Strengthening Security of Blockchain Dutch Auctions

Nathanael Axel Wibisono* §, Jun Kai Koh† §, Zong Yu Lee‡ §
School of Computer Science and Engineering, Nanyang Technological University, Singapore
*nath0016@e.ntu.edu.sg, †c200146@e.ntu.edu.sg, ‡ZLEE043@e.ntu.edu.sg
§ These authors contribute equally to the work.

***Abstract*—This paper examines the implementation of Dutch auctions on blockchain platforms. We weigh the relative importance of critical metrics, security, privacy and scalability, ultimately focusing on security. Dutch auctions, characterized by a decreasing price model, are especially relevant for Initial Coin Offerings (ICOs), as they efficiently match supply with demand. The study explores the security challenges posed by malicious actors, such as bidders and auctioneers, and the specific threat of front-running attacks. To address these issues, it proposes a pull-based refund process and self-enforcing smart contracts to enhance security and fairness. Additionally, it introduces encrypted pre-orders as a novel solution to mitigate front-running, ensuring a more equitable and efficient auction process. The paper also discusses the limitations of these solutions, particularly in computational efficiency and gas costs, suggesting further exploration into off-chain computation as a potential remedy. Overall, the paper highlights the need for robust security measures and innovative solutions to maintain the integrity of Dutch auctions in a blockchain environment.**

***Index Terms*—dutch auctions, blockchain technology, smart contracts, security, privacy and scalability in blockchain**

Terminology

- **Initial Coin Offerings (ICOs):** A fundraising method that involves selling cryptocurrencies.
- **Smart Contract:** A self-executing contract with the terms of the agreement between buyer and seller being directly written into lines of code.
- **Blockchain:** A decentralized, distributed ledger technology that records transactions across multiple computers.
- **Bidder:** An individual or entity that places a bid in an auction.
- **Buyer Position:** The number of bids placed by a bidder.
- **Seller / Auction Owner:** The entity that owns the items or tokens being auctioned.
- **Clearing Price:** The final price at which a product or token is sold in an auction.
- **Auctioneer:** The individual or entity that conducts the auction.
- **Reserve Price:** The minimum acceptable price set by the seller for the item being auctioned.
- **Initial Price:** The starting price of the item or token in an auction.
- **Front-running:** An unethical practice where someone exploits their knowledge of upcoming transactions to their advantage.
- **Gas Wars:** A situation in blockchain auctions where bidders significantly increase transaction fees to prioritize their bids.

I. Introduction

A. Dutch Auction

Dutch auctions present a distinctive contrast to the traditional English auction. In an English auction, the price is successively raised until only one bidder remains, who then purchases the item at the final bid price. In contrast, a Dutch auction operates on a decreasing price model, where the price is systematically lowered until a buyer is willing to make a purchase at the current price [1].

Dutch auctions are particularly advantageous for Initial Coin Offerings (ICOs) where multiple coins or tokens are available for sale, as its decreasing price mechanism efficiently matches supply with demand, ensuring that each coin or token is sold at market price. This is effective in establishing fair market value for each coin or token, especially in a scenario where the exact valuation is uncertain or fluctuating.

For a Dutch auction to commence, the auctioneer begins by declaring key parameters such as the auction's duration, reserve price, initial (starting) price, and the specific token address, by interacting with the smart contract constructor. The auctioneer then sets the allowance of the token to the auction, which acts as a cap on the number of tokens that can be distributed. Only after these two foundational steps can the auctioneer initiate the auction. This ensures transparency and fairness, as all parameters are predefined in the blockchain, visible and immutable to all participants.

In a Dutch auction, most auction parameters, including the quantity of unsold tokens, are openly accessible to all observers. This transparency allows bidders to adapt their strategies in response to the remaining token supply. For example, when the number of tokens available dwindles, bidders are more likely to purchase at a price exceeding their initially planned threshold.

Dutch auctions can conclude in one of two distinct ways. Firstly, the auction may end upon reaching its expiration time. At this juncture, the auction's price will have descended to its lowest point, known as the clearing price. Secondly, the auction can also end prematurely when all available coins or tokens have been claimed. This outcome transpires either when a final bidder purchases the remaining tokens at the current price or when the decreasing price mechanism triggers an automatic distribution of the remaining tokens among previous bidders who had set their purchase bids at higher prices. In this case, the auction concludes not due to time expiration, but due to the total allocation of the auctioned assets [1], [2].

B. Evaluating Security, Privacy and Scalability in Dutch Auction

1) Security: The foundational principle of a Dutch Auction is to maintain fairness and integrity in the bidding process. Security concerns, such as front-running, bid manipulation, or denial of service attacks, critically undermine this integrity, leading to a potential loss of trust among participants. This is particularly significant in a blockchain context, where transactions are irreversible and no centralised authority exists to resolve disputes. Therefore, ensuring the security and integrity of every transaction is paramount.

2) Privacy: Privacy in on-chain Dutch auctions is vital for several reasons. Firstly, in a transparent blockchain environment, if bid amounts and bidder identities are exposed, it can lead to strategic manipulation or collusion, undermining the fairness of the auction. Secondly, in the context of Initial Coin Offerings (ICOs) or token sales, privacy helps protect investors from potential targeted attacks or phishing scams, as their investment amounts and wallet addresses remain concealed.

3) Scalability: Scalability in on-chain Dutch auctions is crucial due to the potentially high volume of transactions as prices decrease. To address this, efficient smart contract design is essential to minimise the computational load on the network. Layer 2 solutions, like rollups or sidechains, offer a viable way to handle a high throughput of transactions while maintaining lower fees. These measures are key to ensuring that Dutch auctions on blockchain platforms are efficient, cost-effective, and can handle the increased demand without compromising on security or decentralisation.

4) Prioritising Security: When evaluating the critical attributes of blockchain technology in terms of Dutch Auction – privacy, scalability, and security – security stands out as the paramount priority. While privacy ensures the protection of user data and scalability addresses the network's ability to handle a growing number of transactions, these features hinge on the assurance of a secure environment. If a blockchain cannot guarantee security, its ability to protect user privacy and efficiently scale becomes moot.

Furthermore, in Dutch auctions, privacy takes a backseat given their inherent open-outcry nature. This shift towards greater information transparency benefits both auctioneers and bidders, resulting in heightened trading efficiency and broader participation. Consequently, the unique characteristics of Dutch auctions render privacy considerations less pertinent.

II. Background and Related Work

In the next section we present our research into security considerations of implementing an on-chain dutch auction, and describe the scenarios of how malicious actors may seek to exploit vulnerabilities in the dutch auction. We also drew inspiration from the implementations of sealed-bid auctions and how the strategies employed to ensure confidentiality can help to resolve the security issues of the on-chain dutch auction.

A. Malicious Bidders

AkuAuction, a widely-used smart contract for auctioning AKU Non-fungible Tokens (NFTs), had 34 Million USD frozen due to flaws in its dutch auction smart contract. One of the defects was the ”frozen refunds exploit” in the refund mechanism of the AkuAuction contract. [3] The process of refund in AkuAuction was as follows:

1) Identify the first bidder.
2) Issue a refund in ETH.
3) Verify the successful completion of the refund.
4) Proceed to the next bidder and repeat the process.

A malicious buyer can exploit this logic process by creating a smart contract that would always return a failure message when ETH was sent to it. As a result, the AkuAuction's refund logic continually attempted to resend the ETH, unable to bypass this malicious bidder. This attack effectively blocked subsequent bidders from receiving their refunds, as the contract was unable to move past the compromised bid.

B. Malicious Auctioneers

A malicious auctioneer can significantly disrupt the intended logic and fairness of the auction process. In such traditional auctions, the role of the auctioneer is crucial for ensuring a smooth and equitable transaction. However, a malicious actor in this position might engage in several detrimental actions. For instance, they could refuse to distribute the tokens to the winning bidders, thereby undermining the very essence of the auction. [4]

Additionally, they might withhold refunds that are due to participants who did not win but are entitled to get their funds back. Another disruptive tactic could be arbitrarily halting the auction midway, leaving the process incomplete and participants in limbo. These actions not only breach the trust inherent in the smart contract but also jeopardise the integrity and reliability of the auction mechanism, leading to potential losses and a diminished confidence in the system.

C. Front-running Attack

In a traditional off-chain dutch auction, the buyer balances between the certainty of winning and the price they pay to win the bid. Bidding earlier increases the probability of winning but also overpaying. On the other hand, the longer a bidder delays placing a bid, the lower the price he has to pay for the item. The best strategy is to bid the highest price a buyer is willing to pay for an item, taking into account the risk that waiting for a lower price could result in losing the item to another buyer. [5]

However, in on-chain dutch auctions, there is a delay between placing a bid and its registration on the blockchain, resulting in a change of optimal bidding strategy. Bidders may find it advantageous to monitor the blockchain for pending bids or transactions. If there are none, it could be beneficial to wait, potentially acquiring tokens at a lower price. Conversely, if there are several pending transactions, it becomes more prudent to secure the item at that price, as the risk of losing it by waiting for a further price drop is significantly higher. [6]

When a substantial number of pending transactions for a bidding token exceed the available supply, it can trigger a ”gas war” in an on-chain Dutch auction. In this scenario, the competition moves away from the auction price itself and focuses on the gas price – the fee paid for processing transactions on the blockchain. This shift essentially transforms the auction into an ascending-price format as the bid that is processed first, typically the one with the highest gas price, becomes the winner. This phenomenon can result in a decrease in the net auction revenue. In such scenarios, a significant portion of the money that could have been spent on the actual bid amount in the auction gets diverted towards gas costs.

The emergence of “gas wars” in on-chain auctions leads to a decrease in net auction revenue due to high gas costs. As such there is an emerging need for innovative solutions that replicate the efficiency of off-chain auctions in on-chain auctions.

D. Front-running Solution Case Study

In a sealed auction, bids are submitted confidentially and are not revealed until the end of the bidding period. This format inherently protects against frontrunning, a type of attack where an entity observes a bid and then quickly places another bid before the first one is processed, taking advantage of the information delay on the blockchain. By keeping bids concealed, a sealed auction ensures that no participant or observer can preemptively outbid others based on early access to bid information. This confidentiality not only maintains the integrity of the bidding process but also fosters a fair and competitive auction environment, which prevents frontline attack and “gas wars”. [7]

There are primarily two methods to implement a sealed auction: the simple commit-and-reveal method and using a protocol like LibSubmarine. The commit-and-reveal method involves bidders committing a hashed version of their bid, which is later revealed and verified, while LibSubmarine offers a more sophisticated approach to conceal transaction details temporarily.

By incorporating the concept of a sealed bid in Dutch auctions, we can significantly mitigate the risk of frontrunning attacks. Frontrunning typically occurs when potential bidders have access to information about pending transactions, leading to a “gas war”. In a “gas war”, participants aggressively increase their transaction fees to ensure their bids are prioritised, often resulting in a diminished net revenue for the auction. Sealed bids conceal the transaction details from other bidders, creating an environment where each bid remains private until the auction concludes. This lack of visibility into other bids prevents bidders from engaging in frontrunning, as they cannot anticipate or react to the bids of others.

III. Proposed Solution

A. Pull-Based Refund Process

To minimise the risk of possible exploits that might arise from malicious bidders' refund process, a better practice is to require users to initiate the withdrawal of their token winnings and refunds through a withdraw() function, instead of processing refunds in a sequential order. This approach ensures the refund mechanism remains unaffected by malicious buyers, as it isolates the impact of any compromised refund logic to the individual bidder involved, rather than halting the entire refund process.

However, this solution does come with a trade-off in the form of increased gas costs. Since buyers need to actively call the withdraw() function to retrieve their funds, each transaction incurs a separate gas fee. While this results in a slightly higher transaction cost for participants, it is a necessary compromise to bolster the robustness and security of the refund process. Ensuring the integrity and fairness of the auction process justifies this additional cost, as it helps maintain the overall health and trust in the system.

B. Self-Enforcing Smart Contract

To mitigate the risks posed by a malicious auctioneer in on-chain Dutch auctions, it is vital to limit the role of the auctioneer to setting up initial parameters of the auction. This approach involves predefining key elements such as the auction's duration, initial price, reserve price, and the token address. Once these parameters are established, they should remain immutable to ensure fairness and transparency throughout the auction process.

Subsequently, the core functions of the auction, including the distribution of tokens to winners, the calculation of prices, and the handling of refunds, should be fully automated and governed by the smart contract itself. This concept of a self-enforcing smart contract is pivotal in enhancing the integrity of the auction. By automating these crucial aspects, the system eliminates potential manipulation by the auctioneer and ensures that the auction adheres strictly to its predefined rules. Such a design not only enhances security but also bolsters the efficiency and reliability of the auction process in a blockchain environment.

The implemented solution involves applying a binary search algorithm to determine the clearing price in the dutch auction. In a self-enforcing smart contract, the contract can accurately derive important information without requiring input from other users. Therefore, the following problem needed to be solved by smart contract.

Suppose that there are n bidders, the amount of tokens sold are modeled as.

$$\text{Number of tokens Sold} = \sum_{i=1}^{n} \lfloor \frac{Bids(i)}{ClearingPrice} \rfloor$$

$$Bids(i) = \text{number of bids placed by bidder } i$$

The objective is to find the maximum clearing price such that either the clearing price equals the reserve price or the number of tokens sold equals the initial amount of tokens initiated by the seller. This can be accomplished by binary searching the range of clearing prices.

The time complexity of this approach is $O(n \times \log(T))$ where $T =$ Initial Price $-$ Reserve Price.

While determining the clearing price is commonly performed off-chain to reduce costs, in this scenario, it is executed on-chain to ensure the integrity of the contract. However, it's important to note that this on-chain execution is likely to result in higher gas costs.

C. Encrypted Pre-orders

1) Introduction: To address the issue of frontrunning in Dutch auctions, a novel approach can be employed involving the use of preorders with encryption techniques. In this method, each bid is composed of two components: the bid amount ('bidAmount') and the time at which the bidder wishes to execute the bid ('executeAt'). These components are encrypted together using the public key of the auction, which could be based on advanced encryption standards like Elliptic Curve Cryptography (ECC). This encryption ensures that only the auction, holding the corresponding private key, can decrypt and access the details of 'executeAt' and 'bidAmount'. This timing element is crucial in a Dutch auction, where the price of tokens depreciates linearly over time. By allowing participants to specify when their bid should be executed, they can accurately estimate the token price at that particular moment.

Unlike traditional dutch auction mechanisms where the number of pending transactions can be visible, the preorder method obscures the immediacy and specifics of these transactions. Since these pre-orders may not be executed instantly but at predetermined times, it reduces the propensity for participants to engage in "gas wars" based on the visible demand. Participants can plan their bids more strategically without the pressure of immediate competition, thus aligning more closely with the intended depreciative pricing structure of Dutch auctions.

2) Implementing pre-orders in Dutch Auction: When a state-changing function function is called, smart contract is programmed to check if there are any pre-order bids eligible for execution, specifically pre-order bids where the current time surpasses the 'executeAt' parameter set in the bid. These bids are processed according to the 'executeAt' parameter, meaning that bids with an earlier 'executeAt' timestamp are given priority. This prioritisation is logical in the context of a Dutch auction, as bids committed at an earlier time correspond to a higher price per unit, given the decreasing price model of such auctions.

To guarantee the commitment of participants in pre-order bids within a Dutch auction, it is essential to ensure that these bids are binding, similar to standard bids. This means that bidders must commit the full Ethereum (ETH) balance required for their bid at the time of placing it. In the event that a pre-order bid is only partially executed or not executed at all, due perhaps to the dynamic nature of Dutch auctions where prices decrease over time, there should be a mechanism for bidders to reclaim their funds. This is where the withdraw() function comes into play to retrieve their unspent ETH.

3) Limitations: Implementing pre-orders in a Dutch auction using smart contracts raises significant concerns regarding computational efficiency and associated costs, particularly when it involves complex processes like decryption. Conducting decryption within the smart contract can be computationally expensive, leading to higher gas costs.

To address this challenge, further study and exploration into off-chain computation is advisable. Off-chain computation involves performing certain tasks outside the blockchain and then providing cryptographic proof to the smart contract that the data or computation is valid and trustworthy. This approach can significantly reduce the computational load on the smart contract, thereby lowering the gas costs.

However, this off-chain solution must be designed carefully to ensure that the cryptographic proofs are not only valid but also delivered within a reasonable timeframe. If the off-chain solution fails to provide the necessary decryption data promptly, it could lead to delays or other issues in the auction process.

In cases where the off-chain solution is unable to provide the decryption data in reasonable time, the smart contract itself might need to perform the computation. This fallback mechanism is crucial to prevent potential issues such as distributed denial-of-service (DDoS) attacks, where an attacker might try to overload the off-chain computation process to disrupt the auction.

IV. Conclusions

In conclusion, Dutch auctions implemented through smart contracts face several security challenges that necessitate thoughtful solutions. One prominent issue is the potential for malicious bidders to disrupt the refund process, highlighting the need for a pull-based withdrawal system. This approach empowers participants to initiate their own refunds, thereby mitigating the risk of the entire process being halted by a few bad actors.

Additionally, the threat of malicious auctioneers cannot be overlooked. Such actors might engage in undesirable behaviours like withholding refunds or tokens, or manipulating prices. To counter this, the implementation of self-enforcing smart contracts is crucial. These contracts limit the auctioneer's role to setting initial, immutable parameters, thus ensuring a more secure and fair auction process.

Furthermore, Dutch auctions are susceptible to front-running attacks, where bidders can observe pending transactions and engage in gas wars, ultimately leading to reduced net revenue for the auction. To combat this, incorporating an encrypted pre-order feature is essential to obscure the visibility of bids, reducing the likelihood of gas wars and maintaining a level playing field for all participants. Encrypted pre-orders can balance the auction dynamics, preventing participants from making decisions based on visible demand and ensuring a more efficient and equitable auction environment.

REFERENCES

[1] C. Braghin, S. Cimato, E. Damiani, and M. Baronchelli, “Designing smart-contract based auctions,” Security with Intelligent Computing and Big-data Services, pp. 2–3, 2019. doi:10.1007/978-3-030-16946-6_5

[2] C. Pop et al., “An Ethereum-based implementation of English, Dutch and first-price sealed-bid auctions,” 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), 2020. doi:10.1109/iccp51029.2020.9266180

[3] T. Langston, “$34m locked in a smart contract. Was the AKUTARS exploit avoidable?,” nftnow, 2023. [Online]. Available: https://nftnow.com/features/akutars-exploit-34-million-locked-in-smart-contract/

[4] P. Germouty, E. Larraia, and W. Zhang, “Publicly Verifiable Auctions with Privacy,” Cryptology ePrint Archive, Paper 2023/608, 2023. [Online]. Available: https://eprint.iacr.org/2023/608

[5] M. Bennett et al., “Going, going, gone: Competitive decision-making in Dutch auctions,” Cognitive Research: Principles and Implications, vol. 5, no. 1, 2020. doi:10.1186/s41235-020-00259-w

[6] M. Zhu, “How auction theory informs implementations,” a16z crypto, https://a16zcrypto.com/posts/article/how-auction-theory-informs-implementations/ (accessed Nov. 30, 2023).

[7] M. Kokaras and M. Foti, “The cost of privacy on Blockchain: A study on sealed-bid auctions,” Blockchain: Research and Applications, vol. 4, no. 3, p. 100133, 2023. doi:10.1016/j.bcra.2023.100133