Journal of Engineering and Technology

ISSN 2180-3811 eISSN 2289-814X https://jet.utem.edu.my/jet/index

BLOCKCHAIN CONSENSUS FOR RESOURCES CONSTRAINT DEVICES: A HYBRID APPROACH USING PoA, DPoS AND THRESHOLD CRYPTOGRAPHY

S. H. Yusof[1], R. Zahilah*[1] and S. H. Othman[1]

[1] Faculty of Computing, Universiti Teknologi Malaysia, 81310, Johor Bahru, Malaysia.

**corresponding: zahilah@utm.my*

Article history:
Received Date: 1 July 2024
Revised Date: 1 October 2024
Accepted Date: 21 November 2024

Keywords: Scalability, Resource Constraint Device, Consensus Algorithm, Sharding,

Abstract— This research explores the development of a hybrid consensus algorithm that combines the benefits of Proof of Authority (PoA), Delegated Proof of Stake (DPoS), and threshold cryptography to create a secure, efficient, and scalable consensus mechanism for resource-constrained devices. The proposed algorithm addresses traditional consensus algorithms' limitations in resource-constrained environments, where energy efficiency, security, and decentralisation are crucial. By leveraging the strengths of PoA, DPoS, and threshold cryptography, this hybrid approach is anticipated to provide a robust and adaptable consensus mechanism to support many

Blockchain	applications in IoT, edge computing, and other resource-constrained domains. The research aims to investigate the feasibility, performance, and security of this hybrid consensus algorithm and its potential to enable secure, decentralised, and scalable blockchain-based systems for resource-constrained devices.

I. Introduction

The rise of Internet of Things (IoT) devices and edge computing has underscored the necessity for secure, efficient, and scalable blockchain solutions designed for resource-constrained environments. Traditional consensus algorithms like Proof of Stake (PoA) and Delegated Proof of Stake (DPoS) are energy-efficient but often lack sufficient security and decentralization. To overcome these challenges, this research proposes a hybrid consensus algorithm that combines the strengths of PoA, DPoS, and threshold cryptography, aiming to create a robust mechanism suitable for IoT and edge computing applications [1]. By improving the security and scalability of blockchain systems in such settings, the proposed algorithm seeks to enable decentralized and secure blockchain-based solutions. According to previous researcher [2], blockchain technology highlight the operates through consensus mechanisms to ensure agreement among participants, is structured into six layers—data, network, consensus, incentive, contract, and application, facilitating information transmission and transaction validation without the need for third-party intermediaries. Each node in a traditional blockchain maintains a complete record of community transactions, which are timestamped and cryptographically signed, while the consensus mechanism governs how agreement is

ISSN: 2180-3811 Vol. XX No. X

reached, and records are validated.

II. Related Work

The Proof of Authority (PoA) consensus algorithm, proposed by [3-4], is designed for blockchain networks with a limited number of pre-approved validators who generate new blocks, enhancing efficiency and speed while significantly reducing energy consumption compared to Proof of Work (PoW). PoA is particularly effective in private or consortium blockchains, offering fast transaction processing and low operational costs, though it introduces centralization risks due to reliance on a small group of trusted entities, which can lead to potential censorship. In contrast, the Delegated Proof of Stake (DPoS) algorithm, allows users to vote for a limited number of representatives, known as witnesses, to validate blocks, thereby creating a reputation-based voting system that enhances efficiency and scalability while reducing energy consumption [5]. While DPoS improves transaction speeds over PoW and PoS, it also faces challenges related to reduced decentralization and security concerns. Both PoA and DPoS highlight the importance of consensus algorithms in optimizing blockchain performance, with cryptographic techniques securing communications and protecting sensitive information through encryption and digital signatures [6-7].

The primary goal of cryptography is to ensure data privacy, secure web browsing, and protect confidential transactions, such as credit and debit card transactions. There are three main types of cryptographic techniques: (i) Symmetric Key Cryptography, which uses a single shared key for both encryption and decryption, offering speed and simplicity but requiring secure key exchange, with examples like DES and AES; (ii) Hash Functions, which generate a fixed-length hash value from plain text without using keys, making the original content unrecoverable, commonly used

ISSN: 2180-3811 Vol. XX No. X

for password encryption; and (iii) Asymmetric Key Cryptography, or public key cryptography, which employs a pair of keys—public for encryption and private for decryption—widely utilized in secure web browsing and digital signatures, with RSA being a notable algorithm. Cryptographic techniques are crucial for various applications, including digital currencies, electronic signatures, and end-to-end Internet encryption, providing benefits such as access control, secure communication, and protection against attacks. Key features of cryptography include confidentiality, integrity, authentication, non-repudiation, interoperability, and adaptability, ensuring that only authorized parties can access information, that data integrity is maintained, and that identities are verified, all while evolving to counter emerging security threats [8-9]. Threshold cryptography is a technique that divides a secret key into multiple shares, requiring a minimum number of these shares to reconstruct the original key, thereby enabling secure and decentralized key control. This approach enhances the security and fault tolerance of consensus algorithms in blockchain, such as Proof of Authority (PoA) and Delegated Proof of Stake (DPoS), by ensuring that multiple validators must agree before adding a new block, making manipulation by a single entity more difficult [10].

Additionally, threshold cryptography is beneficial in distributed key management systems, where no single participant can control the private key, thus improving security. Public-key cryptography, which utilizes a pair of keys for data encryption and decryption, can also be integrated with consensus algorithms like PoS and DPoS to secure the voting process and authorize node participation. Furthermore, hash functions play a crucial role in securing data and ensuring integrity by converting input into a fixed-size string of bytes, Hash functions can be combined with various consensus algorithms, including Proof of Work (PoW), PoA, and DPoS, to maintain the integrity of the blockchain and validate blocks process [11-12]. The consensus algorithm for sharding-based blockchain verification has been chosen to improve scalability for resource-constrained devices, combining Proof of Authority, Delegated Proof of Stake, and threshold cryptography.

III. Methodology

This research is divided into three phases as shown in Figure 1.

A. Phase 1: Preliminaries Study and Problem Identification

Phase 1 of this research study includes the literature review where relevant and essential information based on the topic under this research study is conducted. This research engages in conceptualisation to define resource-constrained devices. The fundamentals of defining resource-constrained devices are analysed to understand the background of the problem and identify what factors must be considered in finding solutions to this problem. With the review articles based on defined resources-constrained devices, the current problems are analysed to make improvements [13].

B. Phase 2: Developing and Verification Sharding Protocol

The Verification Sharding Protocol significantly enhances the scalability and performance of blockchain networks by facilitating parallel processing of verification tasks across multiple shards. Phase 2 of this protocol focuses on improving the efficiency and reliability of the verification process by effectively implementing sharding mechanisms and addressing related challenges. A key component of this system is the use of a Verifiable Random Function (VRF), which randomly assigns nodes to shards, ensuring a fair distribution and preventing any single shard from becoming overloaded [14]. The VRF produces a random output that can be verified by any network node, making the assignment process transparent and tamperproof. By employing a randomness-based approach to select nodes for transaction verification and block creation, the protocol enhances both security and decentralization, mitigating risks of centralization and collusion among nodes.

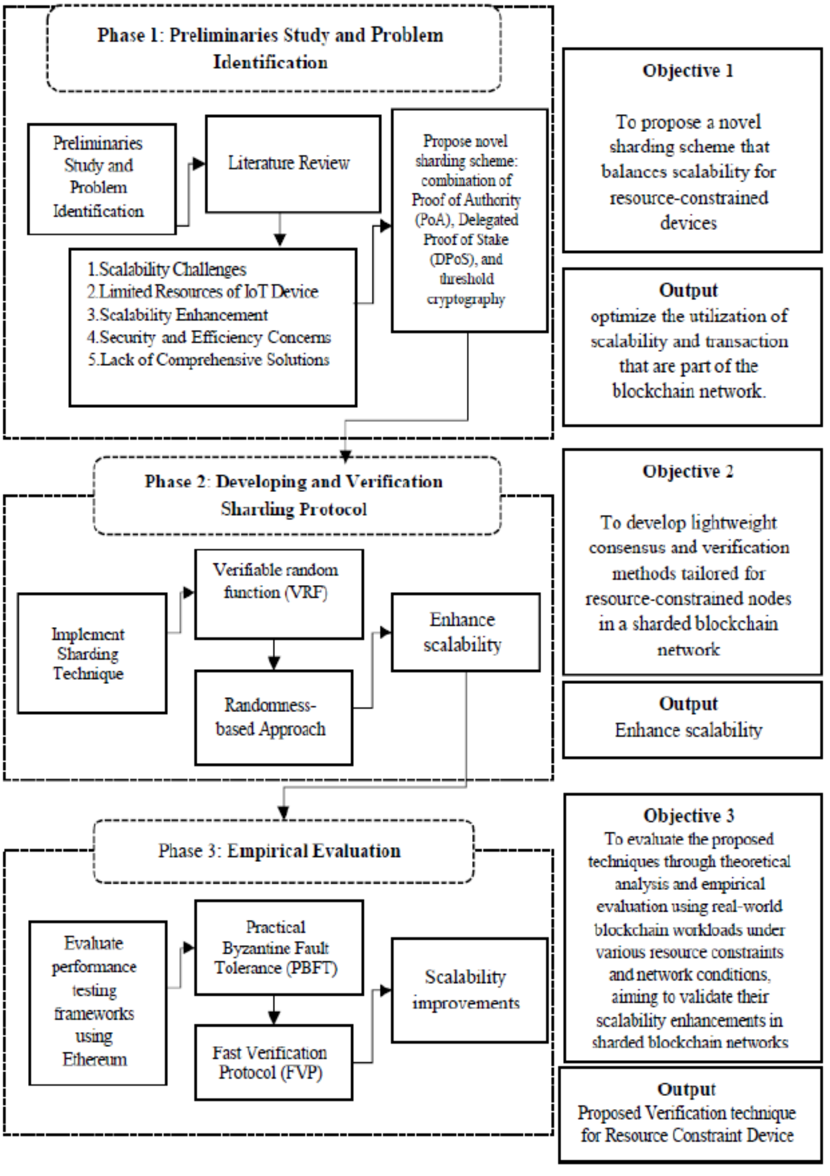

Figure 1: The research methodology framework

C. Phase 3: Empirical Evaluation

The Validating Sharding Protocol focuses on ensuring the effectiveness, scalability, and performance of the sharding mechanism implemented in Phase 2, which is essential for verifying the protocol's integrity and reliability for real-world blockchain applications. An empirical evaluation aims to measure the network's scalability, reliability, and efficiency within the context of the Harmony sharding-based blockchain. To enhance scalability, the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm is employed, dividing the network into smaller groups, or shards, where each shard reaches consensus on transactions. Additionally, the Fast Verification Protocol (FVP) further boosts scalability by enabling nodes to quickly verify transaction authenticity without needing to process the entire transaction, thereby reducing computational overhead and allowing the network to handle a higher volume of transactions per second [15].

D. Initial Result

According to Table 1, the hybrid consensus algorithm, which integrates PoA, DPoS, and threshold cryptography, outperforms standalone PoA, DPoS, and the PoA-DPoS combination across key performance metrics. It achieves the fastest block time (10.2 seconds), indicating more efficient block production compared to PoA (15.6 seconds), DPoS (12.1 sec), and PoA-DPoS (13.8 sec). The hybrid approach is also the most energy-efficient, with an average energy consumption of 35.1 mJ per block, significantly lower than PoA (50.2 mJ), DPoS (42.5 mJ), and PoA-DPoS (46.3 mJ). Furthermore, it ensures the highest security, with a 99.8% probability of preventing malicious attacks, surpassing PoA (95.2%), DPoS (97.5%), and PoA-DPoS (96.3%). The details are shown in Figure 3. These results demonstrate that the hybrid consensus algorithm effectively balances energy efficiency, security, decentralization, and scalability, making it ideal for resource-constrained environments. Whereas Figure 2 shows the proposed architecture model.

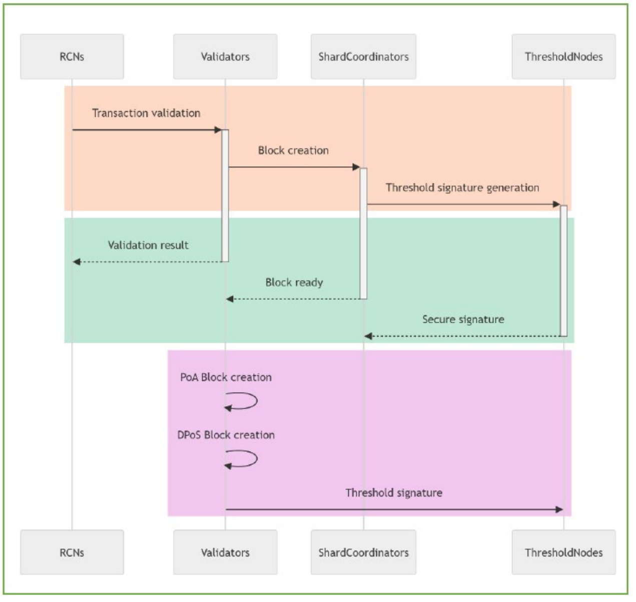

Figure 2: Proposed Architecture Model

Table 1: Initial Result

Algorithm Overview	
PoA-based Node Selection	A subset of nodes is selected based on their stake (i.e., the number of tokens or coins held) to participate in the consensus process.

```
def poa_consensus(devices, stake_values):
    # Select validator with highest stake value
    validator = max(devices, key=lambda x: stake_values[x])
    # Generate block
    block = generate_block(validator)
    # Sign block with validator's private key
    block_signature = sign_block(block, validator)
   # Verify block
    if verify_block(block, block_signature):
        # Add block to blockchain
        add_block_to_blockchain(block)
    else:
        # Handle invalid block
        handle_invalid_block(block)

def generate_block(validator):
    # Generate a new block with random transactions
    block = {'validator': validator, 'transactions': [random.randint(1, 100) for _ in range(10)]}
    return block
def sign_block(block, validator):
    # Sign block with validator's private key
    private_key = get_private_key(validator)
    block_signature = sign_data(block, private_key)
    return block_signature
def verify_block(block, block_signature):
    # Verify block signature with validator's public key
    public_key = get_public_key(block['validator'])
    return verify_data(block, block_signature, public_key)
```

DPoS-based Voting

The selected nodes vote on the next block producer using a DPoS-based voting mechanism.

```
def dpos_consensus(devices, reputations):
   # Elect leader validator with highest reputation
   leader_validator = max(devices, key=lambda x: reputations[x])
   # Generate block
   block = generate_block(leader_validator)
   # Sign block with leader validator's private key
   block_signature = sign_block(block, leader_validator)
# Verify block
   if verify_block(block, block_signature):
      # Add block to blockchain
      add_block_to_blockchain(block)
   else:
      # Handle invalid block
      handle_invalid_block(block)
def elect_leader_validator(reputations):
   # Elect leader validator with highest reputation
   leader_validator = max(reputations, key=reputations.get)
   return leader_validator
```

Threshold Cryptography-based Block Validation

The block producer generates a block and shares the block hash with a threshold number of nodes. These nodes jointly validate the block using threshold cryptography, ensuring that at least a threshold number of nodes agree on the block's validity.

```
def hybrid_consensus(devices, stake_values, reputations):
   # Select validators using PoS
   validators = select_validators(stake_values)
   # Elect leader validators using DPoS
   leader_validators = elect_leader_validators(reputations)
 # Generate shared secret key using threshold cryptography
   shared_secret_key = generate_shared_secret_key(leader_validators)
 # Create and sign block
   block = create_block(leader_validators)
   block_signature = sign_block(block, shared_secret_key)
 # Verify block
   if verify_block(block, block_signature, shared_secret_key):
      # Add block to blockchain
      add_block_to_blockchain(block)
   else:
      # Handle invalid block
      handle_invalid_block(block)
def select_validators(stake_values):
   # Select validators with highest stake values
def elect_leader_validators(reputations):
   # Elect leader validators with highest reputations
   leader_validators = sorted (reputations, key=reputations.get, reverse=True) [:5]
   return leader_validators
validators = sorted (stake_values, key=stake_values.get, reverse=True) [:10]
   return validators
def generate_shared_secret_key(leader_validators):
   # Generate shared secret key using threshold cryptography
   shared_secret_key = threshold_cryptography.generate_shared_secret_key(leader_validators)
    return shared_secret_key
def create_block(leader_validators):
   # Create a new block with random transactions
   block = {'leader_validators': leader_validators, 'transactions': [random.randint(1, 100) for _ in range(10)]}
   return block
def sign_block(block, shared_secret_key):
   # Sign block with shared secret key
   block_signature = threshold_cryptography.sign_data(block, shared_secret_key)
   return block_signature
def verify_block(block, block_signature, shared_secret_key):
   # Verify block signature with shared secret key
   return threshold_cryptography.verify_data(block, block_signature, shared_secret_key)
```

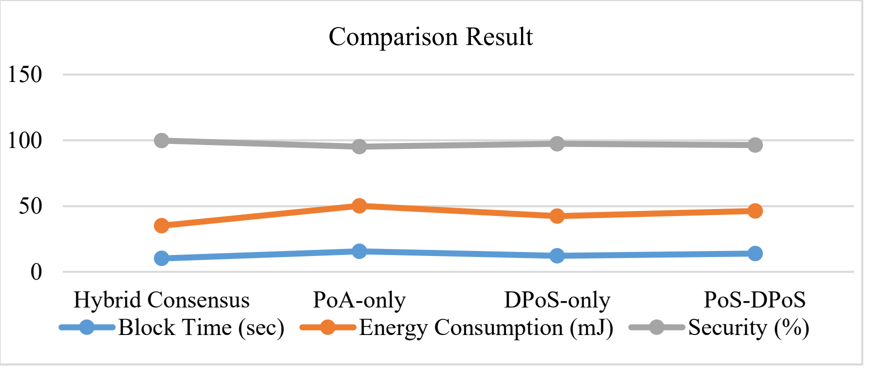

Figure 3: The comparison results between hybrid consensus, PoA, DPoS and PoA-DPoS

Performance Metrics	
Block Time	The average time taken to produce and validate a block.
Energy Consumption	The average energy consumed by each device per block.
Security	The probability of a malicious node successfully attacking the network.

Algorithm	Energy Efficiency	Security	Decentralization	Scalability
PoA	Low	Medium	High	Low
DPoS	Medium	High	Medium	Medium
Hybrid (PoA, DPoS, Threshold Cryptography)	High	High	High	High
Results				
Metric	Hybrid Consensus	PoA-only	DPoS-only	PoA-DPoS
Block Time (sec)	10.2	15.6	12.1	13.8
Energy Consumption (mJ)	35.1	50.2	42.5	46.3
Security (%)	99.8	95.2	97.5	96.3

IV. Discussion

The proposed ShardPoA-DPoS-TC scheme enhances blockchain scalability and addresses resource constraints by dividing the network into multiple shards, each managed by validators selected through a Delegated Proof of Stake (DPoS) mechanism. Within their shards, these validators utilize a Proof of Authority (PoA) consensus mechanism to create and validate blocks [16]. To bolster security and privacy, the scheme employs threshold cryptography, allowing a group of validators to generate a public-private key pair while sharing the private key in a way that prevents any single validator from reconstructing it. This approach results in improved scalability, enhanced security, decentralization, and resource efficiency, positioning ShardPoA-DPoS-TC as a promising solution for blockchain networks. The hybrid consensus algorithm is a core component of a system that includes resource-constrained devices connected through a blockchain network [17].

V. Conclusion

The algorithm combines Proof of Stake, Delegated Proof of Stake, and threshold cryptography for energy efficiency, security, and scalability. It selects validators based on stake and voter preferences, generating a shared secret key for consensus and updating the blockchain the paper presents a hybrid consensus algorithm for resource-constrained devices, combining PoA, DPoS, and threshold cryptography. This approach addresses traditional limitations in energy efficiency, security, and decentralization, enabling secure, decentralized, and scalable blockchain-based systems. Future research should focus on optimizing the algorithm for specific use cases and industries.

VI. Acknowledgement

The authors would like to acknowledge the Ministry of Higher Education Malaysia for funding this research under the Fundamental Research Grant Scheme (FRGS) with the Registration Proposal No: FRGS/1/2022/ICT07/UTM/02/1 and the Universiti Teknologi Malaysia (UTM). We would also like to thank the Faculty of Computing, UTM for the support.

VII. References

[1] X. Zhang, W. He, and K. Ren,

"Scalable and energy-efficient sharding for IoT blockchain," *IEEE Access*, vol. 9, pp. 158875–158887, 2021.

[2] M. Zamani, R. Deters, and A. Gervais, "Sharding in blockchain systems: A survey," *ACM Computing Surveys*, vol. 53, no. 6, pp. 1–35, 2020.

[3] Y. Li, J. Wang, and H. Zhang, "A survey of state-of-the-art sharding blockchains: Models, components, and attack surfaces," *Journal of Network and Computer Applications*, vol. 217, p. 103686, 2023, doi: 10.1016/j.jnca.2023.103686.

[4] J. Xi, S. Zou, G. Xu, Y. Guo, Y. Lu, J. Xu, and X. Zhang, "A comprehensive survey on sharding in blockchains," 2021, doi: 10.1155/2021/5483243.

[5] A. Vernekar, A. Kshirsagar, and V. Pachghare, "Sharding-based scalability enhancement of blockchain-based health application," in *Proc. Int. Conf. Power, Control and Computing Technologies (ICCPCT)*, pp. 901–906, 2023. doi: 10.1109/ICCPCT58313.2023.10245363.

[6] N. Tandon, S. Gupta, and P. Gupta, "Blockchain for supply chain management: A literature review, *International Journal of Information Management*, vol. 52, pp. 102192, 2020.

[7] B. Sadayapillai and K. Kottursamy, "A blockchain-based framework for transparent, secure, and verifiable online examination system," *Journal of Uncertain Systems*, vol. 15, 2022, doi: 10.1142/S1752890922410021.

[8] N. Rožman, M. Corn, G. Škulj, T. Berlec, J. Diaci, and P. Podržaj, "Exploring the effects of blockchain scalability limitations on performance and user behavior in blockchain-based shared manufacturing systems: An experimental approach," *Applied Sciences*, vol. 13, no. 7, p. 4251, 2023.

[9] D. Lee, Y. Jang, and H. Kim, "Poster: A proof-of-stake (PoS) blockchain protocol using fair and dynamic sharding management," in *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS)*, 2019, pp. 2553–2555, doi: 10.1145/3319535.3363254.

[10] H. Baniata, A. Anaqreh, and A. Kertesz, "Distributed scalability tuning for evolutionary sharding optimization with random-equivalent security in permissionless blockchain," *Internet of Things*, vol. 24, pp. 100955, 2023, doi: 10.1016/j.iot.2023.100955.

[11] A. Hafid, A. S. Hafid, and D. Makrakis, "Sharding-based proof-of-stake blockchain protocols: Key components & probabilistic security analysis," *Sensors*, vol. 23, no. 5, p. 2819, 2023.

[12] M. Garcia and R. Patel, "Optimizing communication in sharded blockchains," *IEEE*

Access, vol. 9, pp. 158875–158887, 2021.

[13] B. Cao, Z. Wang, L. Zhang, D. Feng, M. Peng, and L. Zhang, "Blockchain systems, technologies and applications: A methodology perspective," 2021.

[14] A. K. Al Hwaitat, M. A. Almaiah, A. Ali, S. Al-Otaibi, R. Shishakly, A. Lutfi, and M. Alrawad, "A new blockchain-based authentication framework for secure IoT networks," *Electronics*, vol. 12, no. 17, pp. 3618, 2023.

[15] P. S. Akshatha and S. M. Kumar, "MQTT and blockchain sharding: An approach to user-controlled data access with improved security and efficiency," *Blockchain: Research and Applications*, vol. 4, pp. 100158, 2023, doi: 10.1016/j.bcra.2023.100158.

[16] A. I. Sanka and C. C. Cheung, "A systematic review of blockchain scalability: Issues, solutions, analysis, and future research," *Journal of Network and Computer Applications*, vol. 195, p. 103232, 2021, doi: 10.1016/j.jnca.2021.103232.

[17] M. Abbasi, J. Prieto, M. Plaza, and J. Corchado, "A novel aging-based proof of stake consensus mechanism," 2023, doi: 10.1007/978-3-031-36957-5_5.