IBE-IBE: Intent-Based Execution through Identity-Based Encryption and Auctions

Peyman Momeni
Fairblock, University of Waterloo
peyman@fairblock.network

Fig Smith
Squid
fig@squidrouter.com

February 14, 2025

Abstract

This paper introduces a *decentralized and leaderless sealed bid auction* model for dynamic pricing of intents across blockchain networks. We leverage *Multi-Party Computation* (MPC) and *Identity-Based Encryption* (IBE) to improve pricing while ensuring fairness and decentralization. By addressing the vulnerabilities of current centralized or static pricing mechanisms, our approach fosters transparent, secure, and competitive price discovery. We further enhance the confidentiality of intents through *Multi-Party Computation* (MPC), *Fully Homomorphic Encryption* (FHE), and *Trusted Execution Environments* (TEE). Our novel methodology mitigates the risks of frontrunning and centralization while preserving the rapid settlement times essential to decentralized finance (DeFi).

1 Introduction

Intent-based mechanisms are rapidly reshaping the landscape of decentralized finance (DeFi). By allowing users to express *intents*—which define how, when, and under what conditions trades should occur—protocols such as CoWSwap [15] and UniswapX [36] have captured a significant share of on-chain swap volumes. Additionally, cross-chain intent protocols like Squid [29], Across [2], Anoma [4] and deBridge [16] extend these benefits across multiple blockchains and tokens.

In many existing systems, when a user submits an intent, a centralized Request-For-Quote (RFQ) service or specialized relay sets a price. These sometimes fallback to on-chain Dutch auctions, where Solvers bid to fill intents.

Despite the promise and widespread adoption of intents, current intent-based pricing models exhibit notable shortcomings. Below we outline three critical challenges that motivate our work:

1. **Fairness and Decentralization of the Pricing Mechanism:** Many RFQ platforms rely heavily on centralized servers or specialized off-chain actors for price discovery, leaving users vulnerable to censorship and manipulation. On-chain auctions make the mechanism transparent and secure, shifting competition toward offering better prices (and thus better welfare for end users) rather than speed. A *leaderless sealed-bid* design prevents manipulative practices such as shilling, frontrunning, and censorship while preserving incentive alignment.

2. **Variable Input Amount in Intents:** In off-chain RFQ workflows, the user and solver agree on a fixed input amount before the solver commits a quote. However, if the fromAmount is variable or finalized only at execution time, as often desired in cross-chain and composable DeFi contexts, the solver's original quote becomes unreliable or stale. Users need a way to have their intent fulfilled on-chain *without* agreeing to a fixed price up front.

3. **Long Expiry Times:** Cross-chain RFQ typically involves an *expiry time* by which funds must be locked on the source chain to keep the solver's quote valid. This constrains usage to situations where the user can submit a transaction quickly. In more complex workflows (e.g., using CowSwap batch auctions or fiat on-ramps prior to a cross-chain step), users cannot always finalize on the solver's schedule. Additionally, a long expiry gives the user a "free option" on the solver's quote for the duration, forcing solvers to quote more conservatively. By contrast, our auction method produces a final price only for the specific order at the time of bidding, allowing solvers to quote more aggressively.

Our Contributions. This paper addresses the above challenges by presenting the following:

1. **A Leaderless, Sealed-Bid Auction Mechanism for Intents**: We propose a decentralized protocol in which no single actor controls price discovery. Bids are encrypted and revealed only after the bidding period ends, preventing frontrunning, shilling, and censorship.

2. **Threshold IBE for Decentralized Decryption**: We integrate a scalable MPC scheme with minimal overhead— *threshold Identity-Based Encryption* (IBE)— to ensure that the power to decrypt sealed bids is distributed among multiple validators, removing the need for a centralized auctioneer or offchain price discovery mechanism.

3. **Intent Confidentiality Enhancement with FHE and TEE**: While threshold IBE protects bids and can also offer pre-execution or conditional confidentiality for encrypted intents, additional cryptographic techniques such as *Fully Homomorphic Encryption* (FHE) and *Trusted Execution Environments* (TEE) can be leveraged for full confidentiality, given confidentiality and performance trade-offs.

4. **Efficient, Fair, and Secure Price Discovery**: Our approach is designed to be fast enough for time-sensitive trades, fosters open participation, and aligns incentives for both solvers and users.

The rest of the paper proceeds as follows. Section 2 surveys related work, focusing on credible auctions and shill-proofness. Section 3 introduces key cryptographic primitives, including threshold IBE, threshold FHE, and TEEs. Section 4 describes our proposed approach in detail, structuring the methodology for sealed-bid auctions in DeFi intent protocols. Section 5 explores variations of the protocol, extended privacy mechanisms, and trade-offs in real-world conditions. Section 6 points to other impactful applications of leaderless auctions in DeFi. Section 7 discusses implementation details and evaluates its performance for real-world setups. Finally, Section 8 concludes.

2 Related Work

2.1 Credible Auctions and Cryptography

An auction is considered credible when a revenue-maximizing auctioneer has no motivation to misrepresent information about other bidders [3, 22]. Akbarpour and Li [3] established that it is impossible to simultaneously achieve credibility, truthfulness, and bounded communication complexity in auctions. However, this trilemma can be circumvented by assuming that both the auctioneer and buyers are computationally bounded, meaning they cannot violate established cryptographic assumptions. Building on this, Ferreira and Weinberg [18] showed that cryptographic auction mechanisms can achieve all three properties. Their approach requires buyers to submit cryptographic commitments of their bids along with collateral to the auctioneer. The cryptographic commitments prevent the auctioneer from accessing bid information without breaking cryptographic protocols, while the collateral disincentivizes the introduction of fake bids, as doing so would incur a cost [12]. In this work, we utilize threshold identity-based encryption (IBE) to attain the same credibility outcomes without depending on any centralized entities to oversee cryptographic commitments. By decentralizing the decryption and execution processes, the auction's security and integrity are preserved, while minimizing reliance on a single authority. This approach is therefore termed a *Leaderless Auction* [23]. Additionally, as highlighted by Chitra et al [12], our application of blockchain technology strengthens the credibility results of [18] by introducing two critical features that restrict the potential for self-interested behavior by the auctioneer. First, smart contracts enable the concept of credibility to be extended to scenarios where the auctioneer lacks any prior reputation, thereby overcoming a key limitation in the credibility definition proposed by Akbarpour and Li [3]. Second, the use of blockchain allows mechanisms to be executed over a public broadcast channel, preventing the adaptive, undetectable deviations that lead to the negative outcomes described by Ferreira and Weinberg [18].

2.2 Negative Results on Shill-Proof Auctions

Komo, Kominers, and Roughgarden [22] define weak shill-proofness as auction formats in which sellers cannot expect to profit from submitting shill bids in equilibrium, and strong shill-proofness as formats where, even with full knowledge of bidders' valuations, sellers have no incentive to engage in shill bidding. The authors establish that the Dutch auction with an appropriate reserve price is the unique auction format that satisfies both revenue optimality and strong shill-proofness under their model's assumptions. A key assumption in their framework is that shill bidders have access to information about previous bids, allowing them to influence the auction's outcome. However, this assumption does not hold in sealed-bid auctions, where bids are encrypted and remain hidden from both other bidders and the seller. As a result, the conclusions drawn in the paper regarding the uniqueness of Dutch auctions as shill-proof do not extend to sealed-bid formats. By relaxing the assumption of bid transparency, sealed-bid auctions can effectively achieve shill-proofness, indicating that the scope of shill-proof auction formats is broader.

2.3 Dutch Auctions and MEV

Dutch auctions [36] are susceptible to frontrunning, as bidders can wait for the first bid to appear in the mempool and outpace it either by acting faster or by tipping the block

proposer. These auctions can be implemented off-chain in a centralized manner, but this requires trusting a centralized auctioneer. Alternatively, decentralized implementations remain vulnerable to frontrunning and lack a predetermined finalization time. In contrast, sealed-bid auctions conclude at a fixed time, and bidders are not incentivized to delay participation or engage in manipulative tactics. Our sealed-bid approach addresses both issues by eliminating on-chain partial price revelation and avoiding single points of trust.

2.4 Other Cryptographic Approaches

Other cryptographic schemes: Recent works [35] have applied cryptographic primitives like Verifiable Delay Functions (VDFs) [7], timelock puzzles, and Zero-Knowledge Proofs (ZKPs) to decentralized sealed-bid auctions. However, these protocols are slow and costly in decentralized finance (DeFi) settings, as decryption spans several blocks, introducing latency and creating a trade-off between security and performance. Alternatively, other projects such as Zama [41] suggest using threshold Fully Homomorphic Encryption (FHE) or other Multi-Party Computation (MPC) schemes including SPDZ [21]. While secure, these methods require additional ZKPs for input and computation verification, increasing computational overhead and latency, making them impractical for DeFi. Furthermore, FHE-based projects like Fhenix [19], which utilize Zama's TFHE, depend on centralized and off-chain co-processors to manage the high computational costs. Threshold Identity-Based Encryption (IBE) offers a fully decentralized solution, providing similar security guarantees with significantly less computational cost and minimal overhead.

3 Background

This section details three cryptographic building blocks and then formalizes the notion of *intents* in DeFi.

3.1 Threshold Identity-Based Encryption (IBE)

Identity-Based Encryption (IBE) [8, 25] is a public key encryption mechanism where the *public key* can be any valid identifier, such as an email or a user-chosen string. In threshold IBE:

1. The system runs a **Setup** algorithm to produce a *master public key* (MPK) and a set of *master secret key shares* held by different validators.

2. An **Encrypt** algorithm, Encrypt(MPK, ID, m), takes a message m and encrypts it using ID and the publicly known MPK.

3. An **Extract** operation requires a threshold of these validators to combine their shares to produce a *private key* corresponding to a particular identity ID.

4. A **Decrypt** operation uses the private key generated for ID to recover the message.

Mathematically, let n be the total number of validators, and t be the threshold such that $t \leq n$. Each validator i holds a share sk_i of the master secret key. The *Extract* phase can be viewed as:

$$\mathrm{sk}_{ID} \leftarrow \mathsf{Aggregate}\big(\mathrm{sk}_{i_1}, \mathrm{sk}_{i_2}, \ldots, \mathrm{sk}_{i_t}\big),$$

where $\{i_1, \ldots, i_t\} \subseteq \{1, \ldots, n\}$. The resulting sk_{ID} is then used for $\mathrm{Decrypt}(\mathrm{CT}, \mathrm{sk}_{ID})$ on the ciphertext CT. This threshold approach ensures no single validator can decrypt alone, mitigating centralization risks.

3.2 Threshold Fully Homomorphic Encryption (FHE)

Fully Homomorphic Encryption (FHE) [20, 11] allows computation on encrypted data without the need for decryption. In a threshold FHE system, the secret decryption key is also shared among n parties. Formally, each party i holds a share fhk_i of the secret key. A ciphertext c can be transformed into c' under some function f (all while c remains encrypted), and eventually decrypted if at least t parties collaborate:

$$\mathrm{Dec}\big(\mathsf{Aggregate}(\mathrm{fhk}_{i_1}, \ldots, \mathrm{fhk}_{i_t}),\, c'\big) = f(\mathrm{Dec}(c)).$$

Although threshold FHE provides robust privacy, it is computationally expensive for high-throughput DeFi use cases. We explore partial integration of threshold FHE in Section 5 as a supplementary privacy mechanism.

3.3 Trusted Execution Environments (TEE)

A TEE, such as Intel SGX [40] or ARM TrustZone [5], is a hardware enclave that isolates sensitive computations from the rest of the system. When used in auctions, the TEE can decrypt bids internally, compute the winning price or clearing price, and release only the necessary outputs on-chain, all while keeping sensitive data hidden. While TEEs can be powerful, they introduce trust in hardware manufacturers and potential vulnerabilities [37, 38, 27, 26], if the enclave is compromised. Hence, we suggest TEEs as an optional augmentation to threshold IBE rather than a sole solution.

3.4 Intents

An *intent* $\mathcal{I}$ can be defined as:

$$\mathcal{I} = \Big(\mathrm{srcChain}, \mathrm{dstChain}, \mathrm{srcToken}, \mathrm{dstToken}, \mathrm{amountIn}, \mathrm{minAmountOut}, \mathrm{deadline}\Big).$$

The user broadcasts $\mathcal{I}$ or shares it with potential solvers. The parameter minAmountOut ensures the user receives at least this quantity of destToken, while deadline sets a time limit for the swap. In this work, for the sake of completeness, we outline a general-purpose intent protocol. Single-chain protocols like CoWSwap [15] are specific cases where the source and destination chains are the same. For EVM-based single-chain protocols, locking funds on the source chain is unnecessary, as transactions can be signed off-chain and executed later together with the solver's transaction. This adjustment improves user experience but does not alter the general flow of our protocol.

4 Methodology

In this section, we detail our proposed sealed-bid auction methodology for intent-based DeFi protocols. To elucidate the design, we first contrast how *normal* existing intent protocols function and then introduce our *Leaderless Auction* approach. We further break down the methodology into key steps and elaborate on the roles of each party. We also discuss how our model circumvents the weaknesses outlined in Section 1.

4.1 Overview of Existing Intent Protocols

Current intent protocols often follow this flow:

Step 1: User Declares Intent. A user announces $\mathcal{I}$, via an app front end or direct API integration.

Step 2: Solvers Provide Quotes. A single or centralized set of solvers offers quotes, often privately or on a server.

Step 3: User Accepts a Quote. The user accepts the quote, signs a transaction, and sends assets to an escrow or directly to the solver.

Step 4: Solver Fills Intent. The solver handles the swap, returning the output tokens to the user.

Step 5: Settlement Occurs. Cross-chain verification of the fill and release of escrowed tokens to the solver and fees.

Although this design is simple, it remains centralized, its price discovery mechanism is neither dynamic nor optimized, and it lacks confidentiality because other parties can observe both intents and bids.

4.2 Proposed Sealed-Bid Auction Model

To address these shortcomings, we introduce a *sealed-bid* model. The main components are:

1. A **Bid Escrow** contract or on-chain component where user assets are locked.
2. A **Decentralized Auction** mechanism powered by threshold IBE for sealed bids.
3. A **Settlement** phase that unlocks user assets only upon proof of successful execution.

Crucially, no single solver or aggregator can see other participants' bids until the auction ends, mitigating frontrunning and manipulative strategies. As soon as the lock is finalized, an auctionID is generated, and solvers can compete for the best price within the allotted bidding window.

4.3 Auction Logic and Bid Encryption

Let auctionID uniquely identify the user's intent, which includes the lock of funds on the source chain. Under threshold IBE, each solver j computes:

$$\mathrm{CT}_j = \mathrm{Encrypt}(\mathrm{MPK}, \mathrm{auctionID}, m_j),$$

where m_j encodes the solver's proposed exchange rate or other relevant bid parameters. These ciphertexts are submitted on-chain or to a public mempool for finalization. Because they are encrypted, other solvers cannot glean each other's prices.

While frequent auctions are conducted for each intent or for batches of intents (to optimize efficiency), it is important to note that the decentralized key generation for

validators occurs only infrequently–specifically, when the validator set requires modification. Moreover, the extraction of private key shares for each auctionID does not reveal any information about the secret key shares of the individual validators, thereby ensuring that private keys for subsequent auctions can be securely derived.

Furthermore, compared to alternative threshold decryption methods–where each validator would need to send a partial decryption for every bid, resulting in an $\mathcal{O}(n^2)$ message complexity, or alternatively reconstruct the master secret key, incurring an $\mathcal{O}(n^3)$ complexity, the proposed threshold identity-based encryption (IBE) approach incurs only a minimal overhead of $\mathcal{O}(n)$ where n is the number of validators. This efficiency is achieved because each validator requires only one private key share, and a single private key is used for all bids in each auction.

4.4 Auction Phases

We break down the proposed methodology into four phases for clarity.

4.4.1 Phase 1: Intent Expression

The user broadcasts the intent $\mathcal{I}$, then locks the relevant amount of srcToken in a contract on the source chain. This lock event triggers the creation of a new auctionID.

By placing funds into an immutable on-chain contract, the user ensures that all solvers can trust the funds are available. The user's locked assets act as collateral that the solver eventually redeems upon successful execution.

4.4.2 Phase 2: Bidding Window

A short but fixed bidding window (e.g., a few seconds) is open for *all* solvers. Each solver observes the auctionID, prepares a bid m_j (often a ratio specifying how many destToken per srcToken they offer), and encrypts it:

$$\mathrm{CT}_j = \mathrm{Encrypt}(\mathrm{MPK},\ \mathrm{auctionID},\ m_j).$$

They broadcast CT_j on-chain before the bidding deadline. No partial reveals are possible at this stage, ensuring sealed-bid properties. Note that if the user's input amount is variable or uncertain, solvers simply factor the possible range into their final offered rate or incorporate any partial fill logic. Although establishing an initial auction price is not strictly necessary, introducing a minimum threshold e.g. quote and slippage enhances the user experience and guides bidding behavior by providing a clear baseline for both participants and solvers. Notably, it is unnecessary to set this minimum price on-chain, as solvers can be informed via the existing intent protocol mechanisms.

4.4.3 Phase 3: Decentralized Decryption

After the bidding window closes, a threshold t of validators run the $\mathsf{Extract}$ operation for auctionID. This produces a decryption key $\mathrm{sk}_{\mathrm{auctionID}}$ used to open all ciphertexts:

$$m_j = \mathrm{Decrypt}(\mathrm{CT}_j,\ \mathrm{sk}_{\mathrm{auctionID}}).$$

The bids are revealed collectively on-chain. Next, invalid bids should be filtered and malicious bidders can be further slashed. The best bid, say $m_{j^\star}$, is chosen e.g. highest

exchange rate or the best price for the user. A summary of winning bid information is posted, along with a proof or direct listing of all bids for transparency.

Note that while the highest bidder wins the auction, the execution price can be determined using different auction formats: it may correspond to the highest bid in a first-price sealed-bid auction, the second-highest bid in a second-price sealed-bid auction (Vickrey auction) [6], or a hybrid approach combining elements of both.

4.4.4 Phase 4: Settlement and Asset Release

The winning solver $j^\star$ proceeds to fill the order on the destination chain. Once evidence of a successful fill is relayed back on-chain (e.g., via a cross-chain verification protocol or relaying mechanism), the locked source-chain funds are released to the solver. The user receives the destToken on the destination chain, and the transaction is considered complete.

4.5 Multi-Winner Auctions and Partial Fills

Large user requests may need multiple solvers to fill. In that case, the system can allow *multi-winner auctions*, awarding partial fills to multiple top bidders, each receiving a fraction of the user's locked funds commensurate with their bid. This approach can improve liquidity and competitiveness for large trades.

4.6 Auction Termination and Time Constraints

To function effectively in DeFi, an auction must close quickly. Latency of more than a few seconds can make the final price stale in volatile markets. Our model sets a maximum auction window (e.g., 2–5 seconds). The threshold validators then produce the decryption key within another short interval (e.g., 1–2 seconds). We discuss the trade-offs in Section 5.

5 Intent Confidentiality

The described sealed-bid auction in the previous section (Method 1) is open to any solver and all intents are publicly transparent to any party. We propose a progressive design for sealed-bid auctions for partial or full confidential of intents. This approach incrementally enhances privacy from a baseline of public user intent to full confidentiality using IBE, FHE/MPC, or TEE.

5.1 Method 2: Encrypting User Intents for Access Control and Conditional Decryption

Publicly disclosing sensitive intent information such as amount, slippage, liquidation or stop loss triggers exposes users to malicious actors. Encrypting user intent with threshold IBE ensures that only qualified solvers can access this information. Users encrypt their intents using Identity-Based Encryption (IBE) with the IDs of a selected group of qualified solvers. Validators generate the private key for these solvers very infrequently, typically when the solver set is updated based on factors such as reputation scores, subscription status, or other relevant metrics. Moreover, IBE can also enable conditional decryption of intents to further achieve programmable and pre-execution confidentiality i.e. sensitive

intent information can be encrypted toward market conditions such prices, time, ZKPs or solvers' commitments. In this case encrypted intents will be only decrypted once those conditions are met [25].

Once a solver decrypts the user's intent, they submit an encrypted bid as in Method 1. This creates on-chain sealed-bid auctions where both user orders and solver bids remain private from the general public.

Method 2 will result in:

- **Better execution quality for end users:** In the presence of costly effort and congestion in the intent markets, recent works counter-intuitively show that a planner aiming to maximize user welfare may prefer to restrict entry, resulting in limited oligopoly [13].
- **Monetization and Competition:** It could be powerful to sell the decryption keys to a set of competing solvers to a. monetize b. promote higher quality solvers e.g. through auctioning off the order flow for the highest price. The auction, subscription charges, or choosing solvers based on their scores or quality does not need to happen in every block and can be run infrequently in advance e.g. every week.
- **Confidentiality:** Since intents are encrypted, intents will not be leaked publicly to competitors, bots, or other potential malicious parties. We can achieve a web2 level of privacy by encrypting the intents and only sharing the decryption keys with solvers, preventing frontrunning and other exploitative techniques from any other party other than the solver. In the case of conditionally encrypted intents, frontrunning is restricted even for the solver itself since they need to commit to the execution based on partially encrypted intent.

By exclusively utilizing Threshold IBE and steering clear of Zero-Knowledge proofs (ZK) [31], Fully Homomorphic Encryption (FHE), or other resource-intensive cryptographic techniques, this approach ensures rapid processing and minimal overheads.

5.2 Method 3: Full Confidentiality with Threshold FHE/MPC or TEE

If users prefer not to disclose any trade information to solvers, MPC scehemes such as threshold FHE [11] or SPDZ [21] facilitates computations on encrypted data. Solvers can homomorphically compute functions over the encrypted intent to produce their bids. Threshold decryption at reveal recovers winning bid. Even further, the winning solver itself can further execute a private transfer for the intent execution. TEEs offer an alternative with reduced cryptographic complexity but introduce hardware trust assumptions.

5.3 Trade-Offs:

- **Performance and Trust vs Confidentiality Level:** MPC schemes such as Threshold FHE or SPDZ, and TEE implementations are highly resource-intensive, introducing delays, bandwidth overhead, and larger ciphertext sizes while also imposing new trust assumptions. These complexities can lead to 10x to 100x higher costs or necessitate specialized hardware. Additionally, they often depend on ZKPs, centralized parties, or optimistic approaches to validate inputs and computations over encrypted data, further amplifying performance overhead compared to IBE.

However, compared to IBE, these approaches offer significantly stronger confidentiality guarantees after execution.

- **Optimization vs Confidentiality:** Optimization and confidentiality of intents involves a delicate trade-off, while partial or full encryption of intents safeguards against frontrunning and other malicious exploits, it simultaneously constrains solvers' risk appetite and hampers their capacity for precise, fine-tuned optimizations.
- **Competition vs Cost:** An open auction fosters a fully decentralized and competitive market, reducing monopolies, exploitative pricing, and insider trading. However, the higher cost and complexity of managing a large open market can lead to inefficiencies, malicious activities, and lower execution quality. Therefore, it is crucial to curate a high-quality set of solvers, maintaining a balance between competition and cost. This presents another delicate trade-off between: a) Allowing a monopoly where only a few solvers dominate the system, and b) Managing solver costs and quality through mechanisms like parameterization and control systems.

5.3.1 Optimizing User Welfare

By considering all these parameters, we formally define the problem as an optimization model, where the objective function seeks to maximize user welfare $U(x)$, subject to the inherent trade-offs among performance, trust, confidentiality, optimization, and competition.

$$\max_{x} \quad U(x) = f(P(x), T(x), C(x), O(x), K(x), S(x)) \tag{1}$$

where:

- $P(x)$ - Performance efficiency (latency, bandwidth, ciphertext size)
- $T(x)$ - Trust assumptions (degree of decentralization, reliance on centralized parties)
- $C(x)$ - Confidentiality level
- $O(x)$ - Optimization quality (solver efficiency, cost-effectiveness)
- $K(x)$ - Competition level (number of active solvers, decentralization factor)
- $S(x)$ - Solver cost (computational and resource requirements)

subject to the following trade-offs:

- Performance and Trust vs Confidentiality

$$P(x) + T(x) \leq \lambda_1 C(x) \tag{2}$$

  A higher confidentiality level $C(x)$ typically comes at the cost of performance $P(x)$ and trust decentralization $T(x)$, where λ_1 controls the balance.

- Optimization vs Confidentiality:

$$O(x) \leq \lambda_2 C(x) \tag{3}$$

  Improving optimization efficiency $O(x)$ can conflict with confidentiality $C(x)$, requiring a balance controlled by λ_2.

– Competition vs Cost:

$$K(x) \geq \lambda_3 S(x) \tag{4}$$

A higher competition level $K(x)$ helps prevent monopolization but can increase solver costs $S(x)$, where λ_3 regulates this trade-off.

Method	**User Intent Visibility**	**Overhead**	**Confidentiality**
Public + Enc. Bids	Public	Low	Only Bids
Encrypted Intents	Qualified Solver	Low	Bids and Partially Intents
Threshold FHE/TEE	Private	High	Bids and Intents

Table 1: Comparison of methodologies, balancing overhead and privacy.

In practice, our baseline recommendation is the **IBE-Encrypted Intents + Enc. Bids** option, given its low overhead and clear confidentiality and execution quality benefits. If further intent confidentiality is required, FHE encryption (or other MPC schemes) of the intent or a TEE-based solver can be leveraged.

5.4 Challenges

(1) Fairness and Decentralization Our sealed-bid design removes the reliance on a centralized actor for price discovery, forcing all solvers to compete on price rather than on mempool manipulation or speed. Threshold IBE ensures no single party can decrypt or alter bids prematurely.

(2) Variable Input Amounts A user can declare a variable input (or fromAmount range) in the locked contract. Solvers bid on the condition that the final fill must lie within the declared range. Since the auction is sealed-bid, solvers are not discouraged by uncertain input amounts; they simply price accordingly and lock in if they win.

(3) Long Expiry Issue By making the auction short-lived and triggered only when the user locks funds, we avoid forcing solvers to stand by a quote for extended periods. Solvers quote only for a *specific* user order within a brief window, removing the "free option" problem and permitting tighter pricing.

5.5 Potential Vulnerabilities

Collusion of Validators: If t out of n validators collude, they could potentially reveal partial bid information before the official decryption. It is worth mentioning that the consequence of such an attack is losing confidentiality, and neither the safety of the network, loss of funds nor private information regarding the identities of users. In this case, the system's confidentiality will downgrade to the current state of public blockchains, and the execution quality will be lower.

However, validators and operators should be incentivized to protect the bid with respect to the stakes in the game. The solution lies in building robust networks where compliance can be enforced without compromising decentralization. The transition will involve integrating permissionless compliance mechanisms, where incentives are aligned to encourage honest validator behavior. Approaches like Proof of Stake (PoS) and Active

Validator Set (AVS) [32] ensure network economic security, while cryptographic traitor tracing [9] and slashing mechanisms deter malicious actors.

An interesting enhancement is the integration of multi-party computation (MPC) including threshold IBE and FHE with trusted execution environments (TEEs). MPC distributes trust among multiple parties, while TEEs ensure that no single party can access their individual shares. This hybrid approach significantly strengthens security, making the protocol practically unbreakable. A malicious actor would need to compromise the TEE of at least t validators, potentially using different TEE technologies, which adds an additional layer of security-in-depth [25, 24].

Replay Attacks and Double-Fills: The protocol must ensure that once user funds are released, the auction is permanently closed. Otherwise, an adversary might attempt to reuse the same on-chain event to trigger another auction. Proper indexing of auctionID and verifying state transitions on-chain mitigate this.

Invalid Bids: A solver could submit a very high sealed bid to win, only to fail at actual settlement. A slashing or deposit mechanism can address this: each bid must be accompanied by collateral that is forfeited if settlement is not completed promptly.

6 Auctions in Decentralized Finance

In this work, we designed and implemented leaderless sealed-bid auctions for a general-purpose intent protocol. This auction mechanism can also be adapted for various applications in decentralized finance (DeFi), including the following impactful use cases:

6.1 Fair and Simple Launch

Token launches in DeFi often rely on the bonding curve method, which is frequently manipulated, leading to a poor user experience and inefficient price discovery. Auctions help mitigate the impact of sniper bots and bad actors by aligning incentives. Both sealed-bid auctions and Dutch auctions align incentives by encouraging bidders to submit bids that reflect their true valuation of the asset. However, existing Dutch auctions such as Liquidity Bootstrapping Pools (LBPs) as seen in Fjord [33], pose significant challenges for users:

1. In practice, many users struggle to understand the logic behind Dutch auctions' decreasing prices, making them hesitant to participate.
2. The bidding period in Dutch auctions is unpredictable and can take an extended time to complete, leading to low engagement and suboptimal price discovery for sellers.
3. The winning bid in an onchain Dutch auction can be frontrunned.

This work proposes a **Leaderless Sealed-Bid Auction**, which:

– Prevents malicious strategies like sniping and frontrunning.

– Fosters competitive price-discovery through incentive-alignment

– Provides a simple and intuitive user experience.

– Ensures full transparency by operating entirely on-chain, without reliance on centralized parties.

6.2 Real-World Assets and Non-Fungible Tokens

Selling a Non-Fungible Token (NFT) through an open first-price auction is a widely used method. Additionally, Real-World Assets (RWAs) [28] such as real estate, carbon offset credits, and power grid allocations, and finance have been gaining popularity in the blockchain industry. However, implementing these auctions on-chain presents several challenges:

1. First-price auctions are vulnerable to malicious tactics like bid shilling, where the auctioneer or the artist inflates bids to maximize revenue.

2. NFT auction platforms, such as Stargaze Auctions [30], often extend the auction window by a fixed time after the last bid to prevent frontrunning. This leads to a poor user experience and an unpredictable auction duration.

3. The winning bid in an on-chain auction can be frontrun, compromising fairness.

The auction proposed in this work enables fair price discovery and improves the user experience for NFT launches. Additionally, it creates opportunities to implement real-world asset auctions on blockchain networks.

6.3 Lending Market Auctions

Auction-based mechanisms offer a robust framework for efficient and equitable price discovery in lending markets [34]. Unlike traditional models that rely on fixed or continuously fluctuating interest rates, a sealed-bid auction system allows borrowers to submit bids representing the maximum interest rate they are willing to pay, while lenders place offers reflecting their minimum acceptable rate. The system subsequently determines a single, market-clearing interest rate, ensuring all participants transact at a uniform and transparent rate.

The introduction of an auction-based approach in lending markets provides several advantages:

– **Fair and Transparent Pricing:** The auction mechanism ensures that all participants, regardless of size or influence, transact at the same interest rate, eliminating preferential treatment and enhancing fairness.

– **Efficient Price Discovery:** By aggregating supply and demand in a competitive environment, the auction determines an equilibrium interest rate that accurately reflects market conditions, improving capital allocation.

– **Enhanced Liquidity:** Periodic batching of orders allows for improved liquidity by matching borrowers and lenders more effectively while reducing slippage and transaction costs.

– **Predictable Capital Deployment:** Borrowers and lenders benefit from greater predictability in loan terms and interest rates, mitigating uncertainty and fostering long-term planning.

– **Decentralization:** By reducing reliance on intermediaries, auction-based lending mechanisms promote a more trustless and inclusive financial ecosystem.

The implementation of auction mechanisms in lending markets represents a significant advancement in financial market structure. By enhancing fairness, improving price discovery, increasing liquidity, and reducing reliance on intermediaries, this model offers a viable alternative to traditional lending frameworks. Future research may explore optimization techniques and hybrid models that further refine auction-based lending systems.

7 Implementation and Performance

To evaluate the performance of our sealed-bid auction, we utilize an Apple M1 Max and conduct 100 experimental runs with a real-world decentralized test network. Our findings indicate that the validator set can scale to 185 validators. We opt for 10 validators because this number sufficiently represents real-world scenarios, while larger validator sets pose additional operational complexities and cost for our experiments. Table 2 presents the average execution times along with 95% two-sided confidence intervals.

Our results confirm the feasibility of this approach using basic hardware resources, even for high-performance proof-of-stake (PoS) blockchains. For instance, the average *block key extraction* time (consisting of block key share aggregation, verification, and block key computation) for 10 validators is 126.39 ms, which is substantially below the block time of most mainstream blockchains, such as Ethereum [39] (∼12 s) or Cosmos SDK-based chains [14] (∼1–5 s).

We have implemented this work within the open-source *FairyRing*[1] repository, which leverages ABCI++ vote extensions [1] to:

1. Submit private key shares in `BeginBlock`,

2. Finalize auctions in `EndBlock`, thus ensuring one-block finality and guaranteeing successful execution.

In our deployment, we observe an average block time of 1.58 ± 0.1 seconds over 6,056,499 blocks, allowing auctions to conclude within this interval. This speed is well suited for intent-matching applications and is significantly faster than required bidding window in most existing DeFi auctions, including token launches, NFT auctions, and lending markets.

We additionally measure encryption and decryption times for random 256-byte messages across up to 1000 bids. On average, decryption takes 1.1 ms while encryption requires 6.8 ms, both of which are negligible compared to block time. Since each bid is independently encrypted, decryption of bids can be also parallelized. To optimize bandwidth overhead and runtime, we employ a hybrid encryption scheme [17], using ChaCha20 [10] for the bid contents and identity-based encryption for the corresponding symmetric key.

[1] `https://github.com/Fairblock/`

Table 2: Mean values of execution time for different phases of the auction.

Auction Settlement (ms)	Key Extraction (ms)	Decryption (ms)	Encryption (ms)
0.26 ± 0.01	126.45 ± 0.14	1.15 ± 0.02	6.88 ± 0.03

8 Conclusion

In this work, we introduced a decentralized, sealed-bid auction mechanism that addresses core weaknesses in existing intent-based protocols. By removing reliance on centralized auctioneers for price discovery, we reduce the risk of censorship and manipulation. Our design strikes a pragmatic balance between cryptographic rigor and the latency constraints of modern DeFi. By ensuring fairness, privacy, and efficiency, it serves as a robust platform upon which future intent protocols can be built, ultimately promoting a more efficient, transparent and user-centric financial ecosystem.

Acknowledgments

We thank Tarun Chitra, Arbitrum Foundation, Anoma, Rova, Stargaze, and Plume for their invaluable feedback and support.

References

[1] Abci++ vote extensions. `https://docs.cosmos.network/main/build/abci/vote-extensions`. (Accessed on 02/01/2025).

[2] debrdige. `https://across.to/`. (Accessed on 04/01/2025).

[3] Mohammad Akbarpour and Shengwu Li. Credible auctions: A trilemma. *Econometrica*, 88(2):425–467, 2020.

[4] Anoma. `https://anoma.net/`. (Accessed on 12/03/2021).

[5] Arm trustzone. `https://www.arm.com/technologies/trustzone-for-cortex-m`. (Accessed on 04/01/2025).

[6] Lawrence M Ausubel et al. A generalized vickrey auction. *Econo0 metrica*, 1999.

[7] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In *Annual international cryptology conference*, pages 757–788. Springer, 2018.

[8] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001.

[9] Dan Boneh, Aditi Partap, and Lior Rotem. Accountability for misbehavior in threshold decryption via threshold traitor tracing. In *Annual International Cryptology Conference*, pages 317–351. Springer, 2024.

[10] Chacha20 and poly1305 for ietf protocols. `https://www.rfc-editor.org/rfc/rfc7539.txt`. (Accessed on 04/03/2022).

[11] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23*, pages 409–437. Springer, 2017.

[12] Tarun Chitra, Matheus VX Ferreira, and Kshitij Kulkarni. Credible, optimal auctions via public broadcast. In *6th Conference on Advances in Financial Technologies (AFT 2024)*, pages 19–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.

[13] Tarun Chitra, Kshitij Kulkarni, Mallesh Pai, and Theo Diamandis. An analysis of intent-based markets. *arXiv preprint arXiv:2403.02525*, 2024.

[14] Cosmos sdk. `https://docs.cosmos.network/`. (Accessed on 03/26/2022).

[15] Cowswap. `https://https://cow.fi//`. (Accessed on 11/08/2024).

[16] debrdige. `https://debridge.finance/`. (Accessed on 04/01/2025).

[17] Pooja Dixit, Avadhesh Kumar Gupta, Munesh Chandra Trivedi, and Virendra Kumar Yadav. Traditional and hybrid encryption techniques: a survey. In *Networking communication and data knowledge engineering*, pages 239–248. Springer, 2018.

[18] Matheus VX Ferreira and S Matthew Weinberg. Credible, truthful, and two-round (optimal) auctions via cryptographic commitments. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 683–712, 2020.

[19] Fhenix. `https://www.fhenix.io/`. (Accessed on 04/01/2025).

[20] Craig Gentry. *A fully homomorphic encryption scheme*. Stanford university, 2009.

[21] Marcel Keller. MP-SPDZ: A versatile framework for multi-party computation. Cryptology ePrint Archive, Paper 2020/521, 2020.

[22] Andrew Komo, Scott Duke Kominers, and Tim Roughgarden. Shill-proof auctions, 2024.

[23] Leaderless auction. `https://www.paradigm.xyz/2024/02/leaderless-auctions`. (Accessed on 04/01/2025).

[24] Peyman Momeni, Setareh Ghorshi, and Jose Maria Leong Chua Zheng. Multimodal mpc and tee. `https://hackmd.io/@Fairblock/rkSiU78TR`. (Accessed on 04/01/2025).

[25] Peyman Momeni, Sergey Gorbunov, and Bohan Zhang. Fairblock: Preventing blockchain front-running with minimal overheads. In *Security and Privacy in Communication Networks*, pages 250–271. Springer Nature Switzerland, 2023.

[26] Sgx root key. `https://securityonline.info/intel-sgx-security-compromised-root-provisioning-key-extracted/`. (Accessed on 04/01/2025).

[27] Sgx fail. `https://sgx.fail/`. (Accessed on 04/01/2025).

[28] Plume. https://plumenetwork.xyz/. (Accessed on 04/01/2025).

[29] Squid coral. https://www.squidrouter.com/. (Accessed on 04/01/2025).

[30] Stargaze auctions. https://www.stargaze.zone/auctions. (Accessed on 04/01/2025).

[31] Xiaoqiang Sun, F Richard Yu, Peng Zhang, Zhiwei Sun, Weixin Xie, and Xiang Peng. A survey on zero-knowledge proof in blockchain. *IEEE network*, 35(4):198–205, 2021.

[32] EigenLayer Team. Eigenlayer: The restaking collective. *White paper*, pages 1–19, 2024.

[33] Fjord. https://app.fjordfoundry.com/token-sales. (Accessed on 04/01/2025).

[34] Term finance. https://www.term.finance/. (Accessed on 04/01/2025).

[35] Nirvan Tyagi, Arasu Arun, Cody Freitag, Riad Wahby, Joseph Bonneau, and David Mazières. Riggs: Decentralized sealed-bid auctions. Cryptology ePrint Archive, Paper 2023/1336, 2023.

[36] Uniswapx. https://docs.uniswap.org/contracts/uniswapx/overview. (Accessed on 04/01/2025).

[37] Jo Van Bulck, Marina Minkin, Ofir Weisse, Daniel Genkin, Baris Kasikci, Frank Piessens, Mark Silberstein, Thomas F Wenisch, Yuval Yarom, and Raoul Strackx. Foreshadow: Extracting the keys to the Intel SGX kingdom with transient out-of-order execution. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 991–1008, 2018.

[38] Stephan van Schaik, Andrew Kwong, Daniel Genkin, and Yuval Yarom. SGAxe: How SGX fails in practice, 2020.

[39] Gavin Wood. Ethereum: A secure decentralized generalized transaction ledger, 2014.

[40] Bin Cedric Xing, Mark Shanahan, and Rebekah Leslie-Hurd. Intel software guard extensions (Intel SGX) software support for dynamic memory allocation inside an enclave. In *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016*, New York, NY, USA, 2016. Association for Computing Machinery.

[41] Zama. https://zama.ai. (Accessed on 04/01/2025).