadeh, and F. Vasefi, “A blockchain-based IoT security solution using multichain,” in *Proc. IEEE 13th Annu. Comput. Commun. Workshop Conf.*, 2023, pp. 1105–1111.
[32] R. Lan, G. Upadhyaya, S. Tse, and M. Zamani, “Horizon: A gas-efficient, trustless bridge for cross-chain transactions,” 2021, *arXiv:2101.06000*.
[33] B. Pillai, K. Biswas, Z. Hóu, and V. Muthukkumarasamy, “Cross-blockchain technology: Integration framework and security assumptions,” *IEEE access*, vol. 10, pp. 41239–41259, 2022.
[34] O. Wu, S. Li, Y. Wang, H. Li, and H. Zhang, “Modeling cross-blockchain process using queueing theory: The case of cosmos,” in *Proc. IEEE 28th Int. Conf. Parallel Distrib. Syst.*, 2023, pp. 274–281.
[35] LI FI, “LI.FI’s intent/solver marketplace,” 2026. Accessed: Mar. 23, 2026. [Online]. Available: https://docs.li.fi/lifi-intents/introduction

Tingda Shen is working toward the PhD degree with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications. His research interests include blockchain applications, sharding, and data security.



Yebo Feng received the PhD degree in computer science from the University of Oregon. He is currently a research fellow with Nanyang Technological University. His research interests include network security, blockchain security, AI security, and anomaly detection.

Jin Dong is currently a PhD Professor with the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing. His research interests include integrated circuit design, blockchain, privacy computing, and AI.

Konglin Zhu received the master’s degree in computer science from the University of California, Los Angeles, CA, USA, in 2009 and the PhD degree from the University of Göttingen, Germany, in 2014. He is currently an associate professor with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include Internet of Vehicles, edge computing, and distributed learning.

Lei Jiao received the PhD degree in computer science from the University of Göttingen, Germany. He was a technical staff member with Nokia Bell Labs, Ireland. He is currently a faculty member with the University of Oregon, USA. His research interests include machine learning systems, edge computing, cloud computing, network optimization, and network economics.

Lin Zhang received the PhD degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 2001. He joined Beijing University of Posts and Telecommunications in 2004. Since 2011, he has been a professor with the Beijing University of Posts and Telecommunications. He is currently the director of Beijing Big Data Center. His research focuses on mobile cloud computing.

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on June 22,2026 at 08:40:56 UTC from IEEE Xplore. Restrictions apply.