2025 17th International Conference on COMmunication Systems and NETworks (COMSNETS) | 979-8-3315-3119-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/COMSNETS63942.2025.10885767

Flexiswap: A Cross-Chain Decentralized Exchange Protocol

Kirti Singh*, Vinay J. Ribeiro*, Susmita Mandal†
* Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, Mumbai, India
Email: kirtisingh@cse.iitb.ac.in, vinayr@iitb.ac.in
† Institute for Development and Research in Banking Technology, Hyderabad, India
Email: msusmita@idrbt.ac.in

***Abstract*—Decentralized exchanges (DEXs) have emerged as a promising solution to enhance trustlessness in blockchain ecosystems and mitigate security threats associated with centralized exchanges. While platforms like Uniswap offer certain advantages, their limitation to exchanging tokens within a single blockchain restricts their utility. Moreover, exchanges relying on atomic swaps, such as AtomicDEX, are susceptible to grief attacks and exhibit low success ratios. In response to these challenges, we introduce FlexiSwap, an auction-based decentralized exchange protocol that leverages atomic swaps for exchanging native tokens between smart contract-compatible blockchains. FlexiSwap stands out by combining multiple bids to facilitate one-to-many exchanges, resulting in improved exchange values, successful execution of large-volume exchanges, and an improved success ratio. The protocol incorporates a penalization mechanism to deter grief attacks, ensuring that participants cannot withdraw from an exchange without incurring a loss once the counterparty has committed its assets. We also prove the protocol's safety, liveness, and grief prevention. Our simulation results demonstrate that combining multiple bids helps achieve better exchange values and order-matching success rates. Additionally, we implemented our protocol on Ethereum-based blockchains. The experimental results show that combining bids increases overall gas usage, but the increase in the average gas user per bid is marginal.**

***Index Terms*—decentralized exchange, atomic swap, hashed time-locked contracts, griefing attack, on-chain exchange**

I. Introduction

Cryptocurrency holders frequently exchange tokens across blockchains with centralized platforms like Binance and CoinDCX [1]. However, such exchanges introduce centralization, contradicting the decentralized ethos of blockchains. Historical failures of centralized exchanges, such as MTGox [2] and FTX [3], have underscored the risks, making decentralized exchanges (DEXs) a crucial alternative.

DEXs enable direct token exchanges without intermediaries, utilizing blockchain-based smart contracts [4] to govern transactions. Two major DEX models exist: liquidity pool-based and orderbook-based. Liquidity pool DEXs, like Uniswap [5], operate through token pools where traders provide liquidity and earn fees. However, Uniswap is limited to a single blockchain, while Stargate [6], a LayerZero-based DEX, enables cross-chain exchanges but relies on external relayers [1] and oracles [7], potentially weakening security.

Orderbook-based DEXs use atomic swaps [8] to facilitate trades after order matching. Based on hashlock and timelock mechanisms, atomic swaps ensure secure exchanges but are prone to delays and griefing attacks(assumed to be intentional as parties get significant locking time), as seen with platforms like AtomicDEX. These swaps are limited to one-to-one trades and have low success rates due to speculative delays.

Motivation:

- Atomic swap-based DEXs restrict trades to one-to-one exchanges, hindering large-volume trades and optimal exchange values.
- Current atomic swap DEXs suffer from low success rates [9] due to speculative delays, griefing attacks and lacking support for large-volume trades.

Contributions:

- Developed FlexiSwap, the first cross-chain DEX with one-to-many swaps and an on-chain auction mechanism.
- Introduced a hedged atomic swap-based mechanism [10] to mitigate griefing attacks and enhance success rates.
- Enabled partial exchanges, improving flexibility in trades.
- Provided proof of FlexiSwap's security, liveness, and griefing resistance.
- Deployed FlexiSwap on EVM-based testnets (Ontology and Sepolia), demonstrating better exchange values and success rates through simulations.

Table I highlights FlexiSwap's advantages in extensibility, grief prevention, support for one-to-many swaps, and security, distinguishing it from existing DEX platforms.

II. Background

The rise of multiple blockchains and cryptocurrencies has created a vast market for speculative trading. Bitcoin [11], as the oldest, holds the highest value, while Ethereum's [12] adaptability through smart contracts secures its second place.

Centralized exchanges dominate trading space due to their speed but have drawbacks:

- Compromise blockchain principles and security.
- Vulnerable to attacks due to a single point of failure.
- Exercise full control over users' assets.

The authors would like to thank the Institute for Development and Research in Banking Technology (IDRBT) (project number: RD/0121-IDRBTB9-001) and IITB TRUST LAB (project number: DO/2024-SBST002-015) for financial support to conduct and present this research.

TABLE I: Comparison of FlexiSwap with other exchanges

Exchange	Extensibility	Grief Prevention	One-to-many swaps	Reduced Security
Uniswap	Limited to tokens within a chain	NA[a]	Yes[b]	No
Stargate	Across Smart Contracts compatible Blockchains	NA[a]	Yes[b]	Yes
AtomicDEX	Most Blockchains	No	No	No
Connext NXTP	Across Smart Contracts compatible Blockchains	No	No	No
FlexiSwap	Across Smart Contracts compatible Blockchains	Yes	Yes	No

[a]Not Applicable, [b]using liquidity pools

Decentralized exchanges (DEXs) address these issues but face challenges:

- Difficult order matching without a central entity.
- Exposure to new attacks, like griefing.
- Slower transactions with multiple on-chain steps.

Atomic Swaps offers a decentralized solution for token exchange, assuming that both parties already know each other.

A. Atomic Swap

Atomic Swap, proposed by Tier Nolan in 2013 [13], is essential to our Flexiswap protocol. It ensures atomicity and security in token exchanges between two blockchains using two primary mechanisms: timelocks and hashlocks [8]. Two participants lock their tokens using a shared hash of a secret on their respective blockchains, ensuring they can only be claimed by revealing the secret. Timelocks ensure that if either party abandons the exchange, they can reclaim their tokens after the timelock expires. The protocol requires careful timing coordination between chains, with one timelock set to expire later, ensuring both parties have enough time to complete the swap. The protocol steps are illustrated in Fig. 1.

While atomic swaps are widely applicable and can be extended using time-based cryptography [14] and multi-signature transactions [15], they are vulnerable to griefing attacks [8], where one party delays or abandons the exchange, causing the other party prolonged wait times to reclaim their tokens.

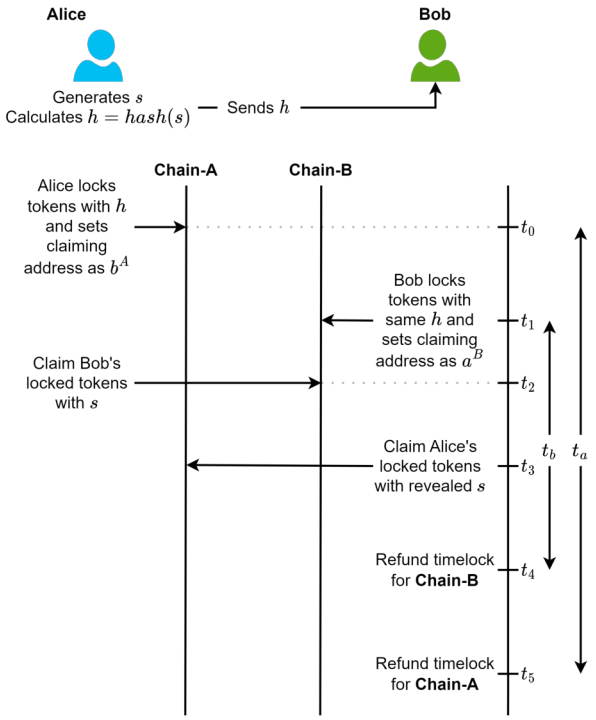

Fig. 1: A Simple Atomic Swap Protocol.

B. Griefing Attack Solutions

Several solutions have been proposed to address griefing attacks involving locking a small premium to safeguard against malicious behaviour. In 2021, Xue and Herlihy [10] introduced the Hedged Atomic Swap, where participants cross-lock premium amounts on each other's chains before locking collateral, allowing efficient claims and reducing the swap process to six steps. Nadahalli et al. [16] proposed a five-step Grief-free Atomic Swap, combining premium and collateral into a single transaction using Bitcoin's multi-signature feature, simplifying the process and reducing the risk of griefing. Majumdar et al. [9] developed Quickswap, which eliminates cross-locking premiums and introduces multiple hashes, allowing early exits if the swap becomes unfavourable. However, this increases the process to eight steps.

III. Related Work

Several decentralized exchange protocols have been developed with unique features and challenges. AtomicDEX [17] utilizes Komodo's DeFi framework for atomic swap-based decentralized exchanges. It employs a peer-to-peer network for order matching, with exchange rates obtained from centralized exchanges. The protocol incorporates a reputation system that assigns participant scores based on transaction behaviour, penalizing those who fail to complete transactions. Despite these features, AtomicDEX faces difficulties matching large orders, a low success ratio, and insufficient protection against griefing attacks. Connext NXTP [18] relies on off-chain bidding to match orders, where multiple bidders offer liquidity at varying price points. The bidder providing the lowest fee is selected, followed by a simple atomic swap. However, this protocol is susceptible to griefing attacks and struggles with large-volume trades due to the lack of a mechanism to aggregate multiple bidders. Uniswap is a popular decentralized Ethereum blockchain exchange known for its Automated Market Maker (AMM) system. Liquidity pools of ERC-20 token pairs enable users to swap tokens based on the constant product formula $x * y = k$, which adjusts exchange rates according to market demand and supply. While Uniswap v2

[5] and v3 [19] introduced efficiency improvements, Uniswap does not support cross-chain swaps. Stargate [6] is a cross-chain decentralized exchange that leverages LayerZero [20], an omni-chain protocol similar to Polkadot [21] and Cosmos [22]. Stargate facilitates cross-chain transactions using smart contract Endpoints, Oracles to transmit block headers, and Relayers to verify transaction proofs. It enhances security by enabling users to select multiple Oracle-Relayer pairs, though a tradeoff between security and performance remains.

IV. FLEXISWAP

Flexiswap is a decentralized protocol enabling secure token exchanges without relying on trusted third parties. It utilizes on-chain bidding on the order-initiating chain to match orders and optimize exchange rates. This bidding process prevents malicious activity and locks the griefing premium. A penalization mechanism based on hedged atomic swaps [10] ensures that a party cannot withdraw from the exchange once the counterparty's tokens are locked. Flexiswap also facilitates large-volume exchanges, supporting one-to-many transactions between a client and multiple bidders. Overall, Flexiswap offers a decentralized and secure solution for token exchanges across smart contract-compatible blockchains. The next section outlines its transaction flow.

TABLE II: Notation used in the protocol

Notation	Description
h_1	Hash generated by Alice
s_1	Preimage of h_1
a	Amount Alice wants to exchange
r_{mn}	Minimum Exchange rate set by Alice
K	Number of bids to be combined
i^B	Alice's $Chain\text{-}B$ address
p_a	Premium amount of Alice for each bidder
$p_a + p_b$	Premium amount of each bidder
r'_i	Exchange rate of i^{th} bidder
h'_i	Hash generated by i^{th} bidder
s'_i	Preimage of s'_i
mx_i	Maximum liquidity provided by i^{th} bidder
n	Total winning bidders of a swap
r_i	Exchange rate of i^{th} bidder
b_i^A	$Chain\text{-}A$ address of i^{th} bidder
t_i	Timestamps in the chains
δ_i	Time window between two successive timestamps

A. Protocol

In the Flexiswap protocol, token exchange follows a sequence of steps with designated timeouts for completion, as shown in Fig. 2. Alice, seeking to exchange a units of $token\text{-}A$ on $Chain\text{-}A$ for tokens on $Chain\text{-}B$, engages in a bidding process, after which the smart contract selects the winning bidders for the exchange. To mitigate griefing attacks, both Alice and the selected bidders must lock premiums: Alice locks p_a tokens for each winning bidder, while bidders lock $p_a + p_b$, where p_a is Alice's potential loss if she causes grief, and p_b is the penalty for a bidder if it causes grief. The exchange proceeds stepwise with compliant parties.

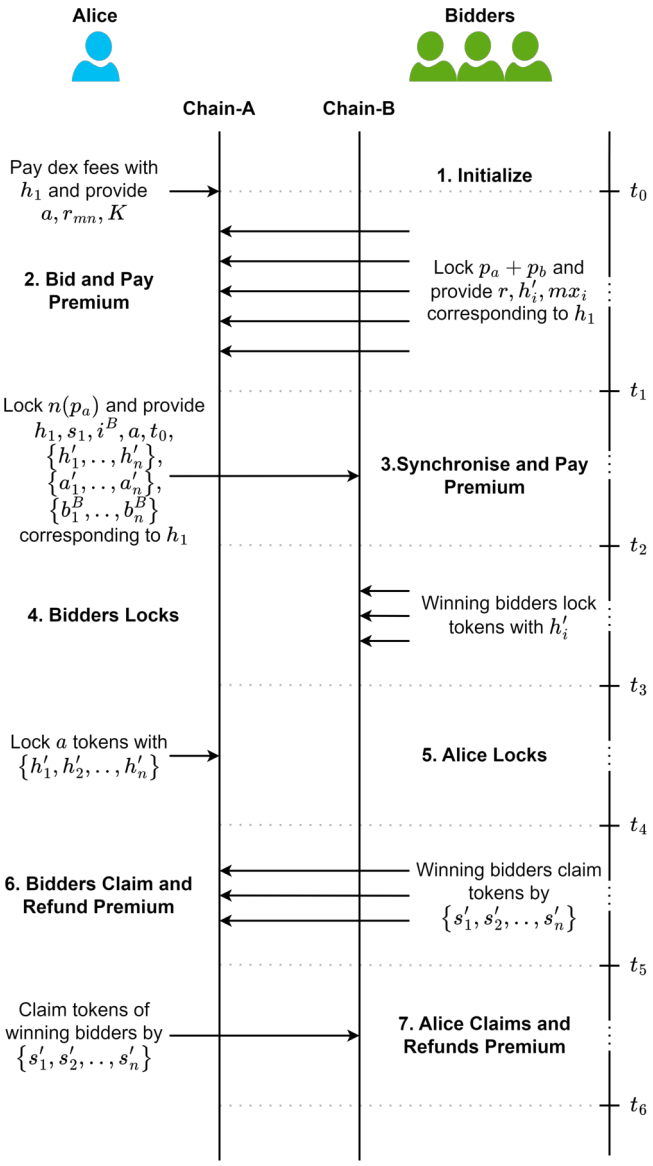

Fig. 2: Flow of exchange based on Flexiswap protocol.

1) **Initialise**: Alice initiates an exchange by selecting a secret value, represented as s_1, and obtaining its hash, denoted as $h_1 = hash(s_1)$. Then, Alice pays a fee to the decentralized exchange (dex) associated with h_1. The dex fee is determined based on the number of bids to be combined, denoted as (K), and is specified by Alice. In this transaction, Alice declares the amount she intends to exchange (a) and the minimum exchange rate she desires (r_{mn}) on $Chain\text{-}A$. Additionally, Alice provides her $Chain\text{-}B$ address, denoted as i^B.
This transaction triggers a bidding process on $Chain\text{-}A$ with an initiation time marked as t_0. Timestamps t_1, t_2, t_3, t_4, t_5, and t_6 are set as timeouts in $Chain\text{-}A$, with t_5 also serving as a refund timelock for Alice on $Chain\text{-}A$. The primary purpose of the dex fee is to deter undesired network spam, and dependence on K is to

restrict Alice from keeping very large values of K.

2) **Bid and Pay premium**: Bidders participate in the auction by locking a premium of $p_a + p_b$ corresponding to h_1, which the system will refund if a superior bid emerges. Additionally, each bidder provides essential information, including the exchange rate r_i, a new hash value h'_i (using secret s'_i only known to bidder i), and the maximum amount of $token\text{-}B$ they are willing to offer (mx_i). The converted value of mx_i into $token\text{-}A$ using r_i must be greater than or equal to a/K to ensure that a maximum of K bidders can fill the exchange. Based on mx_i and r_i, the smart contract calculates a^b_i, the $token\text{-}B$ each bidder must lock. r_i must be greater than the minimum exchange rate specified by Alice. Bidders can place bids on $Chain\text{-}A$ until the timeout t_1 is reached. As bids are submitted, the smart contract updates and may change the winning bidders and the amount they have to lock. The final state is reached upon the expiration of the bidding timer.
3) **Synchronise and Pay premium**: After bidding ends, suppose n bidders win the auction. Alice locks premium p_a on $Chain\text{-}B$ corresponding to h_1. She additionally sends the data $< s_1, i^B, a, t_0, (h'_1, ..., h'_n), (a^b_1, ..., a^b_n), (b^B_1, ..., b^B_n) >$ to synchronize $Chain\text{-}B$. Secret s_1 is required to ensure only Alice can provide the data. This step sets up the timeouts t_2, t_3, t_4, t_5 and t_6 on $Chain\text{-}B$ where t_6 also works as refund timelock for bidders. The bidders have to verify the data sent by Alice for synchronizing before moving to the next step. If Alice sends incorrect data for synchronization, bidders can refund their premium amount after t_4 and leave the exchange.
4) **Bidders Locks**: After verifying the synchronization step, each winning bidder i locks its a^b_i $token\text{-}B$ using h'_i on $Chain\text{-}B$ corresponding to h_1 within t_3 according to the rate r_i. t_6 is the refund timelock for the bidder's locked tokens on $Chain\text{-}B$. For each winning bidder that does not lock tokens before t_3, Alice's premium p_a for that bidder is unlocked. Thus, Alice's premium for a bidder stays locked when the bidder locks collateral.
5) **Alice Locks**: Alice then locks a $token\text{-}A$ for bidders using $(h'_1, ..., h'_n)$ on $Chain\text{-}A$ corresponding to h_1 within t_4. t_5 is the refund timelock for Alice's locked tokens on $Chain\text{-}A$. If Alice does not lock for a bidder j before t_4, bidder j's premium $p_a + p_b$ is unlocked. Thus, the bidder's premium stays locked when Alice locks collateral for the bidder.
6) **Bidders Claim and Refund Premium**: Winning bidders after locking their tokens, claim $token\text{-}A$ from $Chain\text{-}A$ corresponding to h_1 within t_5 using s'_i. Claiming $token\text{-}A$ unlocks the bidder's premium and is refunded in the same step.
7) **Alice Claims and Refund Premium**: Alice then claims $token\text{-}B$ from $Chain\text{-}B$ corresponding to h_1 within t_6 and gets her premium back using the secrets $(s'_1, s'_2, ..., s'_n)$ which were revealed in the previous step. Claiming $token\text{-}B$ unlocks Alice's premium for the claimed bidders, which is also refunded in the same step.
8) **Refund Tokens and Claim Premium**: Alice on $Chain\text{-}A$ can refund her locked $token\text{-}A$ that are not exchanged corresponding to h_1 after refund timelock t_5. Additionally, the premium amount of bidders who caused grief is claimed in the same step as compensation. Bidders can refund their leftover $token\text{-}B$ on $Chain\text{-}B$ corresponding to h_1 after the refund timelock t_6. They also get the premium amount of Alice for griefing as compensation in the same step.

A party must proceed to the next step only when the previous transaction is confirmed and the counterparty faithfully adheres to the protocol. The above exchange flow does not consider the cases where the participating parties may leave the exchange in between. The participating parties can leave at various points in the exchange, causing problems for the counterparty. The following section discusses the cases that can arise when different parties leave between an exchange.

B. Non-Compliant Scenarios

At any exchange stage, a party may abandon the process, causing grief to the counterparty. These parties are referred to as non-compliant. The penalization mechanism addresses this by allowing the counterparty to claim the locked premium from the non-compliant party. Table III outlines the total penalties in various scenarios of claiming and refunding the premiums for Alice and the bidders.

- **Case 1: k bidders lock tokens**
  If k out of n bidders lock tokens before t_3, Alice's premium for other bidders has to be returned. Thus, $(n-k)p_a$ is unlocked to Alice after t_3. Since Alice will also not lock, the premium of $(n-k)$ bidders will be unlocked and can be refunded after t_4. Each of $(n-k)$ bidders can claim $(p_a + p_b)$ after t_4.
  No penalization is required even if $(n-k)$ bidders do not lock their collateral because no party is griefing.
- **Case 2: Alice locks for m bidders**
  If Alice locks collateral for m out of k bidders, the $(k-m)$ bidders' premium is unlocked after t_4. However, Alice must be penalized for these $(k-m)$ bidders as they already have locked their collateral. This is done after t_6 as Alice's premium will remain locked till then for these bidders.
- **Case 3: q bidders claim tokens**
  The premium of q bidders will be unlocked and claimed in the same step, along with the collateral. Premium of $(m-q)$ bidders will stay locked because of not claiming, which Alice can get after t_5 by claiming premium transaction. Since Alice also cannot claim from $(m-q)$ bidders, Alice's premium for these bidders will stay locked till t_6. $(m-q)$ bidders can claim it after t_6. But Alice will eventually gain $(m-q)(p_a+p_b) - (m-q)(p_a) = (m-q)p_b$. Due to this case, bidders must lock more amounts as a premium in the protocol.
- **Case 4: Alice claims from r bidders**

If Alice only claims from r out of q bidders, the bidders can claim Alice's premium after t_6. Thus, Alice eventually loses p_a each to $(q-r)$ bidders after t_6.

TABLE III: Overall penalization in different cases of griefing

Case	Penalization
k bidders lock tokens	No penalisation
Alice locks for m bidders	Alice lose $(k-m)p_a$
q bidders claim tokens	$(m-q)$ bidders lose p_b
Alice claims from r bidder	Alice lose $(q-r)p_a$

V. Analysis

Flexiswap provides improved exchange rates and ensures the liveness and safety of the swap, effectively preventing griefing attacks by cross-locking premium amounts. The following section presents the analysis of these aspects.

A. Protocol Correctness

The Flexiswap protocol assumes that each window between two successive timestamps is significantly bigger than the blockchain confirmation time, i.e. $\delta_{(i)} > t_{conf}$ where $\delta_i = t_{(i)} - t_{(i-1)}$ and t_{conf} is blockchain confirmation time. This ensures every next step in the protocol is performed only after the previous step is complete and finalized in the blockchain. This section explains and proves the properties of safety and liveness in the context of the Flexiswap protocol, thus proving the correctness of the protocol

Property 1: (Safety) Safety ensures that protocol-compliant parties do not lose their locked tokens
Flexiswap ensures exchange security through hashlocks, timelocks, timeouts, and pre-setting the claimant's address in smart contracts. In Flexiswap, only bidder i knows their secret s_i', preventing Alice from claiming their collateral until they claim hers. If all parties comply, premiums are refunded, collateral exchanged, with penalties imposed only for griefing.

Lemma 1. *For any protocol-compliant party, such as Carol, the Flexiswap protocol guarantees the safety of her locked tokens.*

Proof. Consider a protocol-compliant participant, Carol, engaging in Flexiswap protocol. Assuming the Lemma does not hold, Carol loses tokens using the Flexiswap protocol. There can be only four cases where she loses tokens:

1) *Premium loss when Carol initiates:* If a bidder requests a refund after t_6, Carol risks losing her premium p_a. However, the refund only occurs if Carol fails to lock tokens or the bidder fails to claim them. Assuming Carol complies with protocol, the focus is on the latter. If the bidder does not claim tokens, Carol receives the bidder's premium $p_a + p_b$ after t_5, gaining p_b as compensation.
2) *Collateral loss when Carol initiates:* Carol locks collateral only for bidders who have already locked tokens for her. If bidders claim her collateral, the secret is revealed on-chain, allowing Carol to reclaim it. If bidders don't claim, she can be refunded and receive compensation after t_5, ensuring she never incurs collateral loss.
3) *Premium loss when Carol bids:* Carol risks losing her locked premium only if both parties lock collateral and she refuses to claim. The initiator may claim Carol's premium $p_a + p_b$ as compensation after t_5. However, assuming Carol follows the protocol, this situation is avoided, as she will claim the tokens, ensuring no premium loss.
4) *Collateral loss when Carol bids:* Carol risks losing the collateral if the initiator claims tokens before t_6 without locking collateral by obtaining the preimage of Carol's hash. However, assuming the preimage and collision resistance of the hash functions, this scenario is invalid, as the initiator cannot access the secret without Carol's disclosure.

There is no case in which Carol, as either the bidder or the initiator, experiences a loss of premium or collateral. Thus, the assumption that the lemma does not hold is false. Therefore, if Alice is a protocol-compliant party, she will not incur any token loss in a Flexiswap-based exchange protocol. □

Property 2: (Liveness) The property of liveness pertains to the condition where assets held in custody by involved parties must not remain locked indefinitely, even in situations where one of the parties fails to comply with the agreed terms.
Flexiswap ensures liveness through timelocks. When both parties comply, collateral is exchanged, and premiums are refunded. If any collateral or premium remains locked, it can be unlocked after the refund timelocks expire.

Lemma 2. *Assuming Carol is some party, the Flexiswap protocol guarantees the liveness for Carol.*

Proof. Carol can be compliant or non-compliant and have either the initiator or bidder role. Assuming the Lemma does not hold, Carol's tokens remain locked in the protocol indefinitely. The following cases can occur:

1) *Indefinite premium lock when Carol initiates:* After Carol locks the premium, the following cases arise:
   - After t_3, Carol calls for a refund on her premium if bidders don't lock their collateral.
   - Before t_6, Carol calls for a refund on her premium if everyone follows all the protocol steps.
   - After t_6, the bidder claims Carol's premium when Carol fails to refund.

   The first two cases will always succeed if called correctly, as the refund is immutably returned to the funding party. For the third case to fail, the bidder's address in $Chain\text{-}B$ would need to differ, which is impossible since bidders lock collateral only after verifying Carol's information. Therefore, all three cases hold, ensuring the premium will not be locked indefinitely.
2) *Indefinite collateral lock when Carol initiates:* After Carol locks the collateral, the following cases arise:
   - Before t_5, Bidders call the claim on Carol's collateral by providing the secret.

- After t_5, Carol calls for a refund on her collateral.

Even if the bidders fail to claim the tokens in the first case, the second case ensures Carol gets her refund. Thus, collateral is not locked forever when Carol acts as an initiator.

3) *Indefinite premium lock when Carol bids:* After Carol locks the premium $p_a + p_b$, the following case arises:
   - After t_4, Carol calls for a refund on her premium if the initiator does not lock its collateral.
   - Before t_5, Carol refunds her premium along with collateral if the initiator locks its collateral.
   - After t_5, the initiator claims Carol's premium when Carol fails to refund.

   The first two cases will always succeed when appropriately called since the refund is irreversibly sent back to the funding party. For the third case to fail, the initiator's address on $Chain\text{-}A$ would have to be different, which is impossible as it is established during the initial transaction. Consequently, all three cases are valid, ensuring that the premium will not remain locked indefinitely.
4) *Indefinite collateral lock when Carol bids:* After Carol locks the collateral, the following cases arise:
   - Before t_6, the initiator calls a claim on Carol's collateral, providing the secrets.
   - After t_6, Carol calls for a refund on her collateral.

   Even if the initiator fails to claim Carol's collateral, Carol can refund her tokens after t_6. The second case can only fail if Carol's address is changed in the blockchain, which is impossible. Thus, collateral will not stay locked indefinitely in this case.

Either premium or collateral does not remain locked indefinitely. Thus, the assumption that the Lemma does not hold is false. Therefore, for any Carol, tokens will not stay locked indefinitely in the Flexiswap protocol. □

B. Griefing Attack

A griefing attack is caused when a party refuses to continue the exchange when the counterparty has locked the tokens [8]. This locks the counterparty's tokens for a long time, causing grief. The locked tokens can not be used for any other purpose until locked in the swap.

Lemma 3. *Assuming one or more non-compliant parties, the Flexiswap protocol prevents the griefing attack.*

Proof. Assuming the lemma does not hold, the Flexiswap protocol does not prevent the griefing attack. In such a scenario, the following case arises:

1) *No compensation for bidder when Alice does not lock collateral:* If a bidder locks collateral before t_3 and Alice fails to lock before t_4, the bidder can refund their premium before t_4 and subsequently claim Alice's premium p_a after t_6. Thus, the bidder gets compensation for grief.
2) *No compensation for Alice when bidder does not claim:* If both Alice and the bidder lock their collateral, and the bidder fails to claim before t_5, Alice can refund her collateral and claim the bidder's premium $p_a + p_b$ after t_5. The bidder can then claim Alice's premium p_a and refund their collateral P_b after t_6, resulting in a loss of premium p_b. Thus, Alice gets compensation for grief.

There is no case where the parties suffer from the griefing attack on their locked collateral amount without getting compensation. Thus, the assumption that the lemma does not hold is false. Therefore, the Flexiswap protocol prevents griefing attacks by penalizing the non-compliant parties. □

C. Exchange Value

Flexiswap employs auctions to obtain exchange rates from multiple bidders, who submit their best rates and the maximum amount of tokens they can provide. The protocol selects the bidder offering the best exchange value. If the leading bidder cannot supply enough tokens, multiple bidders with the highest rates are combined to complete the exchange. This auction-based approach yields better prices compared to relying on a single partner.

Flexiswap achieves a higher success rate than other atomic swap-based exchanges by effectively preventing griefing attacks, facilitating large-volume exchanges, and offering superior exchange values.

VI. Implementation and Experiments

Our implementation operates on two Ethereum Virtual Machine (EVM) [23] compatible test networks: Ontology [24] and Sepolia. On Ontology, we deployed a smart contract named $FlexiONT$, while $FlexiETH$ is deployed on Sepolia. Both contracts, coded in Solidity [25], follow the procedural steps outlined in Section 2. The SHA256 hashing algorithm is natively supported on both testnets.

Live auctions are stored by a MongoDB Atlas Cluster [26], where users can verify auction details on the blockchain. Account management and transaction initiation utilize Metamask [27] and RPC nodes [28], supplemented by a user-friendly dashboard for interaction and exchange monitoring.

Functions are written in smart contracts per the protocol and should be called in the same sequence by the parties in the two smart contracts.

A. Simulation

The simulation aims to demonstrate that combining bids leads to improved exchange values and higher success rates in order matching. We utilize data from the Uniswap WBTC-ETH pool [29] for this simulation due to its substantial amount; however, other pools with significant data could also have been used. The initiator's WBTC value is randomly selected from WBTC to ETH exchange data, with the corresponding exchange rate set as the minimum exchange rate. We then use the ETH to WBTC exchange data from the same pool to determine ETH values as the maximum ether, setting the corresponding exchange rate as the bid exchange rate. The maximum ether values are constrained within three hours from the initiation timestamp to limit the bids for the exchange.

The simulation analyzes values for each exchange using a naive method, and our method of combining K bids for $K = 1$ and $K = 5$. The naive method matches exchange orders based on the first bid that successfully fulfils the exchange. At the same time, our approach combines bids by waiting for the auction to complete and matching orders based on the value of K, the highest exchange rate, and the maximum ETH from multiple bids. The simulation is repeated with random WBTC values multiple times, allowing us to obtain exchange values for both the naive method and our approach of combining bids for $K = 1$ and $K = 5$.

We first examine the percentage increase in the exchange value offered by both methods with the minimum value expected by Alice. We select the successful exchanges in the three ways and compute the increase in the exchange value.

$$f = 100 * \frac{1}{n} \sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)}{y_i} \tag{1}$$

Considering n exchanges successful for each of the three ways, for each exchange i, (1) calculates the percentage increase in exchange value. $\hat{y}_i$ represents the new exchange value, and y_i is the minimum exchange value set by the initiator(product of minimum exchange rate and initiator exchange amount). We calculate f for the naive method, clubbing method($K = 1$) and clubbing method($K = 5$). In Fig. 3, the amount of WBTC is put on the x-axis and divided into bins of equal size. The percentage increase is calculated for each bin. Fig. 3 shows that the method of clubbing bids yields better exchange value than the naive method. Also, increasing the value of K yields better exchange value.

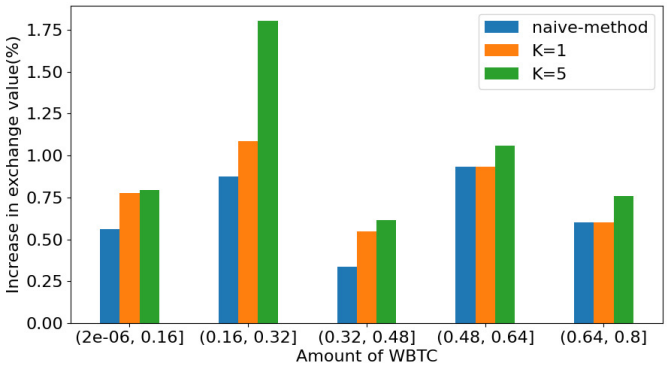

Fig. 3: Comparison of percentage increase in the exchange value relative to initiator's proposed minimum exchange value of naive and clubbing bids methods

Secondly, we examine the order-matching success rate of the two methods from the simulation results. The success rate is calculated for each x-axis bin by dividing the total number of successful exchanges by the total number of exchanges. Fig. 4 shows that the success rate for the naive and clubbing($K = 1$) methods is the same for all bins in the x-axis. This similarity is because, for a successful exchange, both require a bid that can completely fulfil the initiator's exchange amount. Since the same bids are used, if such a bid exists for the naive method, it also exists for the clubbing method($K = 1$). However, when $K = 5$, the success rate improves as lower volume bids also fulfil the exchange by combining with other bids.

Fig. 4 shows that increasing the value of K improves the order-matching success rate and provides a better exchange value to the initiator. The experimental analysis of the protocol helps understand the variation of gas used with different values of K.

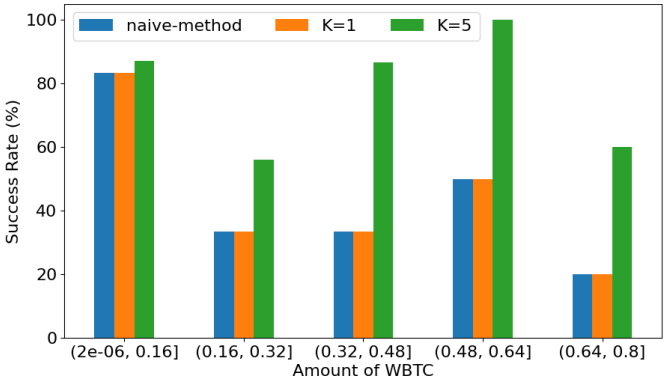

Fig. 4: Comparison of the success rate of naive and clubbing bids methods

B. Experiment

The experiment investigates how overall gas consumption varies with K. Initial values are generated similarly to those in the simulation. The Ontology smart contract is deployed locally on Ganache, a local Ethereum blockchain, to measure gas usage for various transactions. For each exchange amount, an auction is conducted for different values of K, recording the gas consumed for each bid. This process is repeated multiple times using randomly selected WBTC values from the WBTC to ETH exchange data on Uniswap. We first analyze the variation in total gas consumption across auctions as K and the exchange amount change.

$$G = \frac{1}{m} \sum_{i=1}^{m} \left(\sum_{j} b_{i,j} \right) \tag{2}$$

For each exchange, denoted by i, with j bids, equation (2) calculates the total gas used, represented by G. This calculation is performed considering a total of m exchanges successful for all values of K. The experiment is repeated for each value of K, and the total gas used is calculated for each K. In Fig. 5, the amount of WBTC is put on the x-axis, which is then divided into bins of equal size. Thus, the total gas used is calculated for each bin. Analyzing Fig. 5, it becomes evident that the gas usage in the auction rises with an increase in K. This is because larger values of K lead to the success of smaller bids, thereby increasing the overall number of successful bids and consequently raising the total gas usage. Total gas usage is comparably more in the bin (0.136,0.272] because of more bids in the range. The number

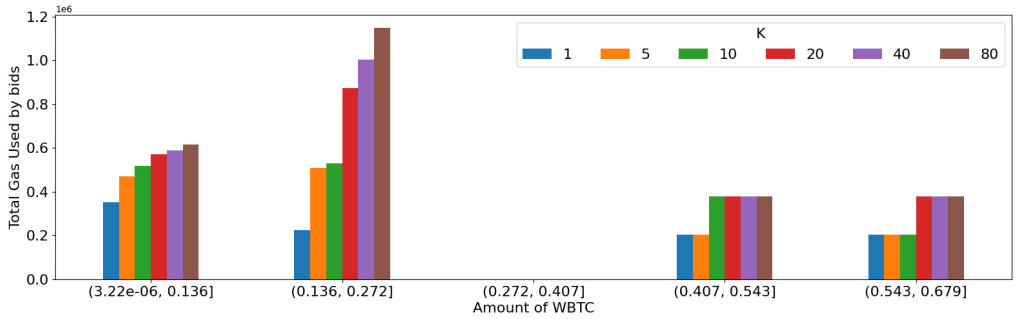

Fig. 5: Comparison of total gas used by bids for various K values and exchange amounts

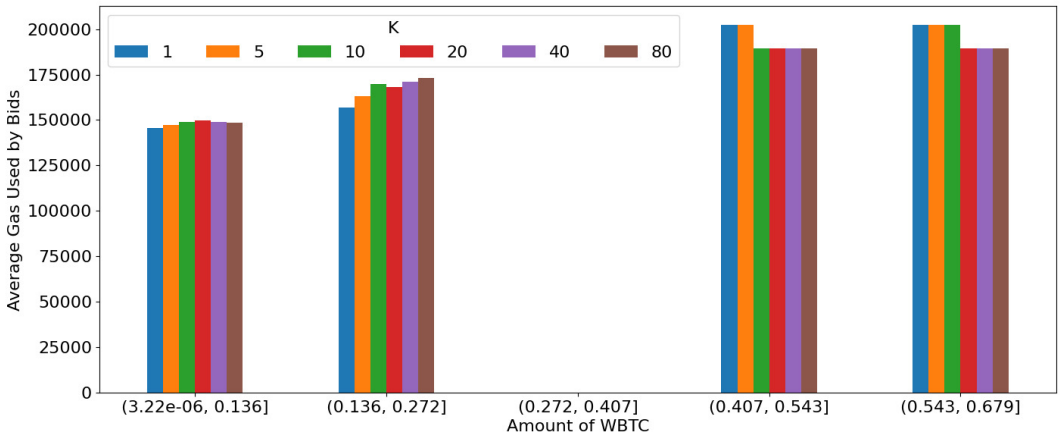

Fig. 6: Comparison of average gas used by bids for various K values and exchange amounts

of bids decreases as the exchange amount increases, reducing the total gas usage. Also, the same total gas usage for different values of K implies that the same number of bids are clubbed. We now calculate the average gas each bid uses for different values of K under different exchange amounts.

$$G_{avg} = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{1}{p_i} \sum_{q=1}^{p_i} b_{i,q} \right) \tag{3}$$

Considering a total of m exchanges that are successful for all values of K, (3) calculates the average gas used by each bid of all the exchanges, denoted by G_{avg}. In (3), p_i is the number of bids in a particular exchange i, and $b_{i,q}$ is the bid q for exchange i. The experiment is repeated for each value of K, and the average gas used by bids is calculated for each K. In Fig. 6, the x-axis represents the quantity of WBTC, which is subsequently divided into bins of uniform size. Thus, the average gas used by each bid is computed for each bin. Examining Fig. 6, it becomes apparent that as the value of K increases, there is a marginal rise in the average gas used by each bid when the quantity of WBTC is small. This is because the bidding is implemented in an efficient heap data structure. However, for larger values of WBTC, the bids in the bins (0.407, 0.543] and (0.543, 0.679] are fewer. Since the first bid consumes more gas than the subsequent bids, for larger WBTC values, the gas from the initial bid cannot be distributed, leading to a higher average gas usage.

Thus, as K increases, the increase in average gas usage per bid is marginal even though the total gas usage increases.

VII. Conclusions

Flexiswap is the first protocol that enables one-to-many decentralized exchanges between blockchains, supporting the creation of exchange platforms for smart contract-based blockchains. Traditional atomic swap-based exchanges encounter challenges of low success rates and susceptibility to griefing attacks. To address these issues, Flexiswap employs an auction mechanism to match orders based on the highest exchange value. Simulations demonstrate that its one-to-many swap capability improves order-matching success rates by combining multiple bids and facilitating large-volume exchanges. Moreover, experiments show that when the value of K increases, the overall gas usage increases moderately, but the increase in the average gas used by each bid is marginal. Additionally, Flexiswap addresses security concerns by implementing a robust penalization mechanism, preventing griefing attacks and maintaining crucial properties such as safety, liveness, and grief prevention.

REFERENCES

[1] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–41, 2021.

[2] S. Rao and C. Shaen, "Mt. gox–the fall of a giant," *Understanding Cryptocurrency Fraud, edited by Shaen Corbet*, pp. 71–82, 2022.

[3] S. Fu, Q. Wang, J. Yu, and S. Chen, "Ftx collapse: a ponzi story," in *International Conference on Financial Cryptography and Data Security*. Springer, 2023, pp. 208–215.

[4] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020.

[5] H. Adams, N. Zinsmeister, and D. H. Robinson, "Uniswap v2 core," 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:221835652

[6] R. Zarick, B. Pellegrino, and C. Banister, "Delta: Solving the Bridging Trilemma," https://www.dropbox.com/s/gf3606jedromp61/Delta-Solving.The.Bridging-Trilemma.pdf?dl=0, Mar. 2022, accessed: Nov. 25, 2023.

[7] H. Al-Breiki, M. H. U. Rehman, K. Salah, and D. Svetinovic, "Trustworthy blockchain oracles: review, comparison, and open research challenges," *IEEE access*, vol. 8, pp. 85 675–85 685, 2020.

[8] 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

[9] S. Mazumdar, "Towards faster settlement in htlc-based cross-chain atomic swaps," in *2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*. IEEE, 2022, pp. 295–304.

[10] Y. Xue and M. Herlihy, "Hedging against sore loser attacks in cross-chain transactions," in *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, ser. PODC'21. New York, NY, USA: Association for Computing Machinery, 2021, p. 155–164. [Online]. Available: https://doi.org/10.1145/3465084.3467904

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

[12] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[13] Tier Nolan, "Alt chains and atomic transfers," https://bitcointalk.org/index.php?%20topic=193281.0, accessed: November 25, 2023.

[14] S. A. Thyagarajan, G. Malavolta, and P. Moreno-Sanchez, "Universal atomic swaps: Secure exchange of coins across all blockchains," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1299–1316.

[15] P. Hoenisch, S. Mazumdar, P. Moreno-Sanchez, and S. Ruj, "Lightswap: An atomic swap does not require timeouts at both blockchains," in *International Workshop on Data Privacy Management*. Springer, 2022, pp. 219–235.

[16] T. Nadahalli, M. Khabbazian, and R. Wattenhofer, "Grief-free atomic swaps," in *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2022, pp. 1–9.

[17] Komodo, "Komodo DeFi Framework and Atomic Swaps," https://developers.komodoplatform.com/basic-docs/start-here/core-technology-discussions/atomicdex.html, Feb. 2021, accessed: Nov. 25, 2023.

[18] A. Bhuptani, "nxtp: A simpler xchain protocol," https://blog.connext.network/nxtp-a-simpler-xchain-protocol-88760697ea04, Aug. 2021, accessed: Jan. 10, 2024.

[19] H. Adams, N. Zinsmeister, M. Salem, R. Keefer, and D. Robinson, "Uniswap v3 core," *Tech. rep., Uniswap, Tech. Rep.*, 2021.

[20] R. Zarick, B. Pellegrino, and C. Banister, "Layerzero: Trustless omnichain interoperability protocol," *arXiv preprint arXiv:2110.13871*, 2021.

[21] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," *White paper*, vol. 21, no. 2327, p. 4662, 2016.

[22] J. Kwon and E. Buchman, "Cosmos whitepaper," *A Netw. Distrib. Ledgers*, vol. 27, 2019.

[23] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014.

[24] O. Foundation, "Ontology: A new high-performance public multi-chain project & a distributed trust collaboration platform," Ontology Network, Tech. Rep., 2020. [Online]. Available: https://ont.io/wp/Ontology-Introductory-White-Paper-EN.pdf

[25] C. Dannen, *Introducing Ethereum and solidity*. Springer, 2017, vol. 1.

[26] K. Banker, D. Garrett, P. Bakkum, and S. Verch, *MongoDB in action: covers MongoDB version 3.0*. Simon and Schuster, 2016.

[27] W.-M. Lee and W.-M. Lee, "Using the metamask chrome extension," *Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript*, pp. 93–126, 2019.

[28] K. Li, J. Chen, X. Liu, Y. R. Tang, X. Wang, and X. Luo, "As strong as its weakest link: How to break blockchain dapps at rpc service." in *NDSS*, 2021.

[29] "Wbtc-eth pair," https://v2.info.uniswap.org/pair/0xbb2b8038a1640196fbe3e38816f3e67cba72d94, accessed: Jan. 18, 2024.