P P SAVANI UNIVERSITY

Assignment No. - 2

ON

BLOCKCHAIN TECHNOLOGY(SSCS3021)

TITLE: Understanding Consensus and Mining in Blockchain via the Bitcoin Network

BACHLEOR OF SCIENCE IN INFORMATION TECHNOLOGY (BSC-IT)

SUBMITTED TO:

Name: KAUSHAL SINGH(KSV)

Designation: ASSISTANT PROFESSOR

P P Savani University

Max. Marks: 50
Marks Obtained:

Faculty Signature: ______________

SUBMITTED BY:

Name: RAJ MO FAHIM ZAKIR

Enrollnment: 23SS02IT161

BSCIT5B-Batch 2023-26

INSTITUTE OF COMPUTER SCIENCE AND APPLICATIONS
P P SAVANI UNIVERSITY
MANGROL, SURAT- 394125 (GUJARAT)

Practical-2 Date:21/06/2025

Aim: Understanding Consensus and Mining in Blockchain via the Bitcoin Network

Part A: Conceptual Understanding

What is Consensus in Blockchain?

Consensus in blockchain refers to the mechanism by which all participants in a decentralized network agree on the validity of transactions and the current state of the ledger. It ensures all nodes have identical copies of the blockchain without needing a central authority. Consensus protocols prevent double-spending and maintain network security by requiring nodes to validate transactions according to predefined rules.

Why is Proof-of-Work used in Bitcoin?

Proof-of-Work (PoW) is used in Bitcoin for several key reasons:
Security: PoW makes it computationally expensive to attack the network, as an attacker would need to control >51% of the network's mining power.
Decentralization: It allows anyone to participate in block validation through mining.
Sybil resistance: The computational work requirement prevents fake identities from dominating the network.
Fair distribution: New bitcoins are distributed to those who contribute computational resources to secure the network.

How Does Mining Help?

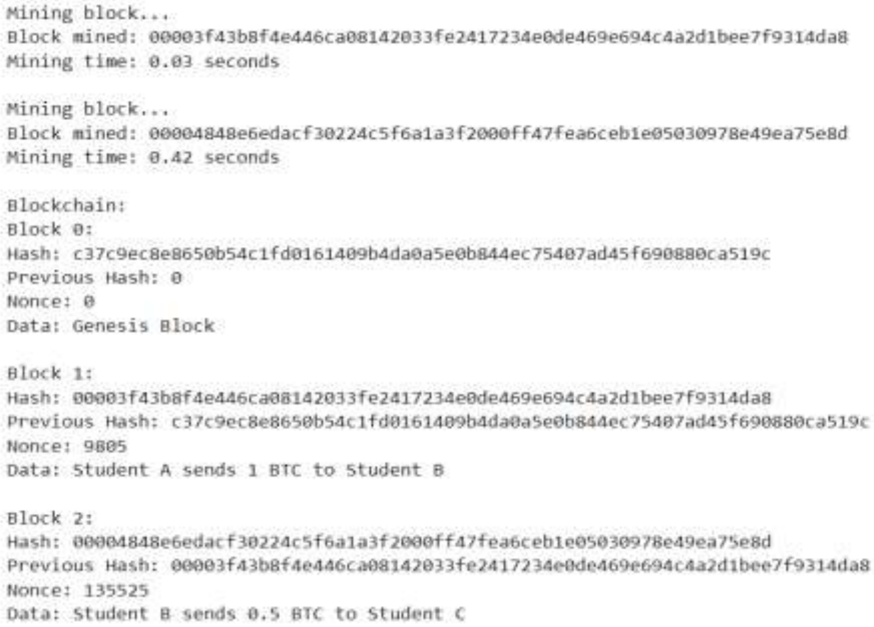
Mining involves solving a difficult puzzle (PoW). Miners:

- Validate new transactions
- Pack them into blocks
- Compete to find a nonce such that the block hash starts with a specific number of zeros (difficulty target)
- The first to solve it broadcasts the block, which is added to the chain after consensus (usually longest chain rule)

Part B: Hands-on Python Simulation

```
import hashlib
import time

class Block:
    def __init__(self, index, previous_hash, timestamp, data, nonce=0):
        self.index = index
        self.previous_hash = previous_hash
        self.timestamp = timestamp
        self.data = data
        self.nonce = nonce
        self.hash = self.calculate_hash()

    def calculate_hash(self):
        block_string = 
f"{self.index}{self.previous_hash}{self.timestamp}{self.data}{self.nonce}"
        return hashlib.sha256(block_string.encode()).hexdigest()

def mine_block(block, difficulty):
    print("\nMining block...")
    start_time = time.time()

    while block.hash[:difficulty] != '0' * difficulty:
        block.nonce += 1
        block.hash = block.calculate_hash()

    end_time = time.time()
    print(f"Block mined: {block.hash}")
    print(f"Mining time: {end_time - start_time:.2f} seconds")
    return block

def create_genesis_block():
    return Block(0, "0", time.time(), "Genesis Block")

def add_block_to_chain(chain, data, difficulty):
    last_block = chain[-1]
    new_block = Block(len(chain), last_block.hash, time.time(), data)
    mined_block = mine_block(new_block, difficulty)
    chain.append(mined_block)

# Initialize blockchain
blockchain = [create_genesis_block()]
difficulty = 4  # Hash must start with 4 zeros
```

```
# Add blocks to the chain
add_block_to_chain(blockchain, "Student A sends 1 BTC to Student B", difficulty)
add_block_to_chain(blockchain, "Student B sends 0.5 BTC to Student C", difficulty)

# Display blockchain
print("\nBlockchain:")
for block in blockchain:
    print(f"Block {block.index}:")
    print(f"Hash: {block.hash}")
    print(f"Previous Hash: {block.previous_hash}")
    print(f"Nonce: {block.nonce}")
    print(f"Data: {block.data}\n")
```



```
Mining block...
Block mined: 00003f43b8f4e446ca08142033fe2417234e0de469e694c4a2d1bee7f9314da8
Mining time: 0.03 seconds

Mining block...
Block mined: 00004848e6edacf30224c5f6a1a3f2000ff47fea6ceb1e05030978e49ea75e8d
Mining time: 0.42 seconds

Blockchain:
Block 0:
Hash: c37c9ec8e8650b54c1fd0161409b4da0a5e0b844ec75407ad45f690880ca519c
Previous Hash: 0
Nonce: 0
Data: Genesis Block

Block 1:
Hash: 00003f43b8f4e446ca08142033fe2417234e0de469e694c4a2d1bee7f9314da8
Previous Hash: c37c9ec8e8650b54c1fd0161409b4da0a5e0b844ec75407ad45f690880ca519c
Nonce: 9805
Data: Student A sends 1 BTC to Student B

Block 2:
Hash: 00004848e6edacf30224c5f6a1a3f2000ff47fea6ceb1e05030978e49ea75e8d
Previous Hash: 00003f43b8f4e446ca08142033fe2417234e0de469e694c4a2d1bee7f9314da8
Nonce: 135525
Data: Student B sends 0.5 BTC to Student C
```

Code Workflow Description

1. **Block Structure:** Each block contains index, previous hash, timestamp, data, nonce, and its own hash.

2. **Mining Process:**

- The mine_block function repeatedly increments the nonce until the block's hash meets the difficulty requirement (leading zeros).
- This simulates the computational work required in real Bitcoin mining.

3. **Blockchain Creation:**

- Starts with a genesis block.
- New blocks are added by mining them with the specified difficulty.
- Each new block references the previous block's hash.

4. **Consensus Simulation:**

- The longest valid chain is implicitly followed as new blocks are only added to the end
- All nodes would accept the chain with the most cumulative proof-of-work

Observations on Mining Time and Difficulty

1. **Difficulty Impact:** Higher difficulty values (more leading zeros required) exponentially increase mining time as more hash computations are needed.

2. **Nonce Behavior:** The nonce increments linearly until a valid hash is found, demonstrating the trial-and-error nature of PoW.

3. **Time Variability:** Mining time varies significantly between blocks even with identical difficulty, mirroring real-world blockchain behavior.

4. **Security Trade-off:** The simulation shows how increasing difficulty enhances security but also increases resource consumption, illustrating Bitcoin's security-vs-efficiency balance.

Conclusion:

This simulation successfully demonstrates the core principles of blockchain technology: decentralized consensus via Proof-of-Work, the immutability of data using hashing, and secure transaction validation through mining.