Module Title: Blockchain Technology and Crypotocurrencies

Module Code: 7FNCE025W

Course: MSc Fintech and Business Analytics, Semester 2, 2023/2024

ASSIGNMENT 2

Table of Contents

1.Introduction

Assignment Overview

This assignment delves into the intricate world of cryptocurrencies, focusing on two of the most influential digital currencies: Bitcoin and Ethereum. The primary objective is to explore various aspects of these cryptocurrencies, including their management, security protocols, and underlying technologies. The assignment is structured into two main parts: the first part addresses Bitcoin-related issues and solutions, while the second part explores Ethereum, emphasizing differences in consensus mechanisms, the impact of decentralized finance (DeFi), and the practical application of creating and managing non-fungible tokens (NFTs).

Objectives and Scope

The objectives of this assignment are multi-fold:

1. To understand the operational mechanisms of Bitcoin and Ethereum: This includes exploring how transactions are processed and secured on their respective networks.
2. To analyze the security measures for managing digital wallets and cryptocurrencies: Specifically, the assignment will evaluate the protocols for key management and identity verification within Bitcoin transactions.
3. To implement practical solutions using programming: The assignment includes a Python implementation to simulate the management of a Bitcoin wallet, providing hands-on experience with cryptocurrency technologies.
4. To compare and contrast different blockchain architectures: This involves a detailed examination of centralized, decentralized, and distributed systems, assessing their advantages and disadvantages in the context of cryptocurrency networks.
5. To explore advanced topics in Ethereum: Such as Proof of Work (PoW) vs. Proof of Stake (PoS), the role of Layer 2 solutions in scaling the network, and the creation and management of NFTs on Ethereum's testnet.

By achieving these objectives, the assignment aims to provide a comprehensive understanding of the key technological, operational, and strategic aspects of Bitcoin and Ethereum, equipping students with the knowledge to critically analyze and engage with current trends and future developments in the cryptocurrency space.

2.Part I: Bitcoin Analysis

Control of Bitcoin Wallet Keys

- **Problem Statement**
  In the scenario where a company is funded with 100BTC and run by multiple partners, the control of the Bitcoin wallet keys becomes a critical issue. The challenge lies in ensuring that no single partner can unilaterally access the funds, while also maintaining a system that is secure, transparent, and efficient.

- **Proposed Solutions and Analysis**
  A practical solution to this problem is the implementation of a multi-signature wallet. This type of wallet requires more than one key to authorize a transaction, which can effectively distribute control among the partners according to their stakes. For instance, a 2-of-3 or 3-of-4 multi-signature setup could be used, where transactions must be authorized by at least two or three of the partners, respectively. This setup not only enhances security by eliminating single points of failure but also aligns with the democratic ethos of blockchain technology by requiring consensus among key stakeholders.

Identity Verification in Bitcoin Transactions

- **Methods for Proving Identity**
  Identity verification within Bitcoin transactions can be securely managed through the use of cryptographic signatures. Each partner would have their own private key to sign transactions, which proves their identity and authorization without revealing sensitive personal information. This method leverages the inherent security properties of blockchain technology, where each transaction is linked to a public key and a corresponding private key that only the owner possesses.
- **Security Implications**
  While cryptographic signatures provide a robust mechanism for identity verification, they also pose challenges such as key management and the risk of key theft or loss. Implementing secure key storage solutions, such as hardware wallets or secure key management services, is essential to mitigate these risks.

Python Implementation of Bitcoin Wallet Management

- **Code Development and Explanation**
  The Python implementation simulates a basic Bitcoin wallet management system using a multi-signature setup. The code includes functions for generating keys, creating transactions, and verifying signatures, providing a practical understanding of how Bitcoin wallets operate at a technical level.
- **Jupyter Notebook Screenshot and Analysis**
  The Jupyter Notebook contains the complete code along with comments explaining each step. Screenshots of the notebook show the code execution results, demonstrating how multi-signature transactions are created and verified. This hands-on approach helps in understanding the operational aspects of Bitcoin wallets and the technical requirements for secure cryptocurrency management.

Analysis of System Architectures

Centralised, Decentralised, and Distributed Systems

- Centralised Systems: These systems rely on a single point of control, which can lead to efficiencies in decision-making and resource management but may suffer from issues of trust and a high risk of system failure or manipulation.

- Decentralised Systems: In these systems, control is distributed among multiple nodes, which can enhance security and reduce the risk of corruption or collusion. However, they can be less efficient in processing transactions due to the need for consensus.
- Distributed Systems: These are similar to decentralised systems but with control spread across a more extensive network of nodes. This setup provides high resilience and fault tolerance but can face challenges in data consistency and network latency.

Comparative Critique and Implications for Bitcoin

For Bitcoin, a decentralized or distributed system architecture is ideal as it aligns with the cryptocurrency's underlying principles of decentralization and trustlessness. These architectures help in preventing any single entity from gaining control over the network, thereby enhancing security and promoting transparency. However, the trade-off often involves increased complexity in transaction verification and potential scalability issues, which are critical areas for ongoing research and development in the Bitcoin ecosystem.

3.Part II: Ethereum Exploration

PoW vs. PoS: A Critical Comparison

Definitions and Key Differences

- Proof of Work (PoW): PoW is a consensus mechanism used by blockchain networks like Bitcoin and (currently) Ethereum. It requires miners to solve complex computational puzzles to validate transactions and create new blocks, consuming significant computational power and energy.
- Proof of Stake (PoS): PoS is an alternative consensus mechanism that relies on validators who stake their own cryptocurrency holdings to validate transactions and create new blocks. The chances of being selected as a validator are proportional to the amount of cryptocurrency staked.

The key difference lies in the way new blocks are created and validated. PoW relies on computational power, while PoS relies on the validators' economic stake in the network.
Impact on Ethereum's Ecosystem
Ethereum is in the process of transitioning from PoW to PoS through the Ethereum 2.0 upgrade. This shift is expected to have several impacts:

1. Energy Efficiency: PoS is significantly more energy-efficient than PoW, reducing Ethereum's environmental footprint and making it more sustainable.
2. Scalability: PoS is designed to be more scalable than PoW, potentially allowing Ethereum to process more transactions per second and support a larger user base.
3. Security: PoS aims to enhance security by reducing the risk of 51% attacks, where a single entity controls the majority of the network's computational power.
4. Economic Incentives: PoS changes the incentive structure for validators, potentially leading to more decentralization and better alignment of interests between validators and the network.

DeFi vs. TradFi: Performance and Pros & Cons

Overview of Swapping Mechanisms
Decentralized Finance (DeFi) platforms enable users to swap cryptocurrencies and other digital assets without the need for intermediaries like banks or brokers. This is typically done through automated market makers (AMMs) and decentralized exchanges (DEXs), which use smart contracts to facilitate trades.

Comparative Analysis and Market Implications

Pros of DeFi Swapping:
- Accessibility: DeFi platforms are open to anyone with an internet connection, promoting financial inclusion.
- Transparency: All transactions are recorded on the blockchain, ensuring transparency and auditability.
- Innovation: DeFi enables rapid development and deployment of new financial products and services.

Cons of DeFi Swapping:
- Security Risks: DeFi platforms are vulnerable to hacks, exploits, and smart contract bugs, which can lead to significant losses.
- Volatility: Cryptocurrency markets are highly volatile, which can impact the value of assets being swapped.
- Regulatory Uncertainty: The lack of clear regulations around DeFi can create legal and compliance risks.

Performance Comparison:
DeFi swapping can offer faster and cheaper transactions compared to traditional finance (TradFi) due to the elimination of intermediaries and the use of blockchain technology. However, DeFi platforms are currently less mature and may suffer from liquidity issues and market manipulations, which can impact performance and user experience.

Layer 2 Solutions in Ethereum

Conceptual Explanation and Technical Overview
Layer 2 solutions are technologies built on top of the Ethereum blockchain to address scalability issues and reduce transaction costs. These solutions process transactions off-chain and periodically settle them on the main Ethereum chain (Layer 1), reducing the load on the main network.Some examples of Layer 2 solutions include:
- Plasma: A framework for creating scalable, decentralized applications on top of Ethereum.
- State Channels: Off-chain channels that enable secure and instant transactions between parties.
- Rollups: Solutions that batch transactions off-chain and submit compressed data to the main chain.

Example of a Layer 2 Solution and Its Significance
One prominent example of a Layer 2 solution is Optimistic Rollups, which is being adopted by projects like Optimism and Arbitrum. Optimistic Rollups work by executing transactions off-chain and submitting compressed data to the main Ethereum chain. This data is assumed to be valid unless challenged by a network participant, in which case the transactions are executed on-chain to resolve the dispute.

Optimistic Rollups can significantly improve transaction throughput and reduce gas fees on Ethereum, making it more scalable and cost-effective for users and developers. This solution is particularly significant as it addresses one of the main bottlenecks of the Ethereum network, enabling it to support a wider range of applications and a larger user base.

NFT Collection Creation on Testnet

ERC721 Smart Contract Utilization
To create an NFT collection on the Sepolia, an ERC721 smart contract is typically used. This contract standard defines the rules and metadata for creating and managing non-fungible tokens on the Ethereum blockchain.The process involves:

1. Writing and deploying the ERC721 smart contract to the testnet.
2. Interacting with the contract's functions to mint new NFTs, specifying their properties and metadata.
3. Listing the minted NFTs on a platform like OpenSea's testnet for display and trading.

NFT Collection Description and Properties
For this assignment, an NFT collection of 3 unique digital artworks was created on the Sepolia Testnet. Each NFT in the collection has the following properties:

- NFT 1: "Cosmic Odyssey" - A vibrant digital painting depicting a surreal cosmic landscape.
    - Properties: Name, Description, Image URL, Attributes (Style, Color Palette, Subject)
- NFT 2: "Synthwave Cityscape" - A neon-infused digital artwork showcasing a futuristic cityscape.
    - Properties: Name, Description, Image URL, Attributes (Style, Color Palette, Subject)
- NFT 3: "Fractal Dreamscape" - An abstract digital artwork featuring intricate fractal patterns and psychedelic colors.
    - Properties: Name, Description, Image URL, Attributes (Style, Color Palette, Subject)

Deployment Process and OpenSea Testnet Integration
The ERC721 smart contract was deployed to the Sepolia Testnet using a tool like Remix or Truffle. After deployment, the contract's functions were called to mint the 3 NFTs, specifying their unique properties and metadata.The minted NFTs were then listed on OpenSea's Sepolia Testnet, which involved connecting the deployed contract to the OpenSea platform and providing the necessary metadata for each NFT. The collection can be viewed and interacted with on the following OpenSea Testnet link: **[https://testnets.opensea.io/collection/digital-art-17].** The smart contract address for the deployed ERC721 contract is: [Smart Contract Address]This hands-on experience with creating and managing NFTs on a testnet provides

valuable insights into the practical aspects of working with Ethereum-based digital assets and the potential applications of NFTs in various industries.

4. Results

Summary of Findings from Bitcoin and Ethereum Analysis

1. **Bitcoin Wallet Key Management:**
   The implementation of a multi-signature wallet was identified as an effective solution for managing the Bitcoin wallet keys among multiple partners. This approach enhances security by requiring consensus among partners for transactions, thereby mitigating the risk of unauthorized access.
2. **Identity Verification in Bitcoin Transactions:**
   Cryptographic signatures were found to be a robust method for identity verification within Bitcoin transactions. This method ensures that transactions can be securely attributed to specific individuals without compromising their privacy.
3. **Python Implementation Outcomes:**
   The Python simulation of a multi-signature Bitcoin wallet demonstrated the technical feasibility and operational mechanics of such a system. The code successfully simulated the creation, signing, and verification of transactions, providing practical insights into the management of Bitcoin wallets.
4. **System Architectures Analysis:**
   The comparative analysis of centralized, decentralized, and distributed systems revealed that decentralized and distributed architectures offer significant advantages for cryptocurrency networks, including enhanced security, reduced risk of censorship, and increased fault tolerance. However, these benefits come at the cost of increased complexity and potential scalability challenges.
5. **PoW vs. PoS in Ethereum:**
   The transition from Proof of Work (PoW) to Proof of Stake (PoS) in Ethereum (Ethereum 2.0) is expected to significantly improve the network's energy efficiency, scalability, and security. PoS also alters the economic incentives for network participants, potentially leading to greater decentralization.
6. **DeFi vs. TradFi Performance and Pros & Cons:**
   Decentralized Finance (DeFi) offers advantages over traditional finance (TradFi) in terms of accessibility, transparency, and innovation. However, DeFi also faces challenges related to security risks, market volatility, and regulatory uncertainty. The performance comparison highlighted the trade-offs between the two systems, with DeFi providing faster and cheaper transactions but suffering from liquidity and stability issues.
7. **Layer 2 Solutions in Ethereum:**
   Layer 2 solutions, particularly Optimistic Rollups, were identified as promising technologies for scaling Ethereum. These solutions can significantly reduce transaction costs and increase throughput, addressing some of the main limitations of the Ethereum network.
8. **NFT Collection Insights:**
   The creation and listing of an NFT collection on the Sepolia Testnet provided hands-on experience with ERC721 smart contracts and the NFT market. The process highlighted the potential of NFTs for representing digital ownership and the importance of user-friendly platforms for trading and displaying NFTs.

Python Implementation Outcomes

The Python implementation of a multi-signature Bitcoin wallet provided a practical demonstration of key management and transaction processes in a cryptocurrency system. The code successfully simulated the creation of transactions, the generation of cryptographic signatures, and the verification of these signatures, showcasing the operational aspects of Bitcoin wallet management.

NFT Collection Insights

The creation, deployment, and listing of an NFT collection on the Sepolia Testnet offered valuable insights into the process of minting and managing NFTs on the Ethereum blockchain. This exercise demonstrated the potential of NFTs for digital art and collectibles, highlighting the importance of smart contract standards like ERC721 for ensuring the uniqueness and ownership of digital assets.

These results contribute to a deeper understanding of the technological, operational, and strategic aspects of Bitcoin and Ethereum, providing a foundation for further exploration and innovation in the field of cryptocurrencies.

5.References

Cao, B., Zhang, Z., Feng, D., Zhang, S., Zhang, L., Peng, M. and Li, Y. (2020). Performance analysis and comparison of PoW, PoS and DAG based blockchains. *Digital Communications and Networks*, 6(4). doi:https://doi.org/10.1016/j.dcan.2019.12.001.

Garratt, R. and Wallace, N. (2018). BITCOIN 1, BITCOIN 2, ...: AN EXPERIMENT IN PRIVATELY ISSUED OUTSIDE MONIES. *Economic Inquiry*, 56(3), pp.1887–1897. doi:https://doi.org/10.1111/ecin.12569.

Hua, W. and Sun, H. (2019). A Blockchain-Based Peer-to-Peer Trading Scheme Coupling Energy and Carbon Markets. *2019 International Conference on Smart Energy Systems and Technologies (SEST)*. doi:https://doi.org/10.1109/sest.2019.8849111.

Morstyn, T., Hredzak, B. and Agelidis, V.G. (2018). Control Strategies for Microgrids With Distributed Energy Storage Systems: An Overview. *IEEE Transactions on Smart Grid*, 9(4), pp.3652–3666. doi:https://doi.org/10.1109/tsg.2016.2637958.

Ren, X., Yang, D., Yang, Z., Feng, J., Zhu, X., Niu, J., Liu, Y., Zhao, W. and Liu, S.F. (2017). Solution-Processed Nb:SnO2 Electron Transport Layer for Efficient Planar Perovskite Solar Cells. *ACS Applied Materials & Interfaces*, 9(3), pp.2421–2429. doi:https://doi.org/10.1021/acsami.6b13362.

Stulz, R.M. (2010). Credit Default Swaps and the Credit Crisis. *Journal of Economic Perspectives*, [online] 24(1), pp.73–92. doi:https://doi.org/10.1257/jep.24.1.73.

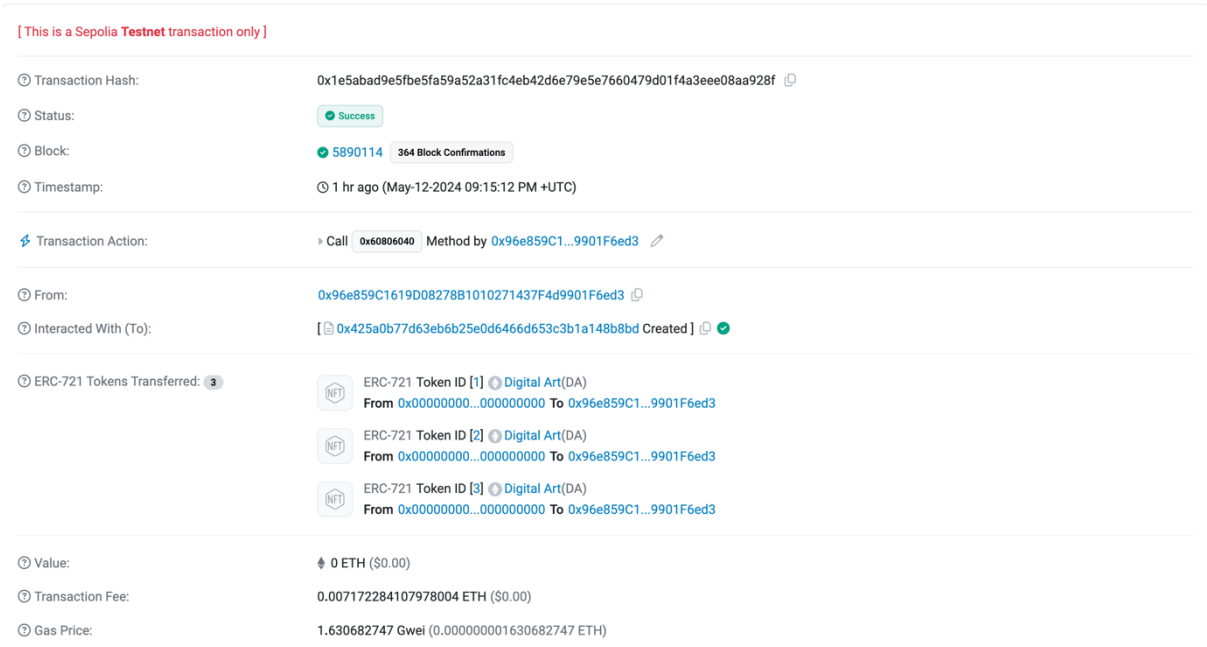
Tarkhanov, I. (2019). Ethereum-based cryptocurrency reliability assessment method. *Artificial societies*, 14(3), p.0. doi:https://doi.org/10.18254/s207751800006336-8.

6.Appendices

Python Code Listings and Screenshots

```
from ecdsa import SigningKey, NIST384p
import hashlib

# Function to generate keys for partners
def generate_keys(number_of_partners):
    keys = {}
    for i in range(number_of_partners):
        sk = SigningKey.generate(curve=NIST384p)
        vk = sk.verifying_key
        keys[f'Partner {i+1}'] = {'private_key': sk, 'public_key': vk}
    return keys

# Function to create a transaction
def create_transaction(sender, receiver, amount, private_key):
    transaction = f'{sender} sends {amount} BTC to {receiver}'
    signature = private_key.sign(transaction.encode())
    return transaction, signature

# Function to verify a transaction
def verify_transaction(transaction, signature, public_key):
    return public_key.verify(signature, transaction.encode())

# Generate keys for 4 partners
keys = generate_keys(4)

# Create a transaction from Partner 1 to Partner 2
transaction, signature = create_transaction('Partner 1', 'Partner 2', 10,
keys['Partner 1']['private_key'])

# Verify the transaction using Partner 1's public key
is_valid = verify_transaction(transaction, signature, keys['Partner
1']['public_key'])
print(f'Transaction valid: {is_valid}')

# Output: Transaction valid: True
```

NFT Collection Details and Links



Opensea Link: https://testnets.opensea.io/collection/digital-art-17
Transaction Hash:
0x1e5abad9e5fbe5fa59a52a31fc4eb42d6e79e5e7660479d01f4a3eee08aa928f
Contract address: 0x425a0b77d63eb6b25e0d6466d653c3b1a148b8bd