Double-Spending Attacks in Cross-Blockchain Ecosystems[1]

Aradhita Mukherjee [a], Luca Olivieri [a], Nabendu Chaki [b], Agostino Cortesi [a]

[a] *Department of Environmental Sciences, Computer Science and Statistics, Ca' Foscari University of Venice, Venice, Italy*
[b] *Department of Computer Science and Engineering, University of Calcutta, Kolkata, West Bengal, India*

Abstract

Blockchain technology has continued to evolve rapidly in recent years. However, the lack of efficient methods to transmit data and value between different blockchains has created many disjoint data silos. Cross-chain technology has been introduced and applied across various fields, including distributed transaction platforms, digital finance, and electronic governance, to address this issue. However, this, in turn, reignited concerns regarding possible vulnerabilities in the cross-chain environment.

In this paper, we aim to explore how double-spending vulnerabilities manifest in cross-chain environments. Specifically, we investigate how attack vectors at consensus-level and network-level may be leveraged or amplified in cross-chain transactions. Although, double-spending attacks are well-documented in single-chain systems, there are still issues and challenges related to these vulnerabilities in blockchain interoperability settings.This allows us to pinpoint the potential security threats associated with cross-chain technology and identify future research directions.

Keywords: Blockchain technology, Blockchain networks, Double-spending attack, Smart contract, Cross-chain technology

1. Introduction

Blockchain, a decentralized and distributed ledger technology, has emerged as a critical innovation due to its ability to provide transparent, immutable, and tamper-proof records of transactions [1]. These features led to its widespread adoption, particularly in the economic and financial spheres, where it has been used as the backbone for systems involved in transferring digital assets, goods, and cash [2, 3, 4]. In recent years, blockchain interoperability [5, 6] has become a pressing need for industries and governments as the adoption of blockchain technology expands across diverse sectors. The current landscape, characterized by isolated blockchain networks, limits the seamless exchange of information and assets, creating inefficiencies and silos that hinder innovation and growth. For industries, interoperability is essential to unlocking the full potential of blockchain by enabling supply chains, financial services, and other ecosystems to interact fluidly across different platforms. Similarly, several governments considered blockchain a strategic technology [7, 8, 9, 10]. Then, they require interoperable blockchains to facilitate cross-border trade, enhance transparency in inter-agency operations, and potentially coordinate international regulatory frameworks, contracts, and agreements [11, 12].

[1] This is the author's accepted manuscript of an article published in *Blockchain: Research and Applications*. The final version of record is available at Elsevier via `https://doi.org/10.1016/j.bcra.2025.100378`

Email addresses: `aradhita.mukherjee@unive.it` (Aradhita Mukherjee), `luca.olivieri@unive.it` (Luca Olivieri), `nabendu@ieee.org` (Nabendu Chaki), `cortesi@unive.it` (Agostino Cortesi)

To address these challenges, cross-chain and multi-chain approaches have been developed to achieve scalability and interoperability.

The cross-chain approach [13] has enabled interaction between different blockchain systems, facilitating asset transfers and data synchronization without merging chains. It supports use cases like decentralized finance (DeFi), token transfers, and wrapped tokens, but also faces challenges such as security vulnerabilities, double-spending attacks, and synchronization issues. Several solutions (e.g., oracles, bridges, and relayers) enable cross-chain interoperability between blockchains but come with significant risks. Oracles fetch external or blockchain-specific data, introducing vulnerabilities like centralization, malicious data injection, or spoofing attacks. Bridges, which handle token exchanges between blockchains, face risks such as smart contract bugs, bridge hacks, and reliance on trusted custodians. Relayers, which validate and transmit data across chains, are vulnerable to collusion, censorship, and misreporting. Together, these mechanisms expand functionality but also introduce attack vectors that can compromise security and trust.

In contrast, multi-chain technology involves parallel blockchains within a unified ecosystem, improving scalability. This approach has benefited from a shared governance and consensus model, enhancing performance and security across the ecosystem. Multi-chain setups are well-suited for high-throughput platforms and intensive decentralized applications (dApps). Cross-chain and multi-chain serve distinct purposes. While the former enables interaction between independent chains, the latter enhances scalability within a single ecosystem. Further differences between the two have been highlighted in Table 1. Together, they have advanced the evolution of blockchain networks, supporting the growing demands of decentralized applications and global transactions.

Table 1. Comparison between Cross-chain and Multi-chain

Feature	Cross-chain	Multi-chain
Purpose	Enables interoperability between distinct blockchains.	Scale transaction throughput across multiple chains in one ecosystem.
Interoperability	Highly interoperable that connects independent chains like Ethereum and Bitcoin.	Limited to chains within the same ecosystem.
Governance	Independent governance per chain.	Shared governance across chains.
Security	Depending on trusted bridges or relays.	Pooled/shared security across chains.
Use case	Token transfers, DeFi, and wrapped assets.	High-throughput applications, dApps.
Challenges	Vulnerable to bridge failures and double-spending.	Requires robust consensus to maintain uniformity.

Blockchain technology, not surprisingly, has become disruptive for digital transfer payments due to its ability to address these long-standing inefficiencies of traditional digital systems without third party intermediaries [2]. A notable issue that blockchain attempts to solve is double-spending. It allows a malicious user or group to spend the same digital asset more than once to improperly access a service or get assets by exploiting various attack vectors. However, the blockchain cannot completely avoid it [14] but can strongly discourage it, involving economic-social and political aspects (e.g., cryptocurrencies market collapse, falling prices, loss of reputation and trust in the service).

However, if this is true for individual blockchains, double-spending can have a completely different meaning in cross-chain transfers. Indeed, it might make sense for the actors of a single blockchain to “collapse” it and have huge gains in other more popular or highly rated blockchains.

The research on double-spending attacks in cross-chain environments is limited and still in an early stage [15]. Such attacks may compromise the security of multiple blockchains. Attackers can exploit discrepancies in transaction confirmation time or security gaps in one chain to enable double-spending across other chains, which may lead to serious issues, like chain reorganizations, invalid transfers, or asset duplication. These disruptions may put consensus and trust in the individual blockchains, and the bridges connecting them are vulnerable. Addressing the detection and mitigation of these attacks in a decentralized, cross-chain setting has thus remained a formidable challenge in a blockchain ecosystem.

In this work, we have explored thoroughly whether the various forms of double-spending attacks seen in single-chain networks, are also feasible in cross-chain environments. Wherever this is not true, we have tried to identify the reasons behind it. We have also simulated various attacks feasible in a cross-chain setting and provided both

theoretical and experimental comparisons between them. The remaining portion of this article has been organized as follows: Section 2 provides a background and preliminary notions on the double-spending problem and blockchain technology. Section 3 investigates related work. Section 4 explores different types of double-spending attacks in cross-chain settings. Section 5 provides an experimental evaluation to replicate feasible double-spending attacks. Section 6 discusses the results and findings of the experimental evaluation. Section 7 concludes the paper and discusses future research directions.

2. Background and Preliminaries

In the physical world, spending currency means transferring one or more physical objects, i.e. coins and banknotes, from one owner to another. Even if this action may seem trivial, behind the scenes, it implies that the sender who spent those objects cannot spend them again because they will be physically moved to the receiver, i.e., the same coin or banknote cannot be physically in two places simultaneously. However, this is not guaranteed in the digital world, where there are no physical objects but only digital object representations. This problem is called *double-spending* and it happens when a digital transfer is not properly managed. The same digital currency could be spent multiple times by the same sender without the receivers immediately realizing it.

In traditional digital transfer payments, it is often necessary to rely on third-parties to ensure that double-spending did not occur. These parties are usually banks, payment processors (e.g., *Visa*, *Mastercard*, *PayPal*, ...), or other financial institutions. These intermediaries ensure the legitimacy of transactions, maintain transaction records, and handle disputes. Trust is placed in centralized institutions that prevent double spending by maintaining a central ledger that tracks all transactions and ensures that once currency is spent, it cannot be used again. Furthermore, they may require long verification times (even several days) to ensure the transfer of high-value economic assets or large amounts of money. In addition to setting arbitrarily fees and prices (credit card fees, bank transfer fees, currency exchange fees, increase fees for international transfers, ...).

Blockchain technology, not surprisingly, has become disruptive for digital transfer payments due to its ability to address these long-standing inefficiencies of traditional digital systems. It can reduce costs, speed up transactions, eliminate the need for intermediaries, and increase financial inclusion. Moreover, in correct settings, it can strongly mitigate the double-spending problem.

In particular, the blockchain has an abstract data structure shared in redundant copies among a trustless, decentralized, and distributed network based on transactions. This data structure is tamper-proof and organizes the data into blocks, each cryptographically linked to the previous one. Any alteration to a block would change its cryptographic hash, making it evident to the network that tampering has occurred, as subsequent blocks would no longer match. Additionally, blockchain functions as a ledger because it chronologically records and permanently stores all transactions, providing a transparent and auditable history of activities.

What manages this data structure is the *consensus mechanism*, such as Proof-of-Work (PoW) or Proof-of-Stake (PoS), that allows network participants to agree on the validity and consistency of new data, preventing unauthorized changes, ensuring trust among decentralized parties, as well as preventing the duplication of digital currency and mitigating double spending issues through a reward and penalty system that makes cheating inconvenient.

For instance, in PoW, the consensus is implemented by network peers called *miners*. A miner is required to solve complex computational puzzles, to have permission to propose a candidate block, which could be added to the chain. If the majority of the network successfully validates the candidate block, it is added to the chain and the miner is typically rewarded with transaction fees and potentially also with newly minted cryptocurrency, depending on the blockchain. Otherwise, the block is discarded without any reward, which also means losing the resources spent (e.g. energy, rental of powerful machines, etc.) to compute the puzzle, which will no longer be valid for other applications. Therefore, adding a double spend in the candidate block could lead to a failure of validation by the network, resulting in economic damage to the cheater. Instead, in PoS, there are no miners. Still, the consensus is managed by peers called *validators*, a subset of the network that freezes an amount of cryptocurrencies called stake to guarantee validation. In this case, there are no puzzles to solve, and the *validator* is chosen randomly with a probability weighted depending on the stake size. The higher the stake, the more likely one will be chosen for the validation. Conversely, it is also a greater risk of losing the stake. Indeed, if the block is successfully validated, it is added to the chain, and the validator receives the reward. Otherwise, if the transaction is not validated, the validator stake is slashed, losing part or all of the staked

cryptocurrencies through penalties. In this case, cheating is discouraged by the loss of the reward and by *slashing*, because to have a good chance of validation, one typically needs a high stake that can be lost if one gets caught cheating. The main differences between PoW and PoS are that the former relies on computational power to validate transactions and secure the network. At the same time, the latter uses cryptocurrency ownership to achieve the same, consuming less energy. The data to add in a candidate block are transactions with a payload sent by clients (e.g. crypto-wallets) to the blockchain network. These transactions are typically validated and collected in memory pools, which can be retrieved during the consensus phase. These transactions are called *unconfirmed transactions*. Once the candidate block is added into the blockchain data structure, it becomes *confirmed transactions* and only after a certain number of subsequent blocks linked to that block are considered finalized transactions, i.e. when the risks of reversion, alteration, and fork[2] are drastically reduced.

3. Related work

Double-spending is especially critical among blockchain attacks because it directly undermines the trust in digital currencies and blockchain technology.

Nakamoto [2] first highlighted this problem within Bitcoin, the first cash system based on blockchain. Since then, numerous studies have examined various double-spending attack strategies.

For instance, Liao and Katz [16] reported that attackers with significant off-chain economic incentives can alter consensus mechanisms by bribing consensus actors to approve malicious actions, such as to fork the blockchain in favour of fraudulent transactions. Game-theoretic analysis and simulations have demonstrated that this strategy can yield a positive payoff for the attacker. In another investigation, Biais et al. [17] showed that blockchains face significant coordination challenges, including multiple equilibria and the potential for forks when modelled as a stochastic game. Ethereum fork and SegWit2x are two such examples. Two models for double-spending attacks in Bitcoin exploiting the time advantage have been presented in Pinzón and Rocha [18]. The authors have argued that Bitcoin's consensus mechanism, based on Pow [19], allows trustless transactions. However, it remains susceptible to double-spending attacks, particularly when attackers accumulate time to mine fraudulent blocks. This time-based model, which considers block-mining times and the progress in chains mined already, differs from the generalised model built on Rosenfeld's work [20].

Another effective model for analyzing double-spending attacks is the quantitative framework introduced by Gervais et al. [21], which has been used to evaluate PoW blockchains with varying network and consensus parameters. This framework has allowed for a comprehensive assessment of security and performance by accounting for factors such as network propagation, block sizes, block generation intervals, and eclipse attacks. By applying this model, the authors have been able to objectively compare various PoW blockchains, revealing that for instance, Ethereum required 37 block confirmations to match Bitcoin's security with just 6 confirmations, and that PoW blockchains have been able to process up to 60 transactions per second without sacrificing security. Additionally, the framework has quantified the impact of stale block rates on resistance to double-spending attacks and selfish mining attacks[3], providing valuable insights into both merchants and miners. On the same note, Hinz [22] has addressed PoW-based blockchain systems and the significant financial losses caused by attacks like double-spending. The author has modeled these attacks as optimal sequential decision problems, considering both simplified and complex approaches. In the simplified model, attackers have engaged in secret mining, timing their payments based on chain length differences. In contrast, the complex model has introduced the option to abandon secret mining, factoring in mining costs and rewards. These models have proven useful for assessing vulnerabilities in PoW systems. However, further development is necessary to address propagation delays, observation uncertainties, and alternative consensus mechanisms. Additionally, the potential for malicious actions that deliberately slow down honest mining must be accounted for to improve the stability of PoW systems. These frameworks have provided complementary approaches for understanding and mitigating double-spending risks in PoW blockchains. A recent study using the Security Risk Management (SRM) model [23] developed a framework to explore two of the most concerning security risks in blockchain systems: Sybil and double-spending attacks. The framework has identified the assets to be protected, classified the threats

[2] A fork is a divergence in the blockchain split into separate paths due to differences in software rules, updates, or accidental conflicts

[3] A selfish mining attack occurs when a miner withholds newly mined blocks to gain a strategic advantage, creating a personal chain to outpace the original chain and increase their share of mining rewards.

triggered by a Sybil attack, recognised factors causing double-spending attacks, assessed the vulnerabilities of these threats, and suggested appropriate counter-measures. The study has evaluated this framework by examining Sybil and double-spending risks in Ethereum-based healthcare applications. Additionally, it discusses how permissioned blockchain systems address these challenges and outlines future research directions. In Zheng et al. [24], two new types of double-spending attacks in the context of PoW blockchains, highlighting the ongoing risks to blockchain security, have been discussed. Unlike conventional double-spending attacks, this study introduces Adaptive Double-spending (ADS) and Reinforcement Adaptive Double-spending (RADS) models, which incorporate more practical network parameters such as confirmation blocks, puzzle complexity, transaction value, and network status. These models convert the double-spending attack into a Markov decision process and use stochastic dynamic programming to determine optimal attack strategies. Numerical simulations show that an attacker can successfully launch a double-spending attack with much less than 51% hashpower under the RADS model. Finally, a double-spending attack that exploits both mining power and connectivity in blockchain systems like Bitcoin and Ethereum, a.k.a. Balance attack, has been proposed in Natoli et al. [25]. This attack only requires delaying specific messages and targets the synchrony assumption of blockchain protocols, where transmission delays are expected to be within a known upper bound. The attack allows an attacker with low mining power to mislead subgroups of honest nodes, resulting in double-spending by convincing them of a valid chain while disregarding others. The study has demonstrated that the attack succeeds with high probability and can achieve double-spending in under 20 minutes. It has also validated the attack empirically by using 'geth' software and hijacking Border Gateway Protocol to reroute traffic in a controlled system.

The current literature on cross-chain bridge security reveals a growing concern around the vulnerabilities inherent in blockchain interoperability. In their paper, Zhang et al. [26] provide one of the most comprehensive systematization's of security issues affecting cross-chain bridges. The authors present a taxonomy derived from 35 real-world attacks, categorizing them into ten types of vulnerabilities. Their analysis identifies critical attack surfaces such as permission, logic, and front-end issues, and proposes structured mitigation strategies supported by detailed Solidity code examples and empirical case studies. In contrast, Cheng et al. [27] focus on improving cross-chain efficiency and security through a decentralized interaction model that leverages sidechains and hash-time-locked contracts (HTLC). This approach eliminates centralisation risks and enhances transaction speed by avoiding reliance on third-party validators. Olivieri et al. [28] highlight the need for adequate cross-chain contract verification tools. Mishra et al. [29] explore broader blockchain security challenges, with a particular emphasis on layered vulnerabilities within cross-chain protocols. The authors emphasize the importance of advanced detection techniques, including deep learning models, to identify and respond to emerging threats in smart contracts and cross-chain bridge infrastructures. Collectively, these studies illustrate the early but evolving nature of cross-chain bridge security research, highlight the pressing need for and effective measures to detect, counter, and prevent blockchain interoperability issues.

Although cross-chain interoperability is increasingly vital for decentralized ecosystems, cross-chain security remains underrepresented in the academic literature. Compared to single-chain systems, cross-chain protocols introduce additional layers of complexity and a broader attack surface. In single-chain environments, security typically relies on well-established consensus mechanisms like PoW or PoS. However, cross-chain solutions such as bridges and relays often depend on semi-trusted entities, including federated multisignature schemes or external oracles, which introduce new vulnerabilities. For example, high-profile exploits like the Wormhole and Ronin bridge hacks, which resulted in losses exceeding 600 million combined, underscore the inherent risks of inadequate verification and trust assumptions. Unlike single-chain systems with unified consensus and auditable logs, cross-chain protocols must reconcile differing consensus models and transaction histories, complicating verification and increasing the potential for inconsistencies. Comparisons among systems such as Polkadot's relay chain (which centralizes security via a shared consensus), Cosmos' Inter-Blockchain Communication (IBC) protocol (which emphasizes sovereignty and light client proofs), and Ethereum-based bridges (which often rely on off-chain validators) highlight the trade-offs between decentralization, efficiency, and security. These differences suggest an urgent need for comprehensive frameworks to evaluate cross-chain security with the same rigor applied to intra-chain systems.

Currently, the state-of-the-art related to double-spending studies is primarily confined to individual blockchains, and to the best of our knowledge, does not pay particular attention to multiple blockchain interactions, lacking to consider cross-chain environments. In particular, inter-chain interactions' inherent complexities and vulnerabilities with the absence of a unified consensus protocol and reliance on trustless or semi-trustless mechanisms to transfer assets between blockchains may create additional attack surfaces.

4. Double-spending Attacks Relevant to Cross-Chain Contexts

This section investigates classical consensus and network-layer double-spending attacks [23] and assesses their applicability, feasibility, and practical challenges in cross-chain environments. Furthermore, it also considers real-world scenarios that share similar characteristics.

4.1. 51% *attack*

A 51% attack [30] requires that an attacker gains control of over 51% of a blockchain network's computing power in PoW, which is typically referred to as the *hashrate* (the total computational power used to process transactions and secure the network), or the 51% of validation power in PoS. With over 51% control, the attacker can manipulate the blockchain's consensus process, allowing them to double-spend currencies, rewrite transaction history, or disrupt the network. For instance, an attacker like Bob could use his control over 51% of the network's computing power to manipulate transactions. Suppose Bob initiates a transaction, spending currency. Since he controls the majority of the network's computing power, he can later rewrite the blockchain's history by creating a new version of the chain that does not include the initial transaction. This allows him to "reverse" his transaction effectively, reclaiming the spent funds. In this case, the counterparty who received the currency in the transaction would be left with nothing, as the blockchain history has been altered. This manipulation does not require creating a new blockchain; the attacker uses their majority control to change the transaction history arbitrarily, enabling double-spending or disrupting the network.

While major networks like Bitcoin and Ethereum have become more resistant due to high hash power, smaller networks, such as Ethereum Classic [31], Bitcoin Gold [32], and Verge—have been more vulnerable to this attack. Besides double-spending, a 51% attack can cause selfish mining, transaction delays, and blockchain forks.

Mitigation strategies in a single blockchain environment include monitoring node computing power, increasing the number of confirmations, limiting large transactions, and incentivising honest mining through transaction fees. Delayed Proof-of-Work (dPoW) has enhanced security by notarising blocks across multiple blockchains, making history immutable, and reducing vulnerability to 51% attacks.

In cross-chain environments, a 51% attack scenario can be demonstrated through a cross-chain bridge attack between Blockchain A and Blockchain B, as illustrated in Figure 1. The attacker, Bob, can leverage the system by manipulating cross-chain bridge transactions and initiating a chain reorganization, ultimately leading to a double-spend attack. This attack involves the attacker gaining control of the computing power in one of the chains, using it to reorganise, and exploiting the cross-chain bridge to double-spend funds across the linked blockchains. Below, we have detailed the steps followed in a 51% attack in a cross-chain environment.

Considering the classic scenario of a cross-chain token transfer via *Lock and Mint* [33] between two blockchains, chain A and chain B:

1. Locking a token TK on Chain A: Bob (the attacker) begins by sending a token TK to a cross-chain bridge smart contract on Chain A. This action locks the token TK, a standard step for cross-chain transfers.
2. Minting wTK on chain B: Upon locking the token TK on Chain A, an equivalent wrapped TK (wTK) is minted on Chain B. This is the mechanism that enables cross-chain asset representation.
3. Receiving and using wTK on chain B: Bob receives the newly minted wTK on Chain B and can use it as he wishes on this chain.
4. Selling or swapping wTK: Bob quickly sells or swaps the wTK on Chain B, converting it to other assets or tokens.
5. Creating a fork and reversing the transaction: Bob then creates a fork of Chain A and reverses the transaction that initially locked his TK. This manipulation results in the TK being restored to Bob's possession on Chain A, as if the transaction never occurred.
6. Original TK regained: With the reversal on Chain A, Bob now has access to his original TK on Chain A, despite having already received and used the wTK on Chain B.
7. wTK on chain B still in circulation: The wTK minted and possibly sold on Chain B is circulating in the market, creating an imbalance or potential vulnerability within the cross-chain bridge system.

Feasibility : In the original analysis, we stated that an attacker would require control over 51% of the network's hash power to successfully execute a double-spending attack in PoW blockchains. However, recent theoretical work has demonstrated that attackers with substantially lower proportions of computational power can also profit from mining-based attacks. For instance, Eyal et al. [34] introduced the *Selfish Mining* strategy, which showed that: (i) attackers with just over 25% of the total network hash power could earn more than their fair share of mining rewards; (ii) profitability becomes guaranteed when the attacker controls approximately 33% of the total hash power, assuming standard network propagation conditions. The expected revenue R for an attacker with mining power fraction α is given by:

$$R = \frac{\alpha(1-\alpha)^2}{1-\alpha(1+\alpha)}$$

Instead, Gervais et al. [21] developed a quantitative model relating confirmation depth (k) to attack success probability (P):

$$P \approx \left(\frac{q}{1-q}\right)^k$$

where q is the attacker's hash rate fraction. This model illustrates that the probability of attack success grows exponentially with increasing confirmation depth and network propagation delays.

Practical Challenges: Despite theoretical feasibility, executing such attacks in cross-chain environments faces significant practical challenges. Coordinating attacks across multiple independent blockchains, each with distinct consensus mechanisms, requires substantial technical expertise and logistical coordination. Furthermore, beyond technical barriers, there are some economic risks, such as loss of trust in the ecosystem and devaluation of associated assets serving as strong deterrents. In addition, observer networks and fraud detection mechanisms significantly reduce the window for successful exploitation. The presence of diverse independent stakeholders across chains improves transparency and makes detection likely, rendering large-scale attacks highly mitigate in real-world scenarios where are involved interaction between large and consolidated blockchains. However, it cannot excluded in scenarios, where there is a disparity in terms of economic and social environments. For instance, cases of interoperability between minor blockchains and consolidated ones. Indeed, blockchains may vary in token value, transaction fees, and reward mechanisms. This may undermine the socio-economic balance that typically discourages malicious behaviours in the blockchain system and consequently leads to increased risks of double spending. For example, the majority of a minor blockchain could decide to bring the blockchain itself to collapse in favor of greater profit on the more consolidated blockchain, making fraudulent actions.

Real-World Scenario: Ethereum Classic 51% Attack (January 2020) [35]. **Similarity :** The Ethereum Classic attack is a example of a 51% double-spending attack that occurred on a single chain. The attacker gained majority control over the network's hash rate, allowing them to reorganize the blockchain, reverse transactions, and double-spend funds. This aligns directly with the core mechanism of a 51% attack, where an entity with sufficient hash power can unilaterally decide the blockchain's state, effectively invalidating previously confirmed transactions .

4.2. Sybil attack

In a Sybil double-spending attack [36], the attacker creates multiple fake nodes (aka *sybil nodes*) to manipulate the network's communication and block propagation. The attacker broadcasts a legitimate transaction $AT0$, spending funds, and simultaneously creates a conflicting transaction $AT1$ that attempts to spend the same funds again. The fake nodes delay the propagation of the legitimate transaction, allowing the attacker to mine a longer chain containing $AT1$. When this longer chain is accepted as the valid chain, it reverses the legitimate transaction $AT0$, effectively allowing the attacker to double-spend. This attack exploits network delays and manipulation, rather than controlling most of the mining power as in a 51% attack. It is a form of double-spending enabled by fake nodes disrupting the chain's consensus process. Although Sybil attacks can lead to a 51% attack if the attacker controls enough fake nodes, consensus mechanisms like Pow and PoS mitigate this risk by making it costly to control enough nodes or staked tokens.

Potential solutions include preventing the addition of consecutive candidate blocks, increasing the number of confirmations, and using identity fees to restrict the creation of Sybil nodes to mitigate this in a single blockchain.

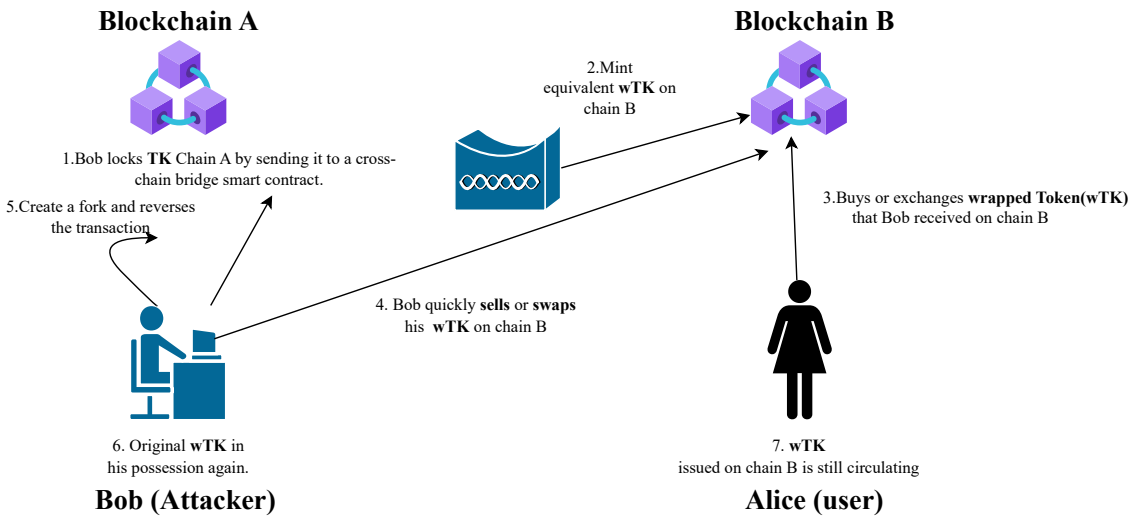

Figure 1. 51% double-spending attack

In a cross-chain environment involving Blockchain A and B, connected via a cross-chain bridge, an attacker launches a Sybil attack using multiple fake identities (Sybil nodes). This disrupts the transaction process across the bridge, allowing for a double-spend across both blockchains. Figure 2 visualizes this Sybil attack within the cross-chain context. Below, we have provided a step-by-step breakdown of the process.

1. Transaction initiation: The attacker starts by sending tokens to a smart contract on Blockchain A. This action initiates the cross-chain transaction, a typical step in moving assets across blockchains.
2. Validation by honest validators: Honest validators on Blockchain A validate this transaction. They confirm the transaction details and send them to the cross-chain bridge, following the standard protocol for cross-chain asset transfers.
3. Bridge relays transaction details: The cross-chain bridge receives and relays the transaction details from Blockchain A to Blockchain B. This step is necessary to trigger the process of minting or unlocking equivalent tokens on Blockchain B.
4. Sybil nodes validate minting/unlocking: The attacker employs Sybil nodes (fake identities controlled by the attacker) to validate the minting or unlocking process on Blockchain B. Since the Sybil nodes act quickly, they bypass the need for consensus from honest validators, securing the newly minted or unlocked tokens on Blockchain B.
5. Attacker holds tokens on chain B: Due to Sybil node validation, the attacker now possesses the equivalent tokens on Blockchain B, as if the transfer was legitimately completed.
6. Double-spending on chain B: The attacker uses Sybil-validated tokens to perform multiple transactions on Blockchain B, exploiting duplicate tokens generated through the cross-chain process. Note that, this is not possible on Blockchain A, as its transactions are final and immutable once validated by honest validators, preventing any manipulation or rollback.

Feasibility : The feasibility of Sybil attacks in cross-chain systems depends on the economic and technical capacity to create or control multiple validator identities. In PoS networks, gaining control of at least 33% of the total stake enables attackers to manipulate consensus processes, particularly in relayer validation for cross-chain token transfers. Game-theoretic models [37] suggest that such an attack becomes economically viable when the expected gains exceed the penalties imposed by slashing mechanisms. For example, in a system with a total value locked (TVL) of $5 million, an attacker controlling $1.65 million in stake could potentially execute profitable censorship or transaction reordering unless countermeasures such as robust slashing policies and checkpoint mechanisms are in place.

Practical Challenges: Despite theoretical feasibility, cross-chain Sybil attacks face significant practical challenges that diminish their likelihood compared to single-chain scenarios. Cross-chain ecosystems inherently involve decentralized operations, where validators or relayers operate across multiple, independently governed blockchains. Unlike single-chain systems, cross-chain environments lack a unified consensus mechanism, making it exceedingly difficult for attackers to gain substantial influence simultaneously across all involved networks. Furthermore, most cross-chain systems incorporate robust cryptographic safeguards (e.g., Merkle proofs, zk-SNARKs) and enforce economic disincentives such as slashing or bonding requirements, effectively deterring malicious behavior. The financial burden of acquiring sufficient stake across multiple blockchains, combined with the operational complexity of coordinating attacks on separate networks, renders successful cross-chain Sybil attacks largely impractical. As a result, these attacks remain unlikely in real-world scenarios with large and consolidated blockchains.

Real-World Scenario: Arbitrum Airdrop Sybil-Farming Attack (2023) [38]. **Similarity :** The Arbitrum Airdrop Sybil-farming attack involved creating over 1,000 fake wallets to exploit the airdrop mechanism and claim free tokens. The attack targeted a single chain (Arbitrum) to exploit the airdrop mechanism. While not a monetary double-spend attack in the traditional sense, it demonstrates the core Sybil attack principle: exploiting multiple fake identities to manipulate a system. In a double-spending scenario, a Sybil attack could achieve similar outcomes by creating false validators or nodes to override consensus and authorize fraudulent transactions.

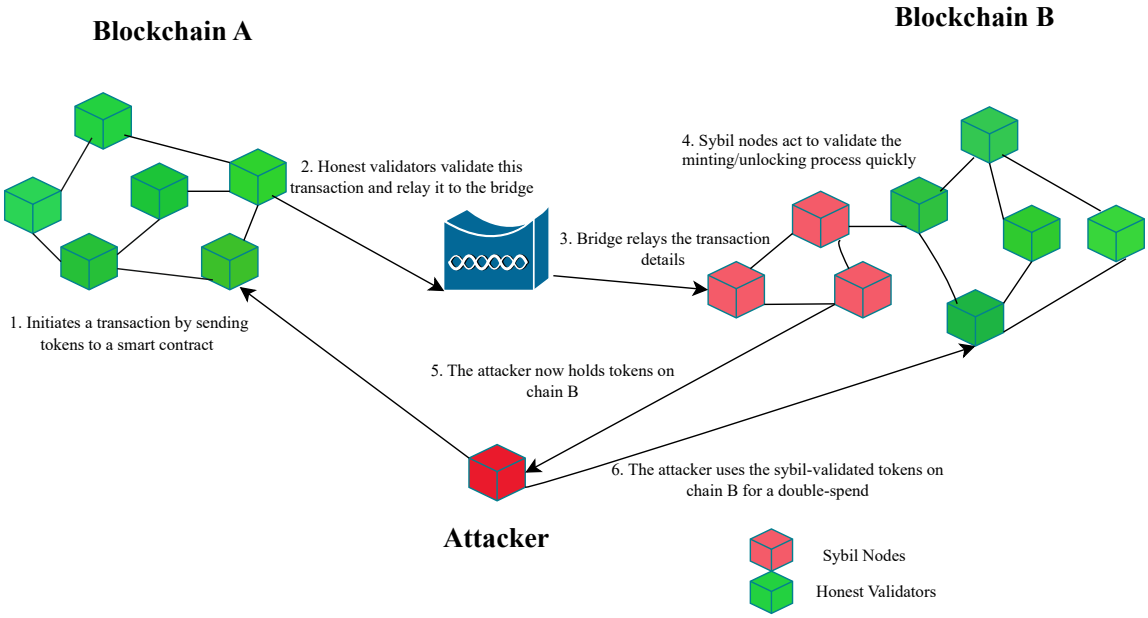

Figure 2. Sybil-based double-spending attack

4.3. Eclipse attack

An eclipse attack [39] targets specific nodes in a blockchain network by isolating them from honest peers exploiting the limited number of connections each node can maintain. The attackers can control the victim's view of the network, feeding it manipulated information or withholding critical updates. This isolation has far-reaching consequences beyond misinformation. In PoW systems, miners can be excluded from participating in block creation and rewards, disrupting the network's consensus process. Similarly, in PoS systems, validators can be prevented from performing their roles, compromising the network's security and fairness. Eclipse attacks, therefore, not only deceive individual nodes but can also weaken the overall integrity and functionality of the blockchain.

Eclipse-based double-spending exploits the isolation of nodes, particularly during fast payments involving *zero-confirmation (*0*-confirmation)* or *N-confirmation* transactions. In a 0*-confirmation attack*, the attacker isolates a mer-

chant's node and sends an unconfirmed transaction (referred to as $AT0$) to the merchant. While the merchant, unaware of the isolation, accepts $AT0$ and releases goods or services, the attacker simultaneously creates a conflicting transaction (referred to as $AT1$) that double-spends the same inputs used in $AT0$. The attacker broadcasts $AT1$ to the rest of the network, which, due to the merchant's isolation, processes and confirms $AT1$ while discarding $AT0$. As a result, the merchant loses the goods or services provided in exchange for $AT0$, which is ultimately invalidated by the network.

In an *N-confirmation attack,* the attacker isolates the merchant and some network validators. The merchant waits for $N-1$ confirmations from the isolated group before releasing goods. The attacker then reveals the true blockchain state, orphaning the isolated confirmations and invalidating the transaction, allowing them to keep the goods without payment.

Mitigating the above attacks includes waiting for more confirmations (for example, 10 blocks), which slows transactions. Other solutions involve turning off direct incoming connections, using allowed nodes for outgoing connections, randomising outgoing address selection, and using feeler and anchor connections to maintain secure and rotating peer relationships. These techniques help prevent isolation and reduce the risk of eclipse-based double-spending.

In a cross-chain context, as illustrated in Figure 3, an Eclipse Attack occurs when an attacker isolates a victim node on one blockchain (Chain B) and manipulates its perception of transactions between interconnected blockchains (e.g., Chain A and Chain B). The interconnection between Chain A and Chain B is typically facilitated by cross-chain bridges, which may involve smart contracts, relayers, or oracles that manage the locking, minting, and transfer of assets or data across the chains.

By controlling the victim's network connections, the attacker can present fraudulent transactions that enable double-spending across both chains, thereby undermining the integrity of cross-chain transactions. Below is a step-by-step breakdown of the process:

1. Identifying vulnerable nodes: The attacker begins by identifying a group of nodes on Blockchain B that are less connected to the rest of the network. These nodes are more susceptible to isolation and manipulation.
2. Isolating the targeted group: The attacker successfully isolates these nodes by controlling their network connections, ensuring they interact only with attacker-controlled nodes rather than with the legitimate network.
3. Initiating a cross-chain transaction: The attacker initiates a valid transaction on Blockchain A by sending tokens to a smart contract associated with a cross-chain bridge. This action triggers the cross-chain transfer process, signalling that the transaction data should propagate to Blockchain B for further validation.
4. Creating a conflicting transaction on chain B: On blockchain B, the attacker uses an isolated group of nodes, manipulated via network isolation, to validate and propagate a conflicting transaction. This isolated group operates without knowing the legitimate state of Blockchain A. Because this isolated node has a manipulated view of the network, it validates the fraudulent transaction of the attacker without knowing the actual state of Blockchain A.

Feasibility : Quantitative evidence from Heilman et al. (2015) [40] illustrates that isolating a node can be achieved with 4–8 attacker-controlled IP addresses under typical conditions. The probability of success, P(eclipse), increases significantly when targeting nodes with lower peer diversity, such as bridge, relayers, or lightly connected validators:

$$P(\text{eclipse}) \approx \frac{N_m}{N_t}$$

where N_m is the number of malicious peers, and N_t is the total peer connections. The simulations show that success rates of 60–70% are achievable against cross-chain nodes with insufficient network randomness. Thus, eclipse attacks represent a severe and quantifiable threat in cross-chain ecosystems.

Practical Challenges: Eclipse attacks isolate specific nodes in a peer-to-peer network by redirecting their connections to attacker-controlled nodes, effectively cutting them off from legitimate network activity [41]. Unlike Sybil attacks, which involve flooding the network with fake nodes, Eclipse attacks focus on isolating individual nodes. In cross-chain environments, attackers could target high-consensus-power nodes, such as validators, bridges, or relayers, to manipulate their view of the blockchain or disrupt inter-chain communication. By isolating these critical nodes, attackers can execute attacks such as double-spending or censorship without requiring control of the majority of the

network. This makes Eclipse attacks particularly feasible in cross-chain scenarios, where communication randomness and peer diversity are often limited.

Real-World Scenario: Multichain Hack (July 2023) [42], Ethereum Classic attacks (2020) [43]. **Similarity:** The Multichain hack (July 2023), which resulted in over $126 million in losses, exploited compromised MPC keys to drain assets across several connected blockchains [44]. Similarly, the 2020 Ethereum Classic attacks showcased how isolated network control allowed attackers to reorganize the chain and double-spend funds [43]. These incidents mirror the core mechanism of eclipse-based double-spending attacks, as described by Marcus et al. [39], where adversaries isolate nodes or validators to manipulate local consensus. In each case, attackers either achieved full or partial network isolation—whether by compromising validation keys, suppressing peer communication, or exploiting weak network structures—allowing them to propagate fraudulent transactions or reorganized chains undetected. These real-world scenarios demonstrate how network-level or infrastructure-level control can substitute for traditional eclipse attacks by similarly enabling malicious state divergence and cross-chain double-spending.

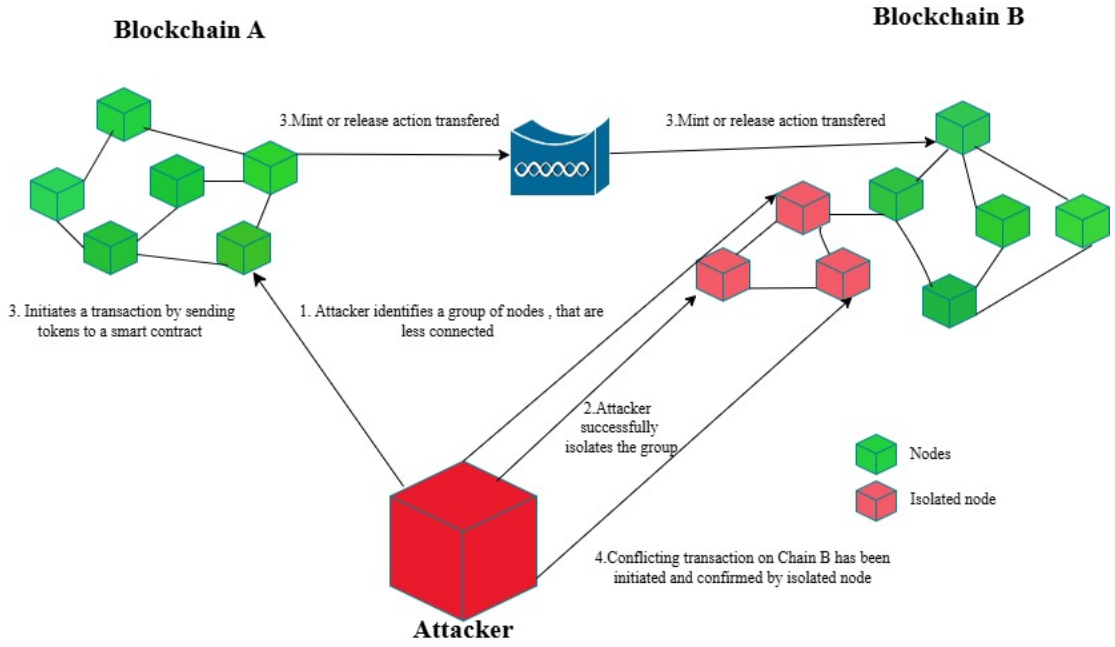

Figure 3. Eclipse-based double-spending attack

4.4. Finney-based attack

The Finney attack [45] has been characterised as a 1-confirmation pre-mining attack where the attacker mines a block with a transaction to itself ($AT0$) but withheld broadcasting it. They then send a second conflicting transaction ($AT1$) to a merchant to pay for goods. Once the merchant has accepted the transaction, the attacker releases the pre-mined block, invalidating $AT1$ and reclaiming its funds, leaving the merchant unpaid. The attack can fail if another block has been mined before the attacker broadcasts its block, with the failure probability being t/T, where t is the time between finding the block and completing the payment, and T is the average time to find a block. While not directly hash rate-dependent, a lower hash rate reduces the likelihood of success. Waiting times for large transactions are logarithmically related to the chain length, as demonstrated by simulations related to the 'Gambler's ruin' problem [45], where the probability of success increases with the attacker's mining power.

Counter-measures include waiting for more block confirmations, especially for large transactions. Techniques such as listening periods, observer nodes, and alerting nodes, commonly used for 0-confirmation race attacks, have also mitigated Finney attacks [46].

A Finney attack in a cross-chain environment occurs when an attacker pre-mines a block on blockchain A with transaction $AT0$, then uses the same tokens in a conflicting transaction $AT1$ on blockchain B. After the transaction

$AT1$ is confirmed, the attacker broadcasts the pre-mined block on blockchain A, invalidating transaction $AT1$ and allowing double-spending across both blockchains. We have illustrated this attack in Figure 4, where an attacker (Bob) exploits cross-chain transactions for double-spending. The steps in the process have been listed belo

1. Pre-mining a block (on blockchain A): The attacker, Bob, pre-mines a block on blockchain A containing $AT0$, which is kept hidden from the network.
2. Initiating a cross-chain transaction (on Blockchain B): Bob uses the same tokens from $AT0$ to initiate a second transaction ($AT1$) on blockchain B, which is accepted as valid since $AT0$ has not been broadcasted.
3. Receiving confirmation and service: $AT1$ is confirmed on blockchain B, and Bob receives the service or asset based on this fraudulent transaction.
4. Broadcasting the pre-mined block (on Blockchain A): After receiving the service on chain B, Bob broadcasts the pre-mined block containing $AT0$ on Blockchain A, making it valid and invalidating $AT1$, resulting in a double-spend.

Feasibility: The Finney attack relies on mining power and confirmation practices to succeed. The probability of success for an attacker with a mining share q engaging in pre-mining is expressed as:

$$P(\text{success}) \approx \frac{q}{1-q}$$

For instance, if $q = 0.1$ (10%), the success probability for targeting 0-confirmation transactions is approximately 11%. However, requiring even a single confirmation significantly reduces the attack's feasibility [47, 40].

Practical Challenges: Finney attacks are predominantly relevant to PoW networks like Bitcoin and Ethereum 1.0. However, they are basically impractical in PoS systems, such as Ethereum 2.0, Cosmos, or Tezos. The attack requires an attacker to mine a block containing their fraudulent transaction while also identifying a merchant or system willing to accept a zero-confirmation transaction [40]. In practice, the likelihood of success is low due to stricter confirmation mechanisms employed by most networks. While theoretically possible with less than 51% of the network's hash power, PoS systems introduce validator nodes that preemptively verify transactions before block inclusion, making this attack infeasible [37]. Conversely, PoW networks rely on supervisor nodes for transaction validation, which may not enforce confirmations as rigorously, presenting a limited but real vulnerability.

Real-World Scenario : Neptune Mutual "Miner" Exploit (February 2024) [48]. **Similarity:** it exploited a bug in the `_transfer` function that allowed "self-transfer" transactions. The "Miner" exploit targeted smart contracts on a single blockchain. These transactions subtracted and added the same value back to the sender's balance, effectively doubling tokens without any real value transfer. Similarly, a Finney attack exploits validation inconsistencies in blockchain networks, allowing attackers to execute duplicate or conflicting transactions before consensus can reject them. Both attacks share a core issue: insufficient validation of transaction logic, which enables duplicate issuance or spending of tokens.

4.5. Time advantage-based attack

This attack extends the 51% attack by using both time and hash power to secretly mine fraudulent blocks for double-spending. The attacker mines ahead of honest nodes, disrupting the network to steal assets. This attack is usually based on two models.

- Generalized model: By including a time component in the mining process, the generalized model allows an attacker to secretly mine blocks with a sufficient time advantage to accomplish double-spending.
- Ledger state model: Both the attacker and honest nodes' most recent block mining times are taken into account by the time-based ledger state model.

The attack's success relies on time advantage, computing power, and block confirmations. In addition to time advantage, a double-spending attack is also feasible if the attacker's processing capacity improves or the number of confirmations decreases. In blockchain systems with great processing power (like Bitcoin and Ethereum), this scenario is extremely uncommon, but it is feasible in those with low processing power.

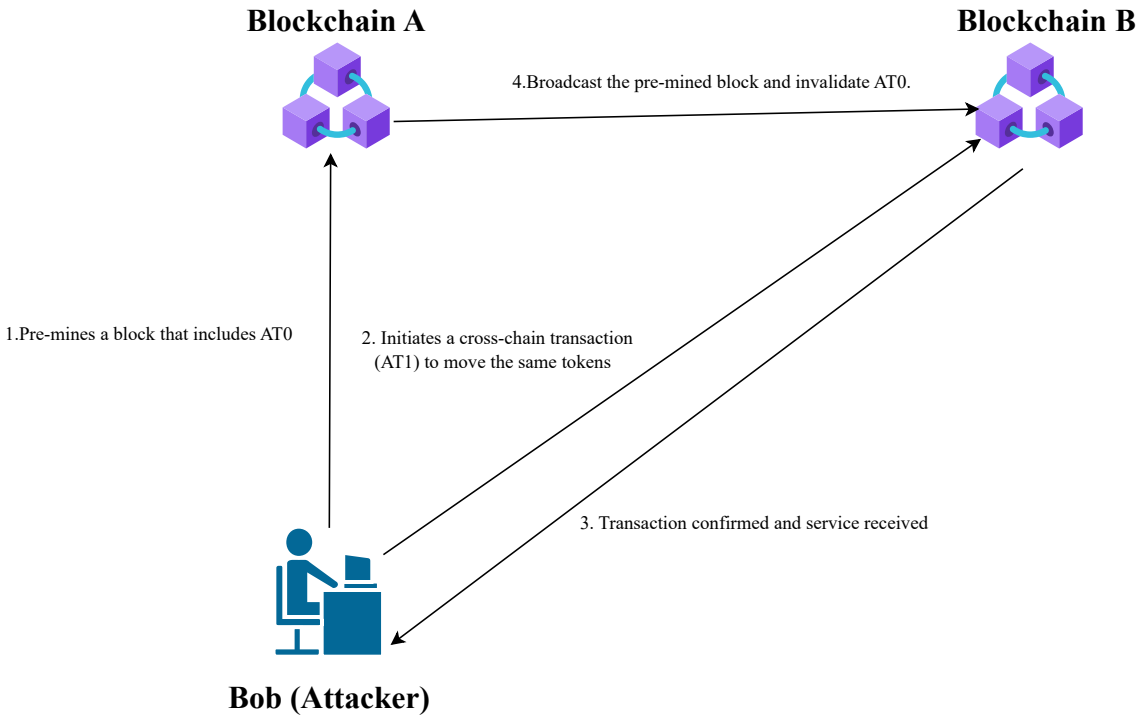

Figure 4. Finney-based double-spending attack

Mitigation strategies in single blockchain includes monitoring block times, limiting time advantage, and increasing confirmations. Implementing a system to limit the time advantage when the attacker and the honest miner last mined a block and monitoring time disparities after the new block is produced have been suggested as ways to combat time advantage attacks. Increasing the quantity of block confirmations to make it more costly for the attacker to execute the attack, is another strategy to prevent this attack.

A time advantage attack in a cross-chain environment occurs when an attacker exploits differences in block times or propagation delays between two blockchains to double-spend assets. The attacker mines blocks faster on blockchain A and manipulates cross-chain transactions to spend the same assets twice—first on blockchain A, then on blockchain B by taking advantage of the time gap. The attack has been illustrated in Figure 5, where an attacker (Bob) exploits cross-chain transactions for double-spending. Here is a detailed breakdown of the process.

1. Initiate a transaction (blockchain A): The attacker (Bob) begins by creating a transaction on blockchain A, locking the funds.
2. Lock funds in blockchain A: The funds in blockchain A are locked after the transaction is initiated.
3. Initiate another transaction (blockchain B): Bob initiates a second, conflicting transaction on blockchain B, aiming to use the same funds.
4. Secretly mine a parallel chain:
   (a) The attacker mines a hidden chain containing fraudulent transactions to maintain control over the double-spending process.
   (b) Honest nodes in blockchain B propagate the legitimate chain, confirming it as the valid chain for blockchain B.
5. Broadcast the fraudulent chain:
   (a) The attacker releases the secretly mined fraudulent chain to override the legitimate chain.
   (b) Funds from blockchain B are now accessible to the attacker, potentially allowing double-spending.

6. Invalidate the transaction: The fraudulent chain causes the original transaction on blockchain B to become invalid, and the attacker reclaims their funds.

Feasibility : Time advantage-based attacks are an extension of selfish mining, exploiting discrepancies in block propagation times to gain an advantage over the network. The success probability of such attacks is described by Rosenfeld's generalized formula:

$$P(\text{success}) \approx \left(\frac{q}{1-q}\right)^z$$

Here, q represents the attacker's share of the network's mining power, and z is the number of consecutive blocks the attacker mines successfully. For example, if $q = 0.25$ (25%) and $z = 3$, the success probability can reach approximately 20% in slow or underpowered blockchain networks. Systems such as cross-chain bridges that rely on low-confirmation mechanisms are particularly vulnerable, especially when block propagation speeds are inconsistent.

Practical Challenges: Time advantage attacks pose significant threats primarily in single-chain public blockchains such as Bitcoin and pre-Ethereum 2.0 systems [18]. These attacks exploit discrepancies in network latency and processing times across nodes, using strategies like selfish mining and block withholding to gain a temporary advantage.However, in cross-chain environments, the feasibility of such attacks is substantially reduced. Cross-chain systems typically involve coordinated consensus across multiple chains, increasing detection latency and reducing the impact of localized propagation delays. Moreover, inter-chain communication protocols often include verification steps or checkpoints that mitigate the effectiveness of time-based manipulation strategies. As a result, attackers cannot easily exploit timing discrepancies without being detected across one or more chains.

Real-World Scenario: Binance Smart Chain (BSC) "Flash Loan Manipulation" Exploit (October 2021) [49]. **Similarity:** The Binance Smart Chain exploit involved an attacker leveraging a flash loan to manipulate the time delay in oracle updates on a single chain. The attacker exploited the lag between price updates to execute transactions based on outdated data, effectively draining funds. Similarly, a time-advantage based double-spending attack exploits delays in transaction finality or updates, enabling malicious actors to manipulate state before consensus catches up .

4.6. PoS long-range-based attack

A PoS long-range attack [50] is said to occur when an attacker has created a parallel chain starting from the genesis block, forging blocks with double-spend transactions to eventually overtake the main chain. The key types include:

- Simple long-range attack: The attacker manipulates timestamps to grow their chain faster than the main chain.
- Posterior corruption: The attacker uses private keys from retired validators to sign blocks on the hidden parallel chain.
- Stake bleeding: The attacker stalls block production on the main chain while building their hidden parallel chain, eventually surpassing it.

In a single chain, mitigation strategies for this type of double-spending attack include several vital approaches. For instance, context-aware transactions bound other transactions to specific time durations, reducing the risk of duplication. The economic finality penalises misbehaving validators financially, creating a strong deterrent. Moreover, key-evolving cryptography introduces a layer of security by requiring new keys for each validation round, limiting the usefulness of compromised keys. Checkpointing restricts block reorganization, helping maintain the integrity of the blockchain's history. Trusted execution environments (TEE) safeguard private keys within secure hardware, protecting them from exposure. Finally, the plenitude rule has leveraged block density to identify the main chain, though it may become ineffective if more than 34% of validators have been malicious. All these methods have reduced, but have not eliminated, PoS long-range attack risks.

Cross-chain token transfers in a PoS blockchain are vulnerable to long-range attacks when malicious actors control old stakes and manipulate the blockchain's history. In this scenario, an attacker with access to a cross-chain system initiates a double-spending scheme by exploiting validators' trust in the longest chain. The process depicted in Figure 6 is explained below.

1. Stake accumulation: The attacker amasses a significant amount of old stake in the blockchain network.

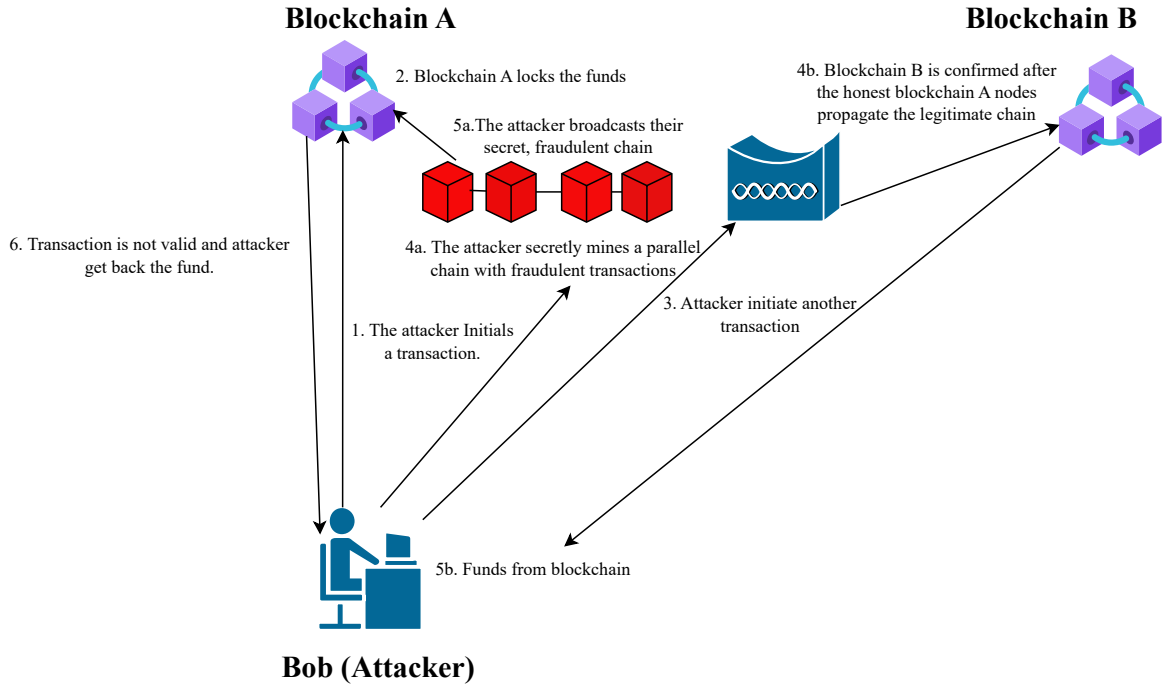

Figure 5. Time advantage-based double-spending attack

2. Alternative chain creation: The attacker creates an alternative blockchain from a historical point.
3. Smart contract usage: The attacker locks tokens in a smart contract meant for cross-chain transfer.
4. Initiating a fraudulent transaction: The attacker initiates a fraudulent transaction on this alternate chain.
5. Legitimate chain confirmation: Honest nodes propagate the legitimate chain (blockchain A), confirming transactions there.
6. Secret chain propagation: The attacker broadcasts their hidden, fraudulent chain (blockchain B) after it has overtaken blockchain A in proof or validation.
7. Final fraudulent confirmation: The fraudulent chain (blockchain B) becomes the accepted chain, allowing the attacker to gain control or funds while invalidating the legitimate blockchain activity.

Feasibility : Long-range attacks in PoS systems typically require control over $\geq$ 33% of the historical stake or access to outdated validators' private keys. Modern PoS blockchains mitigate these threats using checkpointing [51] and key-evolving cryptography [52], making such attacks practically infeasible. The probability of attack success increases if historical validator key material is not properly secured on legacy systems.

Practical Challenges: While long-range attacks may be theoretically possible on isolated PoS blockchains [53], cross-chain environments introduce additional verification layers, strong finality guarantees, and checkpointing mechanisms that significantly increase the complexity of executing such attacks [54]. These features collectively enhance resistance against long-range attacks across multiple chains. Modern cross-chain protocols are specifically designed to mitigate such risks by enforcing secure synchronization and consistent state finality among participating blockchains [55].

Real-World Scenario: Ethereum Classic Long-Range Attack (January 2020) [35]. **Similarity :** The Ethereum Classic long-range attack exploited stale validator keys to re-introduce an outdated chain history. The attacker used these keys to create a new version of the blockchain from far in the past, which was then propagated to unsynchronized nodes. This enabled the attacker to double-spend tokens that had already been used. The attack mimics the PoS long-range attack model described in the paper, where forged or stale validator keys are used to manipulate chain history

and deceive validators or nodes.

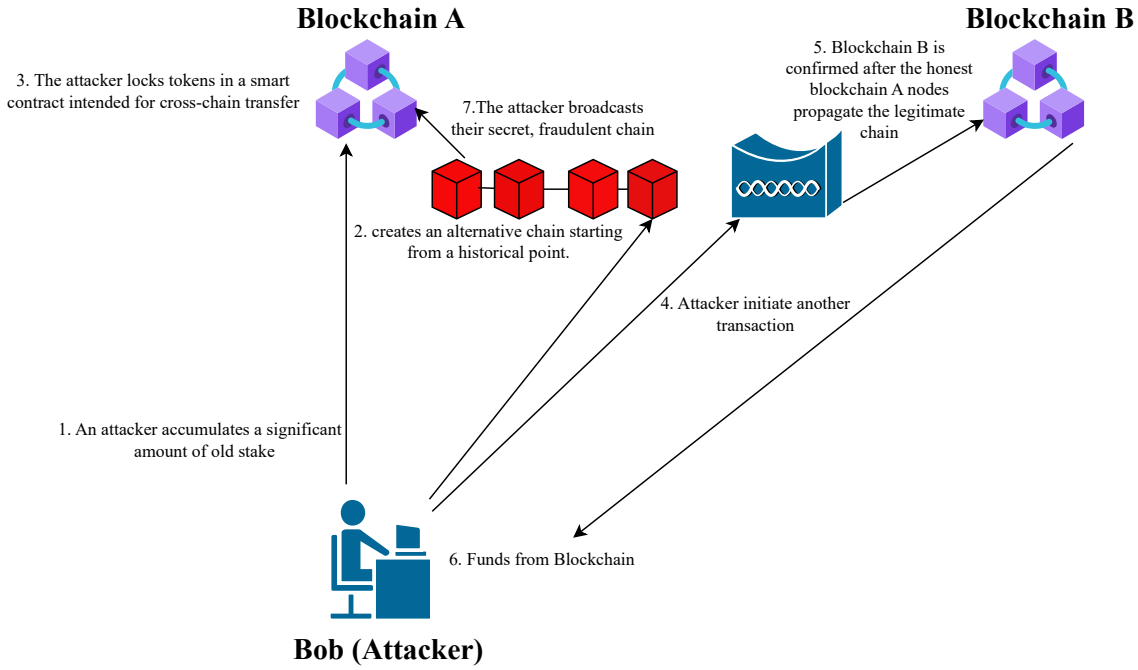

Figure 6. PoS long-range-based double-spending attack

4.7. Border Gateway Protocol hijacking attack

Border Gateway Protocol (BGP) hijacking [56] involves manipulating network traffic by making false routing announcements, effectively redirecting or isolating traffic. In blockchain networks, attackers can exploit BGP hijacking to partition the network or delay block propagation, creating opportunities for double-spending attacks. For instance, fewer than 100 IP prefix hijacks have historically isolated up to 50% of Bitcoin's hash rate. This means that attackers redirect network traffic from a significant portion of Bitcoin miners, disrupting their communication with the rest of the network. This isolation can delay the propagation of blocks, giving attackers a window to execute fraudulent transactions. Moreover, BGP exploits are relatively frequent, with approximately 100 incidents reported monthly. This frequency underscores the potential vulnerability of blockchain networks to these types of attacks if adequate countermeasures are not in place.

Attackers exploit block delays for 0 - or N-confirmation double-spends, triggered forks, initiated block races, or reduced miner revenue. Countermeasures have included multi-homing, using VPNs, random peer connections, and monitoring round-trip time (RTT) for anomalies. Anomaly detection and multiple gateways have improved resilience.

In Ethereum, more block confirmations have reduced BGP hijacking risks, and merchants can query peers to verify transactions. For Bitcoin, encrypting communications, using random TCP ports, and periodic UDP messages have helped detect synchronization issues and mitigate hijacking risks.

Feasibility : BGP hijacking poses a significant threat to blockchain networks by enabling attackers to manipulate Internet routing and isolate critical nodes. Apostolaki et al. [57] demonstrated that hijacking just 85 IP prefixes could isolate 50% of Bitcoin's mining power, while targeting only 13 ASes could disrupt 30% of Bitcoin nodes, emphasizing the centralization of blockchain infrastructure. Efforts to mitigate these risks include the adoption of encryption protocols (e.g., TLS and QUIC) and randomized peer-to-peer communication [58]. These are particularly vulnerable in regions with weak routing security or limited deployment of routing protection measures such as Resource Public Key Infrastructure (RPKI) [59].

Practical: Single blockchain networks rely heavily on public routing protocols such as BGP for inter-node communication, making them susceptible to attacks like traffic redirection, node isolation, or network partitioning [60]. These disruptions impact block propagation and consensus, enabling potential exploits such as double-spending or selfish mining [34]. This vulnerability stems from their inherent dependency on the underlying Internet infrastructure.By contrast, modern cross-chain protocols are designed to reduce reliance on public routing through the use of cryptographic proofs, consensus-based validation, and decentralized peer-to-peer communication. Instead of direct routing between nodes, cross-chain bridges employ validators or relayers to exchange verifiable proofs of state [61]. This architectural decoupling ensures resilience; even if portions of the public network are compromised, cross-chain systems maintain their security and operational integrity. The independence from traditional routing infrastructure is therefore a fundamental factor in their superior resistance to BGP hijacking attacks.

Real-World Scenario: KLAYswap BGP Hijack (February 3, 2022) [62]. **Similarity :** The KLAYswap BGP hijack involved attackers manipulating internet routing (BGP) to redirect ISP traffic to phishing websites. This attack targeted KLAYswap, a decentralized application operating on a single blockchain (Klaytn), demonstrating how routing-level attacks like BGP hijacks can facilitate double-spending or theft even in single-chain environments. This allowed them to steal nearly $1.9M by deceiving users into interacting with malicious smart contracts. Although this attack did not involve forging or reintroducing validator keys, it shares similarities with PoS long-range attacks by isolating users and leveraging rerouted traffic to propagate fraudulent transactions. Both exploits rely on disrupting trust and validation mechanisms, enabling unauthorized spending or state manipulation.

4.8. 0-*confirmation race attack*

The 0−confirmation race attack [63] exploits fast payments where merchants accept payments without confirmation. The attacker sends two conflicting transactions - one ($AT0$) to the merchant and another ($AT1$), with a higher fee, to itself. Since miners prioritize higher-fee transactions, $AT1$ is mined, invalidating $AT0$.

Unlike eclipse or BGP-based 0-confirmation attacks, the 0-confirmation race attack exploits the time gap and fee differences between the two transactions. To counter this, merchants have been advised to wait for at least one block confirmation before releasing goods, although this could expose them to Finney or Vector76 attacks.

For a single blockchain, solutions have included calculating the probability of double-spending attacks based on validation time, improving network policies, and setting block confirmation thresholds relative to transaction value. Detection techniques have involved a listening period to verify transaction legitimacy, inserting observers to relay and detect double spends, and alerting vendors when the network has detected a double spend. Enhanced observers (ENHOBS) [64] have combined listening periods with observers, while the E-cash protocol has used a trusted group of observers to detect double-spending in real time.

In a 0-confirmation race attack, the attacker takes advantage of the time gap before a transaction is fully verified. This allows them to initiate a transfer on one chain while simultaneously spending the same asset on another chain, leading to financial losses across both networks. In Figure 7, we have illustrated this cross-chain 0-confirmation race attack involving double-spending. The steps for such an attack have been listed below.

1. Transaction initialization: The attacker initiates a transaction ($AT0$) on a blockchain (chain A).
2. Transaction processing: Validators on chain A begin processing $AT0$.
3. Simultaneous high-fee transaction:
   (a) The attacker initiates a conflicting transaction $AT1$ on another blockchain (chain B) with higher transaction fees.
   (b) Simultaneously, the attacker relays the transaction $AT1$.
4. Conflicting transaction confirmation: The conflicting transaction $AT1$ is confirmed by the validators of chain A.
5. Funds retrieval: The attacker gains access to the funds from chain B via $AT0$, exploiting the transaction inconsistency, double-spending.

Feasibility: The feasibility of 0-confirmation race attacks can be quantified by analyzing factors such as transaction propagation delays, miner fee prioritization, and network latency. For example, if the propagation delay of a transaction T_0 to miners exceeds the time required for a higher-fee conflicting transaction T_1 to reach them, the success probability of the attack increases significantly. Empirical models demonstrate that in networks with latency under

100 ms and a fee premium of at least 10%, the chance of T_1 being mined first can exceed 60%, whereas implementing a mandatory confirmation period (e.g., waiting one full block) can reduce that probability below 5% (quantitative framework adapted from network delay–throughput studies)[65].

Practical Challenges: Cross-chain 0-confirmation race attacks are significantly less feasible in well-designed systems due to the inherent differences in transaction finality and validation mechanisms across blockchains. Many cross-chain protocols enforce strict safeguards such as mandatory block confirmations, cryptographic proofs (e.g., Merkle proofs), and conflict detection mechanisms before processing asset transfers. For example, protocols like Bitcoin and Wrapped Bitcoin (WBTC) [66] require multiple confirmations to ensure transaction immutability, reducing the likelihood of successful double-spending. Additionally, platforms like Polkadot utilize Cross-Chain Message Passing (XCMP) [67], which enables real-time conflict resolution and transaction validation across chains. These measures ensure that unconfirmed transactions are not prematurely acted upon, effectively mitigating the vulnerabilities that 0-confirmation race attacks exploit. As a result, attackers face increased complexity and significantly reduced success probabilities in cross-chain environments compared to single-chain scenarios.

Real-Life Scenario: Bitcoin Payment Double-Spends (BitPay Incident) (2019) [68]. **Similarity:** In 2019, attackers exploited vulnerabilities in 0-confirmation transactions within the Bitcoin network to execute double-spends against merchants using the BitPay payment gateway. The issue arose because certain merchants and payment systems trusted transactions without waiting for even a single block confirmation. These systems relied solely on the initial broadcast of the transaction to the network, making them susceptible to manipulation and fraud.

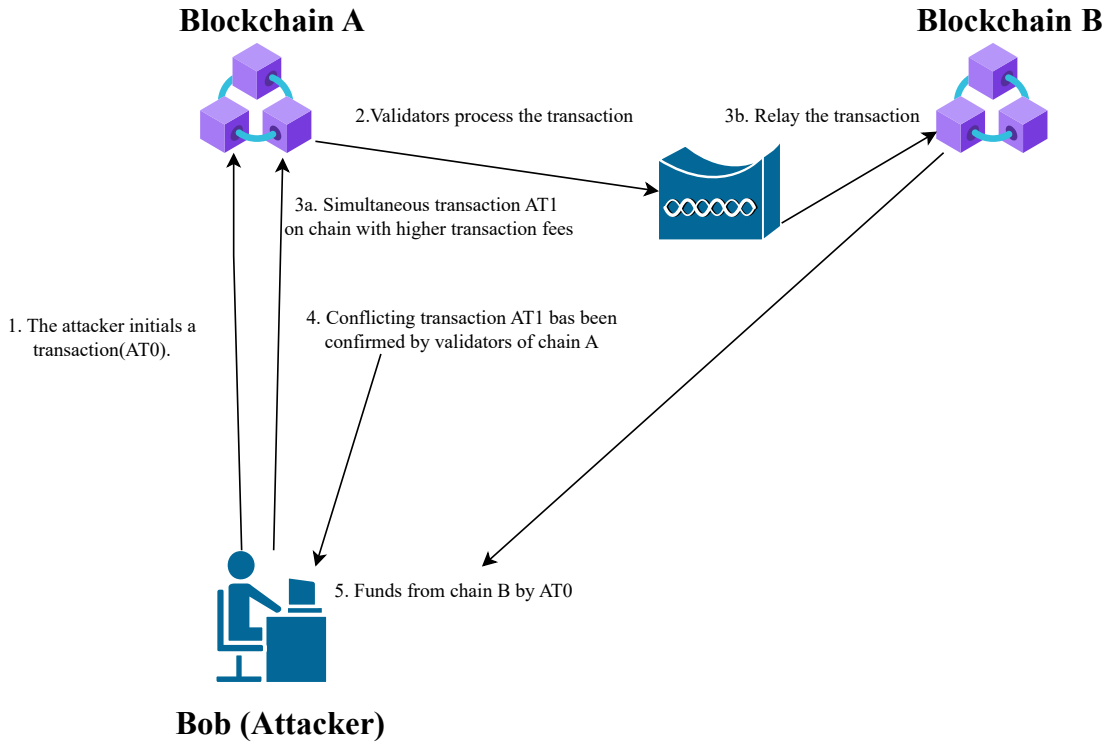

Figure 7. 0-Confirmation race attack based double-spending

4.9. *Vector*76 *attack*

*Vector*76 is a 1-confirmation attack that combines 0-confirmation race and finney attacks together, targeting exchanges and e-wallets (digital wallets used to store and manage cryptocurrencies) with static IP addresses . The

attacker operates two nodes: node A, connected to the exchange, and node B, connected to the broader blockchain. The attacker then creates two conflicting transactions — $AT0$ on node A and $AT1$ on node B, and mines a block with $AT0$ without broadcasting it. Once the exchange has accepted $AT0$ with one confirmation, the attacker immediately withdraws the funds, while $AT1$ eventually becomes valid, resulting in a successful double-spend.

To protect a single blockchain against *Vector*76-based attack, nodes should avoid 1-confirmation transactions and implement countermeasures like listening periods, observer nodes, and monitoring outbound connections for potential double-spending attempts.

In a cross-chain context, the *Vector*76 attack is a double-spending attack that exploits discrepancies in network propagation to target vulnerable nodes across two interconnected blockchains. The attacker manipulates transactions on both chains to maximize the effectiveness of the attack, causing double-spending by exploiting the delays or inconsistencies in transaction propagation between the chains. A cross-chain *Vector*76 attack involving double-spending has been illustrated in Figure 8. Below, we have depicted a step-by-step breakdown of the process.

1. Initiate transaction ($AT0$): The attacker, Bob, initiates a transaction ($AT0$) on blockchain A, which is designed to use funds from blockchain A.
2. Broadcast transaction to weak nodes: The transaction ($AT0$) is broadcast across the network, specifically targeting weak nodes on blockchain A. These nodes are less secure or isolated, making them more vulnerable to manipulation.
3. Initiate conflicting transaction ($AT1$): Bob initiates another transaction ($AT1$) on blockchain B. This transaction conflicts with $AT0$, as it attempts to use the same funds on blockchain B.
4. Confirm transaction on blockchain B ($AT1$): The weak or compromised nodes confirm the fraudulent transaction ($AT1$). This action effectively "spends" the funds on blockchain B.
5. Recover funds on blockchain A: When the attacker controls or manipulates weak nodes on blockchain A, Bob(the attacker) can reverse the $AT0$ transaction, which results in the funds being "returned" to the attacker.
6. Weak nodes confirm $AT0$: While the weak nodes confirm $AT0$ on blockchain A, Bob retains control over the funds because the nodes are manipulated to accept an invalid transaction.
7. Reorganize blockchain A: The longer chain on blockchain A is reorganized to include $AT1$. This final step ensures that Blockchain A reflects the attacker's version of the blockchain, completing the double-spend scenario

Feasibility: Vector76 attacks leverage propagation races and assumptions about transaction confirmation standards to exploit timing vulnerabilities in blockchain systems. These attacks are particularly effective in networks with moderate latency, where simulations demonstrate success probabilities of 20–30% when transactions are accepted after only one confirmation. However, increasing the confirmation threshold to six or more significantly mitigates the risk, reducing the success probability to negligible levels below 1% [69].

Practical Challenges: The Vector76 attack, which exploits network propagation delays to achieve double-spending on a single blockchain, is largely impractical in cross-chain environments. Cross-chain systems rely on mechanisms such as atomic swaps, bridging contracts, or multi-signature schemes, which introduce stringent synchronization and validation protocols. These systems require mutual confirmation across chains, significantly increasing the time and computational resources needed to execute a successful attack. Additionally, latency differences between independent chains, combined with enhanced monitoring and finality guarantees in modern cross-chain designs, mitigate the feasibility of orchestrating simultaneous double-spends [70]. Consequently, the inherent decentralization and varied architectures of cross-chain networks render them resilient against propagation delay-based attacks like Vector76.

Real-Life Scenario: Original Attack Name: Pike Finance Exploit (April 2024) [71]. **Similarity :** The Pike Finance exploit mirrors the Vector76 attack by exploiting temporary inconsistencies to double-spend. In Vector76, an attacker withholds a pre-mined block with a transaction, broadcasts a conflicting transaction, and later releases the block to reverse the payment. Similarly, in the Pike Finance exploit, asynchronous state updates across Ethereum, Optimism, and Arbitrum allowed the attacker to withdraw funds before states synchronized, enabling cross-chain double-spending. Both attacks manipulate system delays to achieve duplicated value transfers.

4.10. Discussion on Double Spending Attacks

Double-spending attacks have garnered significant attention on single blockchains because they directly undermine the core principle of blockchain technology: trustless, verifiable transactions. Recently, these attacks have

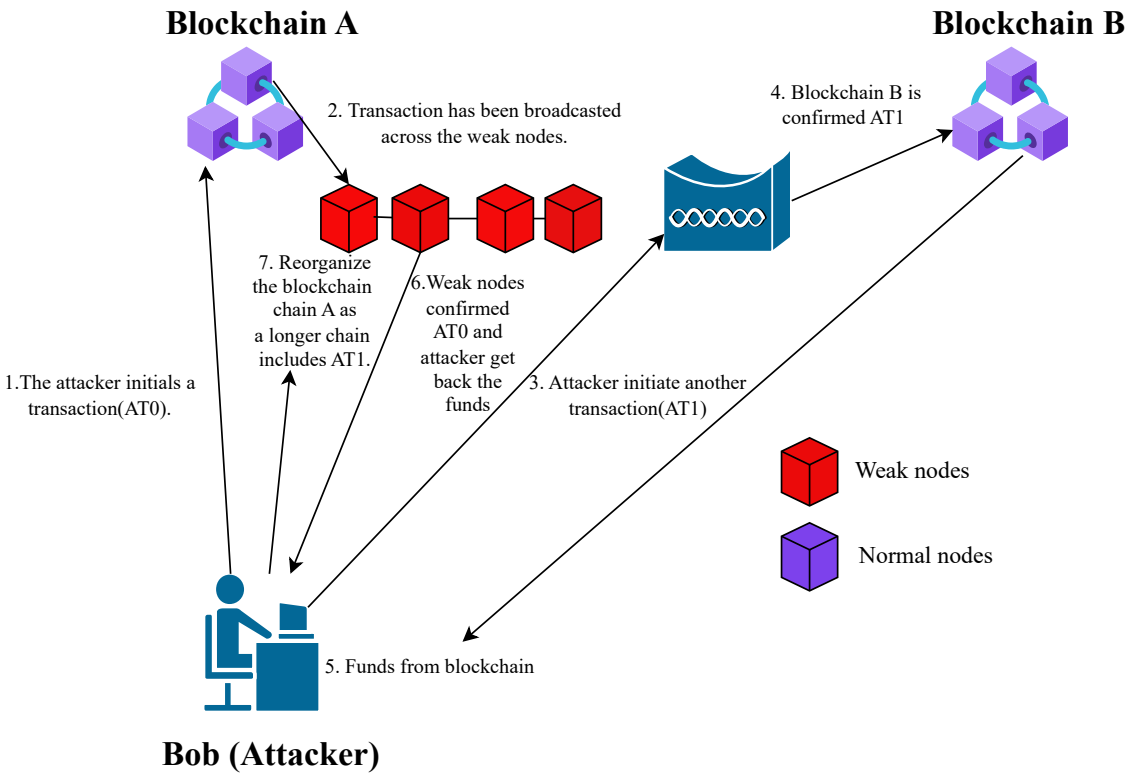

Figure 8. Vector76 double-spending attack

become more complex as they are combined with other types of attacks, making them even more vulnerable. This study analyzed nine types of double-spending attack scenarios across single-blockchain and cross-chain environments. While many of these attacks are theoretically feasible, practical constraints often limit their success. For example, 51% attack, requiring majority control of network power, remains largely impractical in well-established networks due to the immense economic resources needed. Similarly, Sybil attacks fail without sufficient computing power to manipulate consensus. Finney-based attacks and 0-confirmation race attacks exploit specific vulnerabilities, such as merchants accepting unconfirmed transactions or weak protocol safeguards, making them rare in practice. Time advantage attacks, like selfish mining and block withholding, are more common in single-chain systems due to processing inefficiencies, whereas permissioned networks and robust consensus mechanisms fortify cross-chain frameworks. PoS long-range attacks and Vector76 attacks are also significantly mitigated in cross-chain systems through enhanced finality mechanisms, checkpointing, and multi-layered verification processes. Meanwhile, attacks like BGP hijacking, which exploit routing vulnerabilities, have minimal impact on cross-chain networks due to their reliance on cryptographic proofs and decentralized communication. Through this feasibility study, we identified three types of double-spending attacks that are feasible in a cross-chain context: (i) *traditional double-spending attacks*, (ii) *eclipse-based double-spending attacks*, and (iii) *double-spending attacks involving observers*. Observers are integral components of a blockchain system designed to monitor transactions and validate their integrity. Serving as an additional validation layer, they detect and flag potentially malicious or invalid activities, such as double-spending attempts. By maintaining a comprehensive record of invalid transactions, observers enhance the security of the blockchain network and play a crucial role in preventing fraudulent activities.

First, we have theoretically compared the risks and mechanics between eclipse-based and traditional double-spending attacks in a cross-chain context, as shown in Table 2. This analysis has highlighted each attack type's unique

characteristics, vulnerabilities, and implications, emphasizing how cross-chain interactions impact their feasibility and potential consequences. Next, we have explored the impact of observers in cross-chain double-spending attacks, presenting a theoretical comparison between traditional double-spending scenarios and those involving observers, as depicted in Table 3. Our findings indicate that the presence of an observer significantly reduces the likelihood of successful attacks by providing comprehensive visibility across multiple chains and enabling rapid detection and response to suspicious activities. On the other hand, the absence of observers increases the success rates of double-spending attacks, as attackers can exploit the lack of oversight, undermining trust and financial stability within the cross-chain ecosystem. Therefore, the implementation of observer mechanisms is crucial for maintaining the integrity and security of cross-chain transactions, as discussed in Section 4.

Table 2. Comparison between traditional and eclipse-based attacks

Aspect	Traditional Double-Spending Attacks	Eclipse-Based Double-Spending Attacks
Definition	Attempts to spend the same cryptocurrency twice by broadcasting conflicting transactions to the network.	Targets specific nodes by isolating them from the network, controlling the information they receive.
Execution	Relies on propagating a conflicting transaction quickly while the original transaction is being confirmed.	Involves controlling or compromising nodes to propagate conflicting transactions only to those isolated nodes.
Attack surface	Expands in cross-chain scenarios due to different confirmation times and validation mechanisms.	Effective in cross-chain contexts by targeting nodes involved in cross-chain bridges or payment channels.
Propagation dynamics	Depends on transaction propagation times across different chains, which can vary significantly.	Uses targeted isolation tactics, allowing for manipulation of transaction flow beyond traditional propagation constraints.
Confirmation strategies	Relies on the number of confirmations, which can differ between chains, complicating transaction validity.	Exploits confirmation strategies of both chains, enabling scenarios where one transaction is seen as valid while the other is not.
Risks	Vulnerable to delayed confirmations; success relies on fast propagation of conflicting transactions.	More complex to execute, requiring control over multiple nodes; risk of detection is higher due to coordinated manipulation.
Complexity of execution	Generally simpler, relying on rapid transaction submission and network vulnerabilities.	More complex, involving node isolation and coordination to be effective.
Detection risk	Lower risk of detection if propagation is fast and unnoticed.	Higher risk of detection due to the need for coordinated control over multiple nodes and potential discrepancies in confirmations.

5. Experimental Assessment

In this section, we experimentally evaluate the attacks, simulating a realistic cross-chain ecosystem. In particular, we implemented the three main double-spending attack scenarios in a cross-chain environment: traditional double-spending without an observer, double-spending with an observer, and eclipse-based double-spending.

For simplicity, we built a case study of *homogeneous* interoperability between two blockchains. This means a cross-chain ecosystem with different blockchain instances based on the same blockchain protocol, in our case,

Table 3. Comparison between traditional and observer-based attacks

Aspect	Double-spending attack	Double-spending attack with observer
Definition	An attempt to spend the same digital asset more than once within a blockchain network.	An attempt to spend the same digital asset across multiple blockchains, monitored by an observer.
Visibility	Limited visibility into transactions, making it difficult to detect conflicting transactions in real-time.	Comprehensive visibility across chains, enabling the identification of conflicting transactions as they occur.
Detection	Detection is often delayed, relying on the consensus mechanisms of individual chains, which may not communicate effectively.	Real-time monitoring by the observer allows for immediately detecting anomalies and suspicious activities.
Response time	Slow response time due to the need for chain consensus, which can lead to successful attacks before they are noticed.	Rapid response enabled by the observer's alerts allows quick intervention to halt suspicious transactions.
Attack success rate	Higher likelihood of success, especially if the attacker exploits timing discrepancies between chains.	Lower likelihood of success due to the observer's proactive monitoring and intervention capabilities.
User trust	Increased risk of eroding user trust due to potential losses and unresolved conflicts following a successful attack.	Enhanced user trust through effective monitoring and quick resolution of discrepancies, preserving the integrity of the cross-chain ecosystem.
Impact on financial stability	Significant negative impact on financial stability, as successful attacks can lead to asset loss and confusion.	Low impact on financial stability, as observers can facilitate recovery and minimise losses in case of an attack.
Dispute resolution	Complicated and lengthy dispute resolution processes, as affected parties must navigate through multiple chains without oversight.	Streamlined dispute resolution with a central observer coordinating responses across chains, aiding in resolving conflicts efficiently.

Ethereum. Hence, we evaluated and measured crucial performance metrics, including confirmation rate, attacker's income, and miner's income. The simulations allow us to model different scenarios of cross-chain asset transfers and potential attack vectors. Note that homogeneous interoperability is standard to find in the real world, as the majority of the total value locked (TVL) of all blockchains is kept by Ethereum-based blockchains (e.g., Ethereum (main-net), Tron, Arbitrum, etc.) [72].

All experiments have been performed on an Ubuntu 24.04 LTS (64-bit) operating system with 16 GB RAM and an Intel Core i7, 2.81 GHz processor. In particular, we used Ganache [73] and other tools of the Truffle Suite [74] to build, simulate, and debug the Ethereum-based blockchains, smart contracts, user accounts, miners, and tokens. Node.js and Web3.js [75] have been involved to create a bridge between the two independent blockchain networks and to simulate the cross-chain interactions and the double-spending attack scenarios where an attacker tries to spend the same assets on both blockchains. Figure 9 provides a high-level overview of the experimental setup.

5.1. Cryptographic foundations and transaction

The core cryptographic algorithm used to secure transactions in this environment is the Elliptic Curve Digital Signature Algorithm (ECDSA). ECDSA has been employed to generate digital signatures, ensuring that transactions

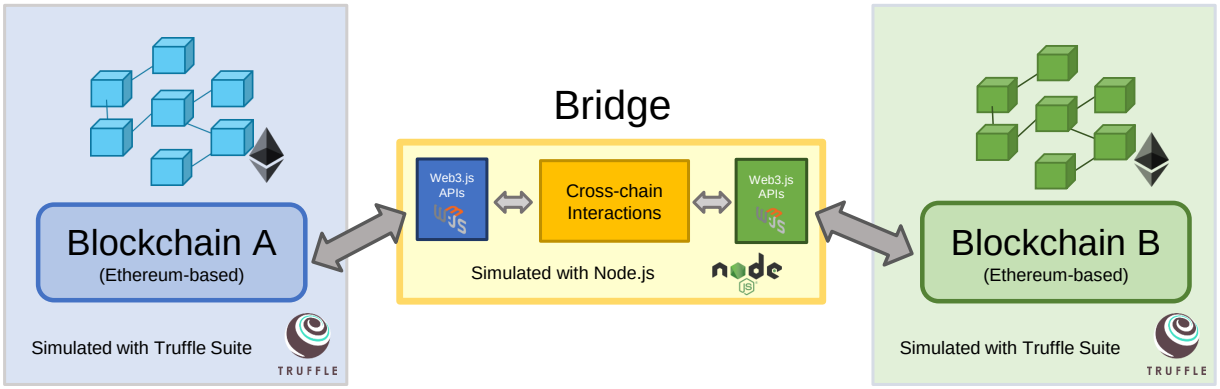

Figure 9. High-level overview of the experimental environment architecture.

between blockchain users are secure and tamper-proof. ECDSA has played a critical role in preventing unauthorized modification of transactions after they have been broadcast to the network. In this setup, user accounts are part of one of the two blockchains. Each account can perform transactions using digital signatures based on ECDSA, ensuring cryptographic integrity between the two blockchains. Users initiate and sign transactions using ECDSA. Once signed, the transactions are broadcast to the network. Miners belonging to both blockchains then verify the authenticity of the transactions and add them to blocks. Each blockchain operates independently with its miners. However, cross-chain transactions must be confirmed on both blockchains to be validated. The mining rewards have been calculated based on honest mining efforts.

5.2. Metrics for evaluation

To evaluate the effectiveness of the scheme, we have focused on the following metrics:

1. Confirmation rate: The confirmation rate for each blockchain transaction has been defined based on its success or failure across the two chains. If a transaction is successful (valid and included in the block) on both blockchains, the confirmation rate is considered 100%. Conversely, if the transaction is unsuccessful (invalid or not included) on both blockchains, the confirmation rate is regarded as 0%. If the transaction is successful on one blockchain but fails on the other, the confirmation rate is considered 50%.
2. Attacker's income: The attacker's income has been estimated based on the success of the transactions on both blockchains.
3. Miner's Income: The miners' income has been estimated based on the success of transactions, and includes both transaction fees and base rewards (for each blockchain).

$$MinerIncome = TransactionFees + MiningReward \tag{1}$$

5.3. Attack Strategies and Assumptions

We implemented and evaluated three double-spending strategies to reflect varying levels of security in cross-chain environments, particularly among Ethereum-based interoperable chains:

- **Traditional Double-Spending (No Observer):** This strategy assumes the absence of cross-chain transaction validation. Transactions are broadcast independently across multiple chains, with no synchronization or consistency checks. As a result, detection of double-spending is delayed, often until multiple confirmations have been reached.

- **Double-Spending with Observer:** This model introduces a monitoring entity (an observer) that continuously validates transaction consistency across chains. By detecting conflicts in real time, the observer significantly reduces the attack success window and helps maintain inter-chain transactional integrity.

- **Eclipse-Based Double-Spending:** In this scenario, the attacker isolates a subset of nodes on one of the chains, limiting their view of the network. This partial eclipse allows the attacker to inject fraudulent transactions without immediate detection, exploiting the disrupted consensus process.

The objective of this evaluation is to compare the *practical feasibility* of these strategies. Metrics such as *confirmation delay*, *attacker profit*, and *honest mining income* serve as comparative benchmarks. It should be emphasized that establishing optimality within a formal adversarial framework is orthogonal to the purpose of this evaluation.

6. Results and Discussion

We simulate the three different types of attack scenarios in our experimental cross-chain environment. In all the simulations, the attacker initiates two identical transactions on two blockchains, transferring funds to victims. In the first scenario, i.e., traditional double-spending without an observer, no mechanism exists to validate the transactions. However, in the second scenario, i.e., double-spending with an observer, the observer verifies the attacker's signed transactions. If the transactions are deemed valid, they are added to the blockchain and mined into blocks. The observer's role mirrors that of blockchain validators in real-world scenarios, where they ensure the legitimacy of transactions, thereby helping to prevent fraud. The final scenario is based on an eclipse attack, in which the attacker isolates a set of blockchain nodes. This allows the attacker to execute a double-spending attack more effectively, as the isolated nodes cannot communicate with the rest of the network and verify the transaction statuses.

The experimental results demonstrate that each double-spending attack strategy varies significantly in feasibility and impact. In the traditional double-spending scenario without an observer, attacks succeed moderately often, with confirmation rates hovering around 50%, reflecting the vulnerability of systems lacking cross-chain monitoring. When an observer is introduced, the success of double-spending attacks drops sharply; most fraudulent transactions are identified before confirmation, resulting in minimal attacker income and lower confirmation rates. This confirms the observer's critical role in maintaining consistency and trust in cross-chain systems. In contrast, eclipse-based attacks are the most effective, with attackers achieving near 100% confirmation rates and maximum income. This is due to the attacker's ability to isolate nodes and manipulate consensus within a partitioned segment of the network. However, such attacks are complex and require greater control over the network topology. Overall, the experiments validate that observer-based monitoring significantly enhances security, while eclipse-based attacks represent a powerful but harder-to-execute threat in cross-chain environments.

To assess the robustness of each attack, we executed every attack scenario many times. The results in Figure 10 show that the presence of an observer effectively prevents the attacker from succeeding. The attacker's income remains minimal in this setup, as most double-spending attempts are thwarted. In the traditional scenario without an observer, the attacker's average income is higher than that in the case of an attack with the observer, but lower than that in the case of an eclipse attack. Eclipse-based double-spending yields the highest income for the attacker, as the isolation of nodes enables the successful execution of fraudulent transactions on both blockchains.

Figure 11 represents the confirmation rates in all three attack types. In most cases, there has been a 100% confirmation rate for both transactions on both the blockchains for eclipse-based attacks, implying that the attacks have been successful. The confirmation rate follows the same trend as the attacker's income. It is the lowest in the case of traditional double-spending without an observer and the highest in the case of eclipse attacks. In the observer scenario, most attacks have been unsuccessful, resulting in high 0% confirmation. The partial success rate (50% confirmation) has been the highest in the traditional setup without an observer but significantly lower in the observer and eclipse-based scenarios.

Figure 12 illustrates the miners' income on both blockchains. Miners receive the transaction fee and a base reward if a successful transaction occurs. In the observer-enabled scenario, miners' income has generally been lower, as most attacks have been detected and prevented, leading to fewer valid transactions. In contrast, miners' income has been higher in eclipse-based attacks due to successful double-spending events, while traditional double-spending attacks without an observer have produced intermediate results.

Our experimental results have shown that eclipse-based double-spending attacks have a higher success rate than traditional double-spending attacks, both with and without an observer. Eclipse attacks have exploited node isolation, making it easier for attackers to spend twice on multiple blockchains. However, these attacks have been resource-intensive and complex to execute. Furthermore, it has also been observed that the presence of an observer has signif-

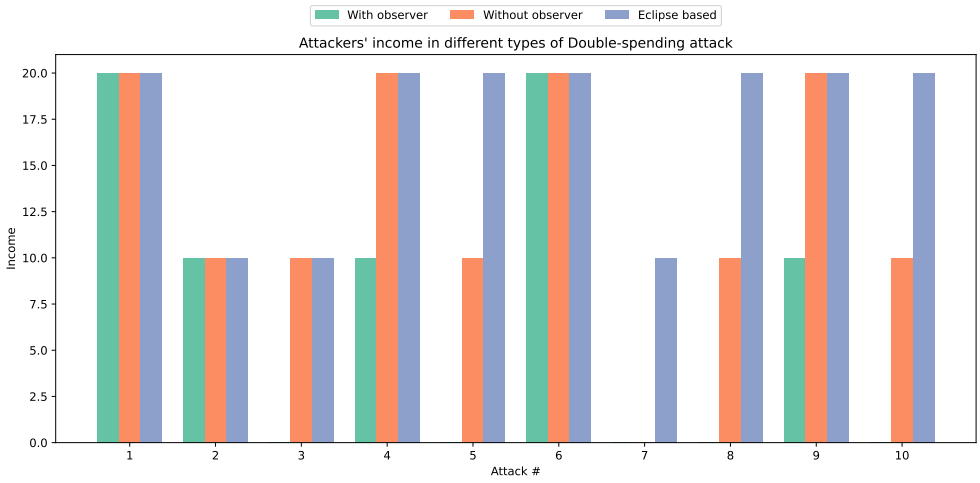

Figure 10. Revenue Generated by Attackers in Double-Spending Scenarios

icantly mitigated double-spending risks. The observer has detected discrepancies and conflicting transactions in real time by monitoring and validating transactions across blockchains. This has allowed the network to reject fraudulent transactions before they are confirmed, thus maintaining the integrity and reliability of the system. The observer has played a crucial role in maintaining trust across interconnected blockchains by preventing double-spending attacks and ensuring consistency in transaction validation.

Case Study: For simplicity, we have limited our experiments to two homogeneous blockchains; however, the experimental setup can include more than two blockchains. This case study illustrates a double-spending attack on a cross-chain network comprising three interconnected blockchains: Chain A, B, and C. Each blockchain operates with its consensus mechanism. The network enables interoperability through a cross-chain bridge, facilitating token transfers between chains. Attackers have taken advantage of synchronization vulnerabilities between the blockchains to execute a double-spending. Here are the steps of the attack, both described in text and structured for representation as a diagram in Figure 13:

1. Legitimate Transaction Across Chains: The attacker has initiated a valid token transfer from Chain A to Chain B via the cross-chain bridge by locking the tokens on Chain A and issuing a corresponding representation on Chain B.
2. Private Fork on Chain A: The attacker has privately mined an alternative version of Chain A, excluding the original transaction. This alternative chain is designed to overwrite the legitimate transaction once it is published.
3. Exploiting Cross-Chain Synchronisation: While Chain A and Chain B have processed the legitimate transaction, the attacker has simultaneously initiated a transfer from Chain B to Chain C, creating a chain of representations across the network.
4. Broadcast of Alternative Chain: The attacker has released the fork of Chain A, invalidating the initial transaction. As a result, the tokens locked on Chain A have been effectively ”unlocked,” allowing the attacker to reclaim them.
5. Finalisation on Chain C: Using the representations from the invalidated transaction, the attacker has finalized a legitimate transfer on Chain C, effectively double-spending the same tokens across three chains.

7. Conclusion and Future Work

In this work, we initially explored the various types of double-spending attacks in single blockchains. We have examined the feasibility of these attacks in a cross-chain ecosystem. Our analysis shows that only three types- traditional, observer-based, and eclipse-based double-spending attacks are feasible in a cross-chain environment. A

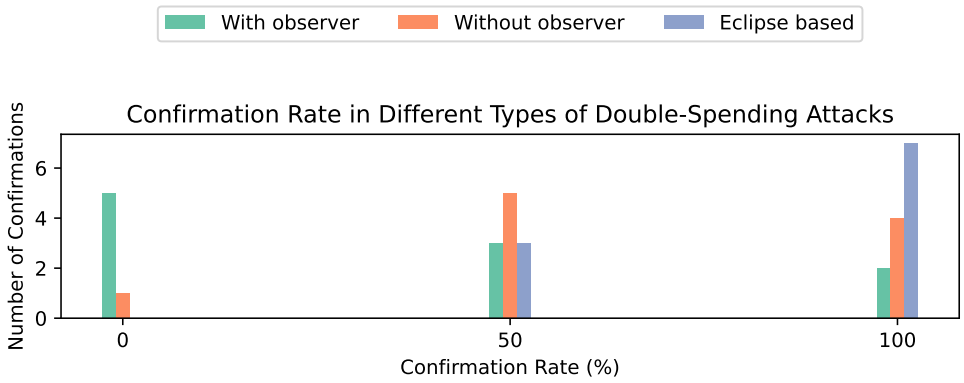

Figure 11. Variation in Confirmation Rates for Double-Spending Methods

comparative analysis, based on parameters such as confirmation rate, attacker's income, and miner's income across both chains, has revealed that the eclipse-based attack is the most threatening since it has the highest confirmation rate. Besides, the traditional double-spending attack proves to be less effective than eclipse-based attacks, while the presence of an observer further reduces the attacker's success rate. This highlights the need for improved cross-chain security. Current systems are either trusted, where reliance on intermediaries like bridges or relayers introduces vulnerabilities; or trustless, which faces security, complexity, and implementation challenges. In the future, the security analysis of privacy-enhancing technologies such as zero-knowledge proofs and threshold signatures can be added to strengthen cross-chain security, providing more robust methods for protecting user privacy and reducing attack surfaces.This will provide stronger mechanisms for safeguarding user privacy and minimizing potential attack vectors. Zero-Knowledge Proofs (ZKPs) hold the potential to revolutionize blockchain technology by improving scalability, enhancing privacy, and offering resistance to quantum attacks. Innovations such as aggregation schemes for more efficient proof generation and verification [76] can reduce computational costs, enabling blockchains to handle larger datasets effectively—benefiting industries like finance and supply chains. Additionally, approaches like Designated Verifier Proofs (DVP) and Blockchain Designated Verifier Proofs (BDVP) further protect privacy by limiting proof verification to authorized parties and safeguarding confidentiality against quantum threats [77]. Together with ongoing advancements in security frameworks, these developments lay the foundation for the deployment of ZKPs in privacy-focused decentralized applications and secure cross-chain communication protocols [78]. Although this work primarily focuses on consensus and network layer attacks adapted to cross-chain scenarios, there is also the need for documenting application-specific cross-chain attacks, such as bridge logic flaws, validator collusion in federated bridges, and oracle-based manipulation. These attacks will be addressed in future work to provide a comprehensive security framework for cross-chain ecosystems.

At present, our study does have a few limitations. In this work, we based our evaluation on a homogeneous blockchain setup to assess the feasibility of double-spending attacks. However, exploring these scenarios on a heterogeneous blockchain could provide valuable insights into the impact of such attacks across varied blockchain ecosystems. Moreover, we have centred our analysis on nine distinct types of double-spending attacks that are particularly relevant in cross-chain environments. Other attacks, such as selfish mining in multi-chain settings, time-dilation attacks, and advanced Sybil attacks, are not included here due to their theoretical nature, high simulation requirements, and dependency on specific cross-chain protocols. This study could be further extended to include other kinds of attacks. Further, we have conducted the experiments using Ganache, a local Ethereum blockchain simulator from the Truffle suite. This has been widely used to develop and test Ethereum applications. However, this setup may not fully

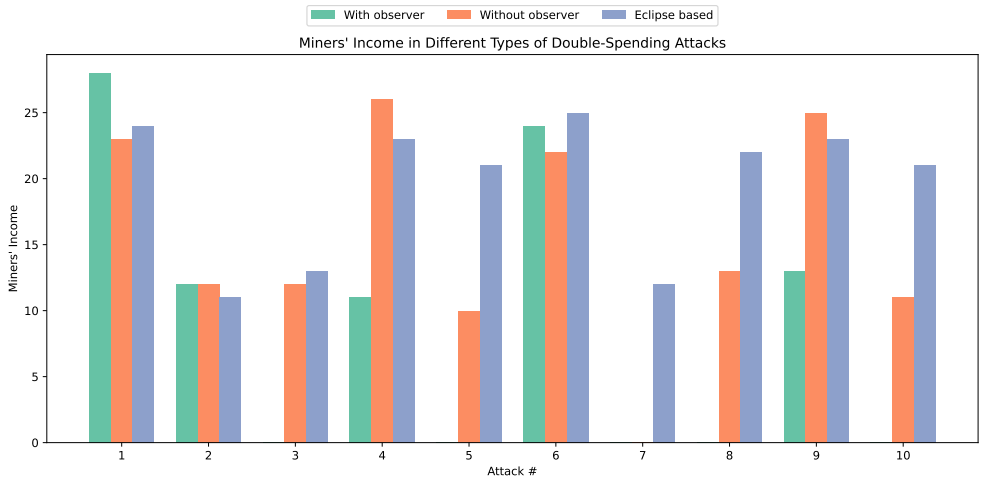

Figure 12. Income Analysis of Miners in Various Attack Types

represent all possible real-world blockchain systems. Our work could be supplemented by using Infura or another setup offering API access to the Ethereum blockchain and IPFS (InterPlanetary File System).

In addition, the issues of technical mechanisms can be also considered, including atomic swap and HTLCs [79] implementations. Atomic swap is a protocol that allow two parties to exchange assets across independent blockchains without a trusted intermediary by using cryptographic assurances that either both sides of the transaction complete or neither does. The swap mechanism is implemented using HTLCs on the chains involved in the swap. HTLCs use a combination of hashlocks and timelocks to enforce conditional payments across chains, requiring the recipient to reveal a cryptographic secret within a specific timeframe to claim the funds. While these techniques are fundamental to decentralized cross-chain interactions, they also introduce distinct security risks. HTLCs, for instance, are vulnerable to timing discrepancies, preimage reuse, and refund race conditions, especially when chains operate under different confirmation speeds. On the other side, atomic swaps can be exposed to griefing, front-running, or incomplete execution in volatile or adversarial environments. These implementations can expand the cross-chain attack surface, creating new vectors for exploits.

Code availability

The implementation of proposed mechanism is available at:
`https://github.com/aradhita1988/DoubleSpendingAttack-CrossChain`.

Acknowledgments

This work has been partially supported by the SERICS project (PE00000014 - CUP H73C2200089001) funded by PNRR NextGeneration EU.

References

[1] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman *et al.*, "Blockchain technology: Beyond bitcoin," *Applied innovation*, vol. 2, no. 6-10, p. 71, 2016.
[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Satoshi Nakamoto*, 2008.
[3] B. Kewell, R. Adams, and G. Parry, "Blockchain for good?" *Strategic change*, vol. 26, no. 5, pp. 429–437, 2017.
[4] W. A. Kaal, "Digital asset market evolution," *J. Corp. L.*, vol. 46, p. 909, 2020.
[5] S. Schulte, M. Sigwart, P. Frauenthaler, and M. Borkowski, "Towards blockchain interoperability," in *Business Process Management: Blockchain and Central and Eastern Europe Forum*, C. Di Ciccio, R. Gabryelczyk, L. García-Bañuelos, T. Hernaus, R. Hull, M. Indihar Štemberger, A. Kő, and M. Staples, Eds. Cham: Springer International Publishing, 2019, pp. 3–10.

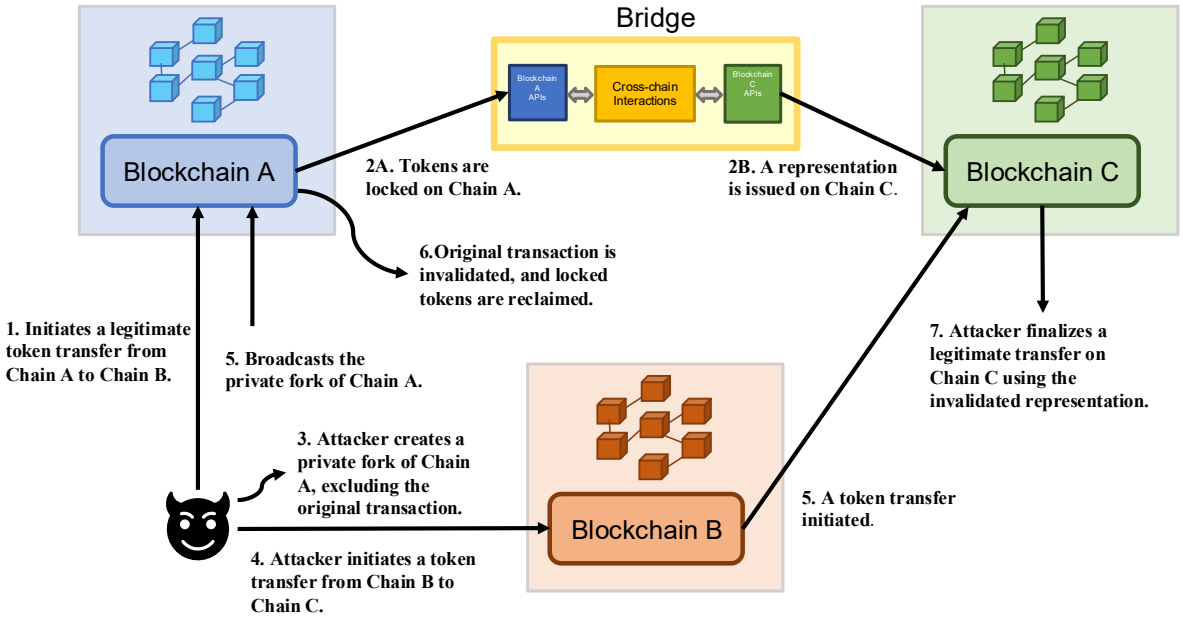

Figure 13. Triple-Blockchain Double-Spending attack in Cross-Chain Synchronization

[6] 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. [Online]. Available: https://doi.org/10.1145/3471140

[7] "Blockchain and Web3 Strategy," 2024, accessed 01/2025. [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/blockchain-strategy

[8] "Blockchain: The India Strategy - Part 1," 2020, accessed 01/2025. [Online]. Available: https://www.niti.gov.in/sites/default/files/2020-01/Blockchain_The_India_Strategy_Part_I.pdf

[9] "China unveils $54.5B National Blockchain Roadmap," 2025, accessed 01/2025. [Online]. Available: https://blockchaintechnology-news.com/news/china-unveils-54-5b-national-blockchain-roadmap/

[10] "Dubai Blockchain Strategy," accessed 01/2025. [Online]. Available: https://www.digitaldubai.ae/initiatives/blockchain

[11] L. Olivieri, L. Pasetto, L. Negrini, P. Ferrara *et al.*, "European union data act and blockchain technology: Challenges and new directions," in *CEUR WORKSHOP PROCEEDINGS*, vol. 3791. CEUR-WS, 2024, 6th Distributed Ledger Technologies Workshop (DLT2024). [Online]. Available: https://ceur-ws.org/Vol-3791/paper30.pdf

[12] L. Olivieri and L. Pasetto, "Towards compliance of smart contracts with the european union data act," in *CEUR Workshop Proceedings*, vol. 3629. CEUR-WS, 2024, 5th Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis (OVERLAY 2023). [Online]. Available: https://ceur-ws.org/Vol-3629/paper10.pdf

[13] T. Zilavy, "What is a hybrid blockchain and why you need to know about it," *URL: https://medium. com/altcoin-magazine/what-is-a-hybrid-blockchainand-why-you-need-to-know-about-it-c7b887d2bae. Accessed: 1st September*, 2019.

[14] Y. Jiang and J. Zhang, "Profitability analysis of time-restricted double-spending attack on pow-based large scale blockchains with the aid of multiple attacks," *IEEE Transactions on Information Forensics and Security*, 2024.

[15] K. Sai and D. Tipper, "Disincentivizing double spend attacks across interoperable blockchains," in *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 2019, pp. 36–45.

[16] K. Liao and J. Katz, "Incentivizing blockchain forks via whale transactions," in *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21*. Springer, 2017, pp. 264–279.

[17] B. Biais, C. Bisiere, M. Bouvard, and C. Casamatta, "The blockchain folk theorem," *The Review of Financial Studies*, vol. 32, no. 5, pp. 1662–1715, 2019.

[18] C. Pinzón and C. Rocha, "Double-spend attack models with time advantange for bitcoin," *Electronic Notes in Theoretical Computer Science*, vol. 329, pp. 79–103, 2016.

[19] A. M. Antonopoulos, *Mastering Bitcoin: Programming the Open Blockchain*, 2nd ed. Sebastopol, CA, USA: O'Reilly, 2017.

[20] M. Rosenfeld, "Analysis of hashrate-based double spending," *arXiv preprint arXiv:1402.2009*, 2014.

[21] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 3–16.

[22] J. Hinz, "Resilience analysis for double spending via sequential decision optimization," *Applied System Innovation*, vol. 3, no. 1, p. 7, 2020.

[23] M. Iqbal and R. Matulevičius, "Exploring sybil and double-spending risks in blockchain systems," *IEEE Access*, vol. 9, pp. 76 153–76 177, 2021.
[24] J. Zheng, H. Huang, Z. Zheng, and S. Guo, "Adaptive double-spending attacks on pow-based blockchains," *IEEE Transactions on Dependable and Secure Computing*, 2023.
[25] C. Natoli, P. Ekparinya, G. Jourjon, and V. Gramoli, "Blockchain double spending with low mining power and network delays," *Distributed Ledger Technologies: Research and Practice*, 2024.
[26] M. Zhang, X. Zhang, Y. Zhang, and Z. Lin, "Cross-chain bridges: Attack taxonomy, defenses, and open problems," 2024.
[27] L. Cheng, Z. Lv, O. Alfarraj, A. Tolba, X. Yu, and Y. Ren, "Secure cross-chain interaction solution in multi-blockchain environment," *Heliyon*, vol. 10, no. 7, 2024.
[28] L. Olivieri, A. Mukherjee, N. Chaki, and A. Cortesi, "Cross-chain Smart Contracts and dApps Verification by Static Analysis: Limits and Challenges," in *CEUR Workshop Proceedings*, vol. 3962. CEUR-WS, 2025, iTASEC & SERICS 2025 Joint National Conference on Cybersecurity 2025. [Online]. Available: https://ceur-ws.org/Vol-3962/paper16.pdf
[29] D. Mishra and S. Phansalkar, "Blockchain security in focus: A comprehensive investigation into threats, smart contract security, cross-chain bridges, vulnerabilities detection tools & techniques," *IEEE Access*, 2025.
[30] S. Sayeed and H. Marco-Gisbert, "Assessing blockchain consensus and security mechanisms against the 51% attack," *Applied sciences*, vol. 9, no. 9, p. 1788, 2019.
[31] P. Perazzo and R. Xefraj, "Smartfly: Fork-free super-light ethereum classic clients for the internet of things," *IEEE Internet of Things Journal*, 2024.
[32] D. G. Baur and L. Hoang, "The bitcoin gold correlation puzzle," *Journal of Behavioral and Experimental Finance*, vol. 32, p. 100561, 2021.
[33] L. Olivieri, A. Mukherjee, N. Chaki, and A. Cortesi, "Blockchain Interoperability through Bridges: A Token Transfer Perspective," in *2024 6th International Conference on Blockchain Computing and Applications (BCCA)*, 2024, pp. 742–748.
[34] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Communications of the ACM*, vol. 61, no. 7, pp. 95–102, 2018.
[35] N. Mutual, "Ethereum classic 51% attacks," 2020, accessed: 2025-07-01. [Online]. Available: https://neptunemutual.com/blog/ethereum-classic-51-attacks/?utm_source=chatgpt.com
[36] S. Zhang and J.-H. Lee, "Double-spending with a sybil attack in the bitcoin decentralized network," *IEEE transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5715–5722, 2019.
[37] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 3–16.
[38] D. News, "Sybil attackers loot airdrops for millions with fake wallets," 2023, accessed: 2025-07-01. [Online]. Available: https://www.dlnews.com/articles/defi/sybil-attackers-loot-airdrops-for-millions-with-fake-wallets/
[39] Y. Marcus, E. Heilman, and S. Goldberg, "Low-resource eclipse attacks on ethereum's peer-to-peer network," in *Proceedings of the 2018 Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2018. [Online]. Available: https://www.ndss-symposium.org/ndss2018/ndss-2018-programme/low-resource-eclipse-attacks-on-ethereums-peer-to-peer-network/
[40] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *24th USENIX Security Symposium (USENIX Security 15)*. USENIX Association, 2015, pp. 129–144.
[41] ——, "Eclipse attacks on {Bitcoin's}{peer-to-peer} network," in *24th USENIX security symposium (USENIX security 15)*, 2015, pp. 129–144.
[42] Chainalysis, "Multichain exploit: Possible hack or rug pull," 2023, accessed: 2025-07-01. [Online]. Available: https://www.chainalysis.com/blog/multichain-exploit-july-2023/
[43] Ethereum Classic Cooperative, "Etc response to recent 51% attacks," https://www.coindesk.com/markets/2020/08/06/ethereum-classic-suffers-second-51-attack-in-a-week/, 2020, https://etccooperative.org/posts/2020-08-08-etc-response-to-recent-51-attacks/.
[44] PeckShield, "Multichain hack: Over $126m drained from bridge via compromised keys," https://twitter.com/peckshield/status/1679338918886793216, 2023, https://blockworks.co/news/multichain-hack-126-million.
[45] K. Nicolas, Y. Wang, G. C. Giakos, B. Wei, and H. Shen, "Blockchain system defensive overview for double-spend and selfish mining attacks: A systematic approach," *IEEE Access*, vol. 9, pp. 3838–3857, 2020.
[46] J. H. Mosakheil, "Security threats classification in blockchains," 2018.
[47] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Cryptocurrency Whitepaper*, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[48] N. Mutual, "Analysis of the miner exploit," 2024, accessed: 2025-07-01. [Online]. Available: https://neptunemutual.com/blog/analysis-of-the-miner-exploit/
[49] K. W. Wu, "Strengthening defi security: A static analysis approach to flash loan vulnerabilities," *arXiv preprint arXiv:2411.01230*, 2024.
[50] E. Deirmentzoglou, G. Papakyriakopoulos, and C. Patsakis, "A survey on long-range attacks for proof of stake protocols," *IEEE access*, vol. 7, pp. 28 712–28 725, 2019.
[51] V. Buterin and V. Griffith, "Casper the friendly finality gadget," https://arxiv.org/abs/1710.09437, 2018.
[52] F. Leone, G. Ateniese, and S. Cimato, "Securing proof-of-stake blockchains with forward-secure signatures," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4774–4786, 2021.
[53] S. Azouvi and M. Vukolić, "Pikachu: Securing pos blockchains from long-range attacks by checkpointing into bitcoin pow using taproot," in *Proceedings of the 2022 ACM Workshop on Developments in Consensus*, 2022, pp. 53–65.
[54] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 931–948.
[55] R. P. Puneeth and G. Parthasarathy, "A cross-chain-based approach for secure data sharing and interoperability in electronic health records using blockchain technology," *Computers and Electrical Engineering*, vol. 117, p. 108331, 2024.
[56] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and D. Mohaisen, "Exploring the attack surface of blockchain: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1977–2008, 2020.

[57] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 375–392.
[58] T. Neudecker and H. Hartenstein, "Short paper: An empirical analysis of blockchain forks in bitcoin," in *Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers 23*. Springer, 2019, pp. 84–92.
[59] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *Proceedings of the 26th symposium on operating systems principles*, 2017, pp. 51–68.
[60] M. K. Siam, B. Saha, M. M. Hasan, M. J. Hossain Faruk, N. Anjum, S. Tahora, A. Siddika, and H. Shahriar, "Securing decentralized ecosystems: A comprehensive systematic review of blockchain vulnerabilities, attacks, and countermeasures and mitigation strategies," *Future Internet*, vol. 17, no. 4, 2025. [Online]. Available: https://www.mdpi.com/1999-5903/17/4/183
[61] N. Li, M. Qi, Z. Xu, X. Zhu, W. Zhou, S. Wen, and Y. Xiang, "Blockchain cross-chain bridge security: Challenges, solutions, and future outlook," *Distributed Ledger Technologies: Research and Practice*, vol. 4, no. 1, pp. 1–34, 2025.
[62] MANRS, "Klayswap: Another bgp hijack targeting crypto wallets," 2022, accessed: 2025-07-01. [Online]. Available: https://manrs.org/2022/02/klayswap-another-bgp-hijack-targeting-crypto-wallets/
[63] C. Pérez-Solà, S. Delgado-Segura, G. Navarro-Arribas, and J. Herrera-Joancomartí, "Double-spending prevention for bitcoin zero-confirmation transactions," *International Journal of Information Security*, vol. 18, no. 4, pp. 451–463, 2019.
[64] I. Osipkov, E. Y. Vasserman, N. Hopper, and Y. Kim, "Combating double-spending using cooperative p2p systems," in *27th international conference on distributed computing systems (ICDCS'07)*. IEEE, 2007, pp. 41–41.
[65] C. Zhao, S. Zhang, T. Wang, and S. C. Liew, "Bodyless block propagation: Tps fully scalable blockchain with pre-validation," *Future Generation Computer Systems*, vol. 163, p. 107516, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X24004801
[66] G. Liao, T. Wang, Q. Yang, Y. Xia, L. Shi, X. Zhao, X. Wu, S. Zhang, A. Chan, and R. Yuen, "Programming on bitcoin: A survey of layer 1 and layer 2 technologies in bitcoin ecosystem," *arXiv preprint arXiv:2409.19622*, 2024.
[67] V. Valastin, D. Morháč, K. Košt'ál, and I. Kotuliak, "Protocol for unifying cross-chain liquidity on polkadot," *Frontiers in Blockchain*, vol. 7, p. 1413840, 2024.
[68] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 906–917. [Online]. Available: https://doi.org/10.1145/2382196.2382292
[69] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *IEEE P2P 2013 Proceedings*. IEEE, 2013, pp. 1–10.
[70] "What is a Vector 76 Attack." [Online]. Available: https://nfting.medium.com/what-is-a-vector-76-attack-d658db4ff9cd
[71] N. Mutual, "Analysis of the pike finance exploit," https://neptunemutual.com/blog/analysis-of-the-pike-finance-exploit/, May 2024, accessed July 2025.
[72] DefiLlama, "Total value locked all chains," 2024, https://defillama.com/chains Accessed 11/2024.
[73] ConsenSys Software Inc., "Ganache - ONE CLICK BLOCKCHAIN," 2024, https://archive.trufflesuite.com/ganache/ Accessed 11/2024.
[74] ——, "Truffle Suite," 2024, https://archive.trufflesuite.com/ Accessed 11/2024.
[75] Web3, "Web3.js - Ethereum JavaScript API," 2024, https://web3js.readthedocs.io/en/v1.10.0/ Accessed 11/2024.
[76] O. Kuznetsov, A. Rusnak, A. Yezhov, D. Kanonik, K. Kuznetsova, and S. Karashchuk, "Enhanced security and efficiency in blockchain with aggregated zero-knowledge proof mechanisms," *IEEE Access*, 2024.
[77] P.-W. Chi, Y.-H. Lu, and A. Guan, "A privacy-preserving zero-knowledge proof for blockchain," *Ieee Access*, vol. 11, pp. 85 108–85 117, 2023.
[78] S. Tyagi and M. Kathuria, "Role of zero-knowledge proof in blockchain security," in *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, vol. 1. IEEE, 2022, pp. 738–743.
[79] M. Herlihy, "Atomic cross-chain swaps," in *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, ser. PODC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 245–254. [Online]. Available: https://doi.org/10.1145/3212734.3212736