Decentralized Blockchain-Based Cryptocurrency Wallet System with Automated Zakat Deduction

Muhammad Hasnain
Dept. of Software Engineering
FAST-NUCES
Pakistan
mhussnainzardari34@gmail.com

Zainab Khurram
Dept. of Software Engineering
FAST-NUCES
Pakistan
zainab.khurram@gmail.com

***Abstract*—This paper presents the design and implementation of a decentralized cryptocurrency wallet system built on a custom blockchain infrastructure with integrated Islamic finance principles. The system implements complete blockchain architecture using Proof-of-Work consensus, UTXO-based transaction model, and cryptographic security mechanisms. A novel feature includes automated Zakat (Islamic alms) deduction at 2.5% monthly intervals, demonstrating the integration of traditional Islamic finance with modern blockchain technology. The system comprises a Go-based backend implementing core blockchain functionality, a React-based frontend for user interaction, and MongoDB Atlas for persistent storage. Performance analysis demonstrates successful transaction processing with cryptographic verification, double-spend prevention, and consistent blockchain integrity. This work contributes to the growing field of blockchain applications in financial technology while addressing specific requirements for Islamic finance compliance.**

***Index Terms*—Blockchain, Cryptocurrency Wallet, UTXO Model, Proof-of-Work, Digital Signatures, Zakat Automation, Distributed Ledger, Go Programming, React Framework, MongoDB.**

I. Introduction

The emergence of blockchain technology, pioneered by Bitcoin in 2008, has revolutionized the concept of decentralized digital currency systems [1]. Traditional financial systems rely on centralized authorities for transaction validation and record-keeping, introducing single points of failure and trust dependencies. Blockchain technology addresses these limitations by distributing transaction records across a peer-to-peer network, ensuring transparency, immutability, and security through cryptographic mechanisms [2].

[Image of centralized vs decentralized network architecture]

This paper presents a comprehensive blockchain-based cryptocurrency wallet system designed for domestic money transfers with an integrated Islamic finance feature: automated Zakat deduction. The system implements core blockchain principles including Proof-of-Work (PoW) consensus, Unspent Transaction Output (UTXO) model, and cryptographic digital signatures using RSA-2048 and SHA-256 algorithms.

A. Motivation

The primary motivation for this project stems from three key objectives:

1) To understand and implement fundamental blockchain concepts including distributed ledger technology, consensus mechanisms, and cryptographic security.
2) To create a practical cryptocurrency wallet system that demonstrates real-world applicability for domestic financial transactions.
3) To integrate Islamic finance principles, specifically automated Zakat calculation and deduction, showcasing blockchain potential for compliance with religious financial requirements [14].

Traditional cryptocurrency systems do not accommodate specific requirements of Islamic finance, particularly the obligation of Zakat—a mandatory charitable contribution calculated as 2.5% of wealth held for a lunar year. This system demonstrates how blockchain technology can be adapted to meet such requirements automatically and transparently.

B. Project Scope

The project encompasses the following components:

- Custom blockchain implementation with genesis block initialization and chain validation.
- UTXO-based transaction model preventing double-spending.
- RSA-2048 cryptographic key pair generation for wallet creation.
- Digital signature mechanism for transaction authentication.
- Proof-of-Work mining with adjustable difficulty.
- MongoDB Atlas integration for persistent blockchain storage.
- RESTful API with 31 endpoints for blockchain operations.
- React-based user interface with real-time blockchain visualization.
- Automated Zakat deduction system with transparent logging.
- Email-based OTP verification for user authentication.

C. Paper Organization

The remainder of this paper is organized as follows: Section II describes the technical architecture and technology stack. Section III details the blockchain implementation including block structure, mining process, and consensus mechanism. Section IV explains the transaction model and UTXO system. Section V covers the wallet infrastructure and cryptographic security. Section VI describes the database design and storage mechanisms. Section VII presents the frontend implementation. Section VIII discusses security considerations. Section IX presents testing results and validation. Section X analyzes challenges faced during development. Section XI presents results and observations. Section XII concludes the paper with future work recommendations.

II. Technical Architecture

The system architecture follows a three-tier model comprising frontend presentation layer, backend application layer, and database persistence layer. This section describes the technology stack and architectural decisions.

[Image of three-tier web application architecture diagram]

A. Technology Stack

1) Backend Framework: Go (Golang) 1.21: High-performance compiled language suitable for concurrent operations with strong standard library support for cryptographic functions [5].

2) Frontend Framework: React 18 with Vite: Component-based architecture for modular UI development and Virtual DOM for efficient rendering of blockchain data [6].

3) Database: MongoDB Atlas (Cloud NoSQL): Document-based storage suitable for blockchain data structures with flexible schema accommodating evolving blockchain requirements [7].

B. System Architecture Design

The system follows a client-server architecture with clear separation of concerns:

1) **Presentation Layer (Frontend):** React components, Context API, Axios interceptors.
2) **Application Layer (Backend):** RESTful API, Middleware, Service layer, Crypto layer.
3) **Data Layer:** MongoDB collections (users, wallets, utxos, transactions, blocks).

III. Blockchain Implementation

This section details the core blockchain implementation, including data structures, mining algorithms, and consensus mechanisms.

[Image of blockchain block structure and merkle tree]

A. Block Structure

Each block in the blockchain contains Index, Timestamp, Transactions, PreviousHash, Hash, Nonce, MerkleRoot, Difficulty, and Miner ID. The hash is calculated as follows:

$$Hash = SHA256(Idx+TS+Tx+PrevHash+Nonce+MR) \quad (1)$$

B. Genesis Block

The blockchain initializes with a genesis block (Index 0) containing the note ”Genesis Block - Blockchain Initialized” and difficulty 4.

C. Proof-of-Work Mining Algorithm

The system implements a Proof-of-Work consensus mechanism requiring miners to find a nonce value such that the block hash starts with a specific number of leading zeros. Default difficulty is 4 (requires hash starting with ”0000”).

IV. Transaction Model and UTXO System

The system implements the Unspent Transaction Output (UTXO) model, similar to Bitcoin, for tracking coin ownership and preventing double-spending [1].

[Image of unspent transaction output UTXO model diagram]

A. UTXO Model Architecture

Each transaction consumes existing UTXOs (inputs) and creates new UTXOs (outputs).

B. UTXO Selection Algorithm

When creating a transaction, the system selects UTXOs using a greedy algorithm: Fetch unspent UTXOs, Sort by amount (descending), Accumulate until total covers amount.

V. Wallet System and Cryptographic Security

A. Key Pair Generation

The system uses RSA-2048 for asymmetric cryptography [16].

[Image of asymmetric encryption and digital signature process]

B. Private Key Encryption

Private keys are never stored in plaintext. They are encrypted using AES-256-GCM [18].

VI. Database Design and Storage

MongoDB Atlas serves as the persistent storage layer. Key collections include Users, Wallets, UTXOs, Transactions, and Blocks. Over 30 indexes optimize query performance.

VII. Frontend Implementation

The React application includes Authentication pages (Login, Register), a Dashboard, Send Money interface, Transaction History, Blockchain Explorer, and Reports & Analytics. Features include real-time balance updates and blockchain visualization.

VIII. Security Considerations

Authentication is secured via JWT and bcrypt. Transactions use digital signatures to ensure non-repudiation. API security includes rate limiting and input sanitization to prevent injection attacks.

IX. Testing and Validation

Cryptographic functions and UTXO logic were unit tested. Integration tests covered the full authentication and transaction lifecycle. Performance testing showed average block times of 2-5 seconds at difficulty 4.

X. Challenges Faced

Major challenges included managing UTXO complexity and balancing Proof-of-Work difficulty for demonstration purposes. These were resolved using greedy algorithms and adjustable difficulty settings.

XI. Results and Observations

The system successfully mined over 150 blocks and processed 500+ transactions. Automated Zakat deductions functioned correctly at 2.5% monthly intervals, skipping zero-balance accounts.

XII. Conclusion and Future Work

This project successfully demonstrates a decentralized cryptocurrency wallet system integrating core blockchain principles with practical Islamic finance requirements. It achieves all primary objectives including PoW consensus, UTXO model, and automated Zakat deduction. Future work includes implementing peer-to-peer networking and smart contracts.

XIII. References

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[2] A. Narayanan et al., "Bitcoin and Cryptocurrency Technologies," Princeton University Press, 2016.

[3] "Go Programming Language Documentation," The Go Authors, 2023.

[4] "React Documentation," Meta Platforms, Inc., 2023.

[5] "MongoDB Documentation," MongoDB, Inc., 2023.

[6] M. A. Khan, "Blockchain Technology in Islamic Finance," Journal of Islamic Financial Studies, 2019.

[7] "RSA Cryptography Standard," RSA Laboratories, PKCS #1 v2.2.

[8] "Advanced Encryption Standard (AES)," NIST FIPS PUB 197, 2001.