Decentralized Public Ledger for Auctions

Bruno Leal
Dept. Ciências de Computadores
FCUP (DCC)
Porto, Portugal
up202008047

***Abstract*—This report presents the design and implementation of a secure, decentralized auction platform based on a public blockchain and a structured P2P overlay network. The system supports English-style auctions with tamper-proof transaction logging and resilient communication using S/Kademlia. Each operation—such as auction creation, bidding, and closing—is stored in the blockchain using cryptographically signed transactions, ensuring non-repudiation and transparency. The architecture was designed with modularity in mind, enabling future integration of Proof-of-Reputation and advanced trust mechanisms. This work demonstrates the feasibility of building trustworthy distributed systems for sensitive operations like auctions.**

I. Introduction

In this project, we developed a decentralized public ledger for managing auctions over a peer-to-peer (P2P) network. Inspired by the design principles of Bitcoin [1] and Ethereum [2], but adapted for auction scenarios, our system stores auction-related transactions (such as creation, bidding, and closure) in a secure blockchain maintained by a distributed network of nodes .

The solution is fully implemented in Java and integrates three core components: a modular blockchain ledger supporting Proof-of-Work (PoW) and designed to accommodate Proof-of-Reputation (PoR); a secure and scalable P2P overlay based on the S/Kademlia protocol for node discovery and message propagation; and an English-style auction mechanism using public-key cryptography to ensure integrity and non-repudiation of bids.

The system tries to ensure resilience through decentralized data propagation, trust mechanisms, and a fault injection module that simulates node failures. All communication between nodes is done over UDP using custom-serialized messages. This report outlines the architecture, design decisions, key features, and assumptions made during implementation.

II. System Architecture

The system is composed of three modular layers: the blockchain ledger, the peer-to-peer (P2P) communication layer, and the auction mechanism. At the core lies a custom-built blockchain, which maintains an immutable sequence of blocks. Each block contains up to four signed transactions, including auction creation, bid submission, and auction closure. Transactions are signed using RSA public-key cryptography and verified before inclusion. Block mining uses a simplified Proof-of-Work (PoW) consensus mechanism with configurable difficulty, and support for Proof-of-Reputation (PoR) is integrated into the design for future extensibility.

The P2P layer is based on a secure implementation of the S/Kademlia protocol. Nodes communicate over UDP sockets and serialize all messages using Java's object streams. Routing tables are updated using the XOR metric to ensure efficient node discovery. Messages such as `STORE`, `FIND_NODE`, and `RESPONSE` enable the propagation of auction data, blockchain transactions, and block synchronization across the network. Reputation and trust mechanisms are designed to be integrated with PoR and to help mitigate Sybil and Eclipse attacks.

On top of this network, the auction system enables any node to act as a seller or buyer. Auctions follow the English auction model, with single-attribute bidding. Each auction is uniquely identified and gossiped to the network. A CLI-based menu allows users to create auctions, place bids, and close auctions, all of which are recorded on the blockchain. To demonstrate resilience, the system supports the dynamic termination of nodes while maintaining consistency and data availability through replication and gossip.

III. Distributed Ledger Design

The distributed ledger is implemented as a non-permissioned blockchain designed to persist auction-related transactions. Each node maintains a local copy of the ledger, composed of a list of sequential blocks. The first block is a genesis block, initialized with a signed transaction that certifies the start of the blockchain. Each subsequent block stores up to four transactions, after which a new block is mined using a simplified Proof-of-Work (PoW) mechanism [1] . The mining process includes computing the Merkle root of the block's transactions and iteratively searching for a valid nonce that satisfies a configurable difficulty target.

Transactions are signed using RSA digital signatures to ensure authenticity and non-repudiation. Before a transaction is added to a block, its signature is validated against the sender's public key. This guarantees that only legitimate actions are recorded. Once a block is mined, it is propagated across the network using the P2P layer, and validated by peers through hash verification, timestamp consistency, and integrity of the block linkage. The system also supports future extension for Proof-of-Reputation (PoR) by associating transaction validity and block mining eligibility with node behavior and trustworthiness [3].

All auction operations—such as auction creation, bidding, and closure—are logged as individual transactions. The blockchain design is modular, making it adaptable to different consensus strategies and scalable across multiple nodes. To maintain system resilience, all valid transactions and mined blocks are broadcast via Kademlia-based message routing to ensure replication and consistency across participating nodes.

IV. Secure P2P Communication

The peer-to-peer communication layer is built on a customized implementation of the S/Kademlia protocol [4], enabling efficient and secure message routing between nodes. Each node in the network is identified by a cryptographic ID derived from its public key, and routing is performed based on the XOR distance between these IDs [5]. The routing table is organized into buckets, and each node maintains knowledge of a subset of the network based on proximity in ID space, promoting scalability and fault tolerance .

To protect against Sybil and Eclipse attacks, the system relies on public key cryptography to enforce identity uniqueness and authenticity. All messages exchanged—such as FIND_NODE, STORE, and RESPONSE—carry metadata that includes the sender's IP, port, and public key, allowing recipients to validate origin and integrity. Trust mechanisms are integrated into the routing logic, preparing the system to support Proof-of-Reputation schemes where misbehaving nodes can be penalized or excluded based on historical actions.

All blockchain operations, including transaction dissemination and block propagation, are handled through this P2P layer. Messages are serialized and transmitted via UDP sockets, with custom logic for deserialization and message handling on the receiving end. The system also supports a gossip-based broadcasting mechanism for critical operations, such as block propagation and auction updates. This decentralized architecture enhances robustness and ensures that all nodes can remain synchronized, even in the presence of faults or targeted disruptions.

V. Auction Mechanisms

The application implements a decentralized English auction system where each node can act as both auctioneer and bidder. Auctions are defined by a unique ID, item name, description, duration, starting bid, and reserved price, and are associated with the public key of the auctioneer. Once an auction is created, it is serialized and broadcasted through the P2P layer using a STORE message, ensuring that all nodes can replicate and persist auction data locally. This guarantees availability and consistency across the network.

Bidding is performed through the CLI interface, where users can explore active auctions, place bids, and inspect their own auctions. Each bid is validated against the current highest bid and the auction's status. Valid bids are propagated using the same gossip mechanism and are persisted through transactions recorded on the blockchain. All bid and auction-related actions are signed using the private key of the user, ensuring non-repudiation and authenticity.

The auction lifecycle includes creation, bidding, and termination. Termination of auctions (by the auctioneer) also results in broadcasted messages, alerting all nodes of the auction's closure and winner. Each of these operations is recorded as a transaction, contributing to a verifiable and immutable history stored in the distributed ledger. The system supports real-time updates across the network, ensuring that all participants have consistent auction state despite the lack of a central authority.

VI. Assumptions and Limitations

Due to the scope and time constraints of the project, several practical and theoretical assumptions were made. First, it is assumed that each node in the network runs in a trusted environment where the user controls the node's private keys securely. The identity of nodes is derived from cryptographic key pairs, and there is no external identity validation mechanism.

The implementation assumes reliable UDP communication over localhost for demonstration purposes, which simplifies NAT traversal and real-world network failures. While the system supports node discovery and message routing using S/Kademlia, no NAT hole punching or long-term node persistence mechanisms are currently implemented. Additionally, auctions and bids are stored in memory and not persisted to disk, which could result in data loss if a node restarts.

Proof-of-Reputation is not fully integrated due to complexity, and the consensus relies solely on a simplified Proof-of-Work model. The mining process is local and does not involve actual competition between nodes. Also, trust and Sybil attack resistance mechanisms are outlined but not deeply enforced at the cryptographic level.

Despite these limitations, the system remains modular and extensible for future improvements such as secure pub/sub, persistent storage, and enhanced trust evaluation mechanisms.

VII. Conclusion

This project presents a fully functional distributed system for secure and decentralized auction management, leveraging blockchain technology and a secure P2P communication layer. The system successfully integrates key components such as a modular public ledger with Proof-of-Work consensus, a Kademlia-based P2P network with support for secure node discovery, and an auction mechanism that ensures bid integrity and transparency through cryptographic transactions.

The implementation prioritizes modularity, allowing the consensus mechanism and trust model to be extended in the future, for instance with full Proof-of-Reputation integration. The system also demonstrates resilience to node failures, and its design encourages fault tolerance through decentralization and replication.

While some limitations remain, especially regarding persistent storage and advanced security policies, the architecture lays a strong foundation for future development. The project showcases the practical application of secure distributed systems, and demonstrates how blockchain and P2P technologies can be combined effectively to solve real-world problems such as auction fairness, traceability, and censorship resistance.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," https://ethereum.org/en/whitepaper/, 2014.

[3] J. Yu, D. Kozhaya, J. Decouchant, and P. E. Verissimo, "Repucoin: Your reputation is your power," *IEEE Transactions on Computers*, vol. 68, no. 8, pp. 1225–1237, 2019.

[4] I. Baumgart and S. Mies, "S/kademlia: A practicable approach towards secure key-based routing," in *Parallel and Distributed Systems, 2007 International Conference on*. IEEE, 2007, pp. 1–8.

[5] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 53–65.