Manpreet Kaur
Shikha Gupta

PERFORMANCE EVALUATION OF A LIGHTWEIGHT CONSENSUS PROTOCOL FOR BLOCKCHAIN IN IoT NETWORKS

Abstract *The consensus protocol is essential in practically every blockchain application. Most of these existing blockchain consensus protocols need massive computational capabilities, substantial energy consumption, and dependency on monetary stakes. These shortcomings in the mainstream consensus approach lead to their unsuitability for low-resource applications like IoT. As a result of this work, a lightweight consensus process referred as Delegated Proof of Accessibility (DPoAC) is implemented and evaluated. DPoAC makes use of Shamir secret sharing, Proof of Stake (PoS) with random selection, and the Inter-Planetary File System (IPFS). The DPoAC operation is composed of four modules: secret generation and distribution, retrieval of secret shares, block creation and verification, and block rewards and penalty. A detailed description of DPoAC has been provided and implemented in JavaScript and experimental results demonstrate that our solution meets the necessary performance and security requirements for a lightweight scalable protocol for IoT systems.*

Keywords blockchain, consensus protocols, DPoAC, secret sharing, IPFS, IoT

Citation Computer Science 26(1) 2025: 27–47

1. Introduction

Blockchain is made up of a number of interconnected blocks that hold an extensive list of transactional information, much like a conventional ledger. The use of cryptographic hash algorithms to store records in these blocks is a notable characteristic that distinguishes blockchain from other traditional approaches. Blockchain is a decentralized public database that groups cryptographically signed transactions into blocks [20]. Following validation and consensus, every block is connected to the block before it to make it temper-proof. As new blocks are added to the blockchain, it becomes progressively harder to change the earlier ones. When a new block is inserted into the blockchain, a broadcast is made to the network to update the local copies of the blockchain retained by the participating nodes [14, 27]. Blockchain offers trust in two-fold. First, the committed transaction records are immutable due to the block-linking method, ensuring that the data in the chain cannot be altered. Secondly, the reliability of the data being entered into the system by P2P nodes is governed by the consensus process. Every piece of information must get the consent of a majority of participants to get included in a block, as guaranteed by consensus [11]. Being a distributed structure, blockchain maintains the integrity of data blocks using consensus protocol. In the past various consensus, algorithms have been developed and deployed in specific applications. Unfortunately, no such algorithm is without flaws. Most contemporary blockchain systems are unable to meet the requirements of any substantial real-world application because of severe restrictions set by privacy, security, and throughput. Many of these limitations are the consequence of difficulties that arise as an outcome of underlying consensus. Hence, consensus methods have played a significant role in the creation of more useful blockchain networks [13, 16].

Proof of Work (PoW), Proof of Stake (PoS), Proof of Activity (PoA), and Delegated Proof of Stake (DPoS) are well-known consensus techniques that have been studied thoroughly in the blockchain domain by researchers [16, 18]. PoW forces participants to resolve a complex computing problem, to increase the difficulty and cost of the process [16]. Many PoW variants have been adopted, including PoS, which requires that a node must deposit a certain quantity of coins in order to be granted block creation privileges [16, 18]. Hence, a node with a high stake is more likely to gain block creation privileges over the other nodes with relatively low stakes [16].

Other variations, such as DPoS and PoA, exist in relation to these widely used consensus methods. DPoS outperforms PoS in terms of efficiency and speed [16]. A group of nodes are chosen to serve as delegates in the network through an election process including stakeholders. These delegators will produce new blocks based on a predefined sequence, and a delegate will be eliminated if it does not produce a block in its turn [16]. DPoS is a more affordable and powerful consensus method than PoW and PoS [16, 18]. Unfortunately, voting could not prevent malicious entities from being selected, especially in small networks, and may pose certain security risks. Miners may typically lose interest in the mining process as the bitcoin reward is reduced by half after every 210,000 blocks mined. As a result, miners will demand a hefty transaction fee for the computational resources used in the mining process. As a result, a hybrid solution known as Proof of Activity (PoA) has been devised, which is a combination of PoW and PoS. As this approach is hybrid in nature, it is more complex to implement and demands heavy computations due to the inclusion of PoW [16].

IoT applications have evolved over time and impose a significant impact on our day-to-day lives. However, with the rapid growth in the deployment of IoT systems, severe issues in the context of scalability and security have been uncovered. The processing power and computational capabilities of current IoT devices are limited. Blockchain technology may be considered as an optimal solution for supplementing these limitations of IoT devices. Blockchain, being a decentralized architecture, reduces the likelihood of center point failure and increases security [16]. Scalability would be increased since data is spread across the network instead of being retained on a single server. The current mainstream consensus approaches, on the other hand, have substantial drawbacks, such as low efficiency, huge power consumption, and increased resource demands. Blockchain applications are impractical due to these limitations, particularly in the IoT/IIoT ecosystem [16, 17]. Due to its significant resource requirements, PoW is particularly unsuitable for resource-constrained applications such as IoT/IIoT [16]. The main barrier preventing PoS approach from being used in IoT networks is the requirement of a stake in cryptocurrency terms, which is irrelevant in IoT networks [7, 11]. Similarly, the key limitation of DPoS in IoT contexts is its dependence on financial stakes to choose delegates [16, 24]. Although the PoA technique is less susceptible to malicious threats, prolonged latencies may arise, that might be undesirable for time-critical IoT applications [6, 16].

These concerns lead us to create a new consensus mechanism that inherits the benefits of existing consensus methods and addresses their shortcomings, particularly in IoT. Therefore, through this paper, we implement Delegated Proof of Accessibility (DPoAC), a new consensus protocol that combines two distinct strategies, namely Proof of Accessibility and Proof of Stake with Randomized Selection. By combining the characteristics of traditional Proof of Accuracy and PoS algorithms with modifications, DPoAC builds on the idea of a hybrid algorithm. Proof-of-accuracy consensus techniques have received minimal research in past. The Proof of Accuracy Protocol [19] selects a coordinator, generates a secret, and distributes it across multiple nodes by separating it into secret components. The willing nodes must compete for access to those nodes in order to acquire the secret components and rebuild the secret. The node that has acquired the capacity to regenerate secrets will be designated as the block generator for new blocks. This study was mostly conceptual, with no practical implementations. The key contributions of this paper are mentioned below.

Although the paper [25] introduced the idea of DPoAC, no specific implementation was provided; only a qualitative analysis was included. Through this work, we have provided comprehensive algorithms to demonstrate the operation of DPoAC.

We implemented the DPoAC protocol and assessed its performance quantitatively against the comprehensive performance matrix. In terms of scalability, security, fault tolerance, overhead, and latency, DPoAC outperforms mainstream consensus protocols.

Paper structure

The remainder of the paper is arranged as below. Section 2 discusses the background and related works. The detailed description and operation of DPoAC are included in Section 3. Section 4 evaluates the performance and provides a performance comparison of DPoAC. At last, Section 5 concludes this research article and outlines future work.

2. Background and related work

2.1. Shamir Secret Sharing (SSS) Scheme

The SSS technique is a distributed method of preserving secrets, particularly cryptographic keys. A secret is divided into many portions called shares [16]. These individual shares are used to piece together the original secret again. A fixed number of shares are necessary to discover the secret using SSS. The threshold represents the minimal amount of shares necessary to disclose the secret [10, 16]. To further comprehend the SSS scheme, consider a secret S that has been broken into n different portions, $S = S_1, S_2, ...S_n$ in a way that satisfy following requirements [16, 25]:

- To regenerate the original secret S the information of a minimum of k distinct secret portions are necessary.
- The information of only $k - 1$ or fewer secret portions cannot determine S.

This approach is known as the (k, n)-threshold scheme because it requires just k data pieces from total n to recover the original secret.

2.2. IPFS

IPFS (Inter-Planetary File System) is a peer-to-peer distributed file system that could be employed instead of HTTP [5, 16]. Unlike HTTP, IPFS employs content-based indexing; when a file is included in the system, it is divided into 256-byte chunks. These chunks include object data and linkages that will be preserved in a Merkel DAG [15]. The system offers one hash value termed as the basic content identifier (CID) [16] to obtain the file from IPFS. The Distributed Hash Table (DHT) is used on IPFS to store information; the distributed option enables the latest hash tables to be made accessible across many places [3, 16]. DHTs are generally used to record and retain information in P2P networks. It is a decentralized solution that uses a hash table-like lookup structure to retain index-value pairs, and users may efficiently access the content associated with a certain index. DHT supports content discovery, network routing, and peer discovery in IPFS [16]. The content-based indexing employed in IPFS makes it a more accessible and stable option for integrating with the blockchain.

2.3. Blockchain-IoT integration

The IoT primarily consists of hardware components, gateways, fog/edge nodes, clouds, and the internet. IoT enables edge devices and Cloud platforms to understand and store data by allowing physical entities to broadcast data over a gateway [7]. These devices could follow instructions to perform a specific behavior from one another over the Cloud. A central manager for the Cloud may also issue commands. The IoT stack and standard protocols collaborate to offer architectural layers that provide services to hardware objects in the IoT network. Currently, centralized architecture is used in the vast majority of IoT solutions [7]. But IoT systems suffer major limitations due to the reliance on central servers. The heterogeneity of IoT devices in use presents a number of difficulties in the context of security, privacy, and interoperability. Furthermore, as more devices are added to an IoT system, network management becomes more complicated, resulting in poor scalability. Server failure, resource- constraint nature of IoT devices, privacy concerns and large-scale data management are some other challenges that draw attention while considering IoT applications. Blockchain offers a P2P network where all network devices share memory and computational resources. The cost of establishing and maintaining clouds, data centers, and networking hardware could thereby reduce. Distributed communication architecture can help solve the problem of a single point of failure [16, 22]. As a significant component of blockchain, cryptographic algorithms enable this framework to have intrinsic security and privacy protections in IoT networks. Additionally, blockchain is able to address problems with data integrity brought on by IoT devices since distributed ledgers are irreversible in nature [4, 16, 17]. Despite the fact that blockchain has built-in security, data integrity, and a lack of central authority, implementing blockchain in resource-constrained IoT networks is exceedingly difficult due to inefficient consensus mechanism, big data storage and high througput needs of IoT systems [16, 17].

Kudin et al. [19] suggested Proof of Accuracy (PoAc) as a theoretical concept only. PoAc relies on evidence of access to the input data required to solve the given problem as well as the provision of a solution to a problem with a certain computational complexity threshold. Practical specifications were not provided; only a theoretical concept was conveyed.

Naz et al. [23] introduced an IPFS-based data-sharing and digital asset-distributed platform using blockchain. This solution increases security and access control by performing authorization operations through a smart contract written by the owner. The suggested solution was implemented on an Ethereum private blockchain. Due to the encryption imposed by the Shamir Secret Sharing algorithm used to hash IPFS data hashes, clients with pending debts for online content have been prohibited from data access. This protects the owner from any unauthorized access to the hash. Users may be able to search for and publish reviews using the smart contract-based review system. Results from simulations have been used to assess the economic feasibility and energy usage of the introduced system.

Zhang et al. [28] proposed a unique hierarchical threshold secret-sharing strategy based on blockchain technology. The secret is available to any approved subgroup of system users, and private shares are spread across various levels of system users. Smart contracts were developed to identify illegal actions and secure the integrity of the secret-sharing mechanism. If users do not honestly follow the regulations, dishonest conduct may be uncovered, resulting in a financial penalty. Finally, participants may duplicate the secret fairly despite the lack of a central authority.

Liang et al. [21] created a secret-based robust data transportation system based on SSS to transfer data across trade centers with the help of secret sharing. Data is stored in a flexible, connected storage system using a consensus approach. Unfortunately, this technique was meant for specific applications, and the importance of power data security was overlooked.

There are various vulnerabilities that can harm online data housed on a central system controlled by a single party. Masayuki et al. [8] presented secure storage that does not require a central server. Unauthorized access to private information has been prevented. The information of each user has been segmented using a hidden sharing scheme. These portions are stored on separate network nodes. By obscuring the most crucial properties of data, it is converted into metadata. By exploring for nodes in the network that hold the data segments, a user may reconstruct the original data. This proposed method was reliable since it enables a user to find desired data even though the network architecture changes. Furthermore, other nodes in the network may discover a fraudulent node via majority rule consensus. But there was no quantitative evaluation of system security.

Geng et al. [9] introduced an improved consensus approach for a broad blockchain network that incorporates a verified secret-sharing technique. Verifiable secret sharing protects privacy, and secure multi-party computing was used to boost privacy, effectiveness, and equality.

Zhou et al. [29] investigated the privacy-protection aspects related to permissioned blockchain in the context of multi-party computing. In this effort, a secure MPC protocol was incorporated into Hyperledger Fabric. Secret sharing, homomorphic encryption, and zero-knowledge proof were all used in the proposed protocol.

Andrian et al. [1] demonstrated how IPFS may be used to increase accessibility and efficiency by sharing data across multiple IPFS nodes. The rate of data flow and status for nodes are provided through a real-time monitoring system. Experimental results showed that the IPFS-based system speeds up throughput and minimizes file replication time when compared to the performance of the existing version.

Aponte-Novoa et al. [2] developed a detailed design and implementation of proof-of-accuracy consensus through this work. To control miner computer power and minimize majority threats, authors focused on democratizing miner involvement in a blockchain. The suggested system has provided the results of simulations conducted in Python.

Ullah et al. [26] developed an IoTChain model by employing IPFS, Ethereum, AES encryption scheme, and Proof of Authority (PoA) consensus mechanism. To store data produced by IoT devices IPFS based blockchain-enabled system was designed. The simulation results were analyzed to examine the performance and transaction expenses.

Kara et al. [12] introduced a novel Proof of Chance (PoCh) consensus method that reduced the need for extensive resources in the mining process. The algorithm chooses a block producer depending on the value of the chance parameter and the target value. The performance of PoCh against different criteria in the IIoT context was supported by the experimental findings.

3. Proposed consensus protocol

Consensus protocol as a vital element of any blockchain application is influenced by a number of threats including huge resource needs and energy usage which restricts the deployment of blockchain in several domains. Due to restricted resources, IoT/IIoT applications cannot employ these high-cost consensus procedures. Delegated Proof of Accessibility (DPoAC) is therefore offered as a novel consensus process that employs the Shamir secret sharing approach, modified PoS with random selection, and IPFS. DPoAC operates in two phases. P2P nodes are sorted according to their accumulated reputation stake. During the early phase, one node is selected randomly among the top P2P nodes as a super node for each round. A secret is generated and divided into n shares by a randomly selected super-node. These shares are encrypted using the RSA encryption algorithm and saved on distinguished IPFS nodes. To reassemble the secret, the nodes will contend for access to these shareholders. The winning node will receive block-generating rights. The appropriate hash value is calculated and a block with legitimate transactions is built in the second phase using modified PoS with random selection. A node holding a high currency stake can win mining privileges over other nodes with a smaller stake in a conventional PoS system. But the stake in currency terms is not applicable for IoT nodes therefore we replaced this monetary stake with the reputation stake. In this customized approach, a node has to prove its reputational stake to the super node in order to secure block creation rights. Moreover, for every successful mining, the winner node is entitled to reputation coins that would increase its likelihood to gain block generation privileges and even the chance to become a super node for future blocks. With this innovative approach, a node with little computing capabilities and minimal monetary stake may still secure block generation rights by demonstrating access to secret shares and reconstructing the secret, which makes the system relatively equitable [16].

3.1. Design of DPoAC

DPoAC design consists of five basic entities, and the DPoAC process is divided into four modules explained below. A list of abbreviations used in various algorithms is shown in Table 1.

Table 1
List of abbreviations and notations

Symbol	Description
F_p	A finite field of integer modulo p
$p, p_1, p_2 \in N$	Prime numbers
n	Number of secret shares
k	Threshold shares
S	Secret generated
$i, j \in N$	Identifiers
$S_1, S_2, ...S_n$	Secret shares
$S_{1e}, S_{2e}, ...S_{ne}$	Encrypted shares
H(S)	Hash value of secret S
CID_i	Content ID for i^{th} secret share

3.1.1. Entities

1. Delegated Super Node: It is a node with a specified reputation stake that is chosen at random out of a group of delegates to begin the process of granting block generation permissions to other nodes [16].
2. P2P Nodes: It is a node in a peer-to-peer network that maintains a private-public key pair as well as extra metadata [16].
3. Miner Nodes: A node with particular interests and capacity to reveal the secret number created by the authorized super node within a certain time frame [16].
4. Secret Shareholders: These peer-to-peer (P2P) nodes will be employed to keep the system's secret shares and will be obliged to release these shares at the demand of peers who have been proven to be authentic [16].
5. Forger Node: This is the winner node, which has successfully reproduced the secret and is entitled to block creation privileges [16].

3.1.2. Modules

1. Secret Generation and Distribution [16]: A delegated super node generates a random secret number S, and its hash, which is concealed from P2P participants, is computed as $H(S)$ and preserved using a (k, n)-threshold cryptographic scheme such as Shamir secret sharing protocol. After then, the secret S needs to be divided into n shares or partitions. Following encryption, those shares, together with any accompanying metadata, are recorded on diverse stakeholders via the IPFS protocol. $H(S)$ will be sent to all other nodes.
2. Secret Shares Retrieval [16]: To reveal this secret, at least k shares out of n must be collected. Therefore, miner nodes make contact with IPFS nodes that hold the secret shares. The first node which has successfully retrieved and accumulated k shares will divulge and prove this secret to the delegated super node. By comparing the revealed secret to $H(S)$, the remaining nodes may easily certify that secret shares had been accessed and the correct secret had been regenerated.

3. Block Creation and Verification [16]: A node will be given the ability to create blocks after it has demonstrated that it has gained access to all shareholders and disclosed the exact secret created by the super node. We refer to this node as a "forger node" at this point. Using its secret key, the forger must compute an encrypted value from the hash of the preceding block. This encrypted data is then hashed, and the first 64 bits of the resulting hash are referred to as the "hit value". The inclusion of a secret key in the preceding computation ensures that a forger obtains a distinct hit value. Forging is assigned to a node that returns a "hit value" smaller than a "target value" [18]. The target value is calculated using Equation (1).

$$Target\,value = T_b \cdot L \cdot R_e \tag{1}$$

   where T_b = "base target value" = preceding block target value, L = the amount of time consumed by the last block, R_e = total reputation coins accumulated or staked [16, 18]. When a block is created, it must be sent to all P2P nodes for confirmation. If greater than 50% of the nodes in the network confirm the block, it will be included in the blockchain [16].
4. Block Rewards and Penalty [16]: If a forger successfully constructs a block that is accepted by a large number of P2P nodes, both the forger and the super node get compensated with reputation coins in an 80:20 ratio. However, if a forger tries to make a false block, the staked reputation coins are lost, and the forger is required to wait a certain period of time before participating in the next block formation round. This method will keep the system safe from malicious attacks.

3.2. Working of DPoAC

When there are transactions in the transaction pool, a super node is randomly selected from the list of delegators. Using the Shamir secret sharing mechanism, a chosen super node will generate the secret number S. The hash value for this secret number will then be computed by the super node and stored as $H(S)$ [16]. The process of secret generation and reconstruction is shown in Figure 1.

Algorithm 1 demonstrates the detailed process of the generation of secret shares and $H(S)$. We utilized IPFS to store and access these shares, and IPFS is based on content addressing, so anyone with a given CID (Content Identifier) may access the data. However, any node with a CID can access the data but not modify it since minor changes in the data would result in a new CID (data de-duplication). Transport encryption is provided by IPFS as an intrinsic feature to prevent third-party/malicious user eavesdropping when data is being transported from one node to another. However, content encryption is missing that secures the data even if it has been accessed by fraudulent users. Therefore, content encryption is essential before storing the data on IPFS.

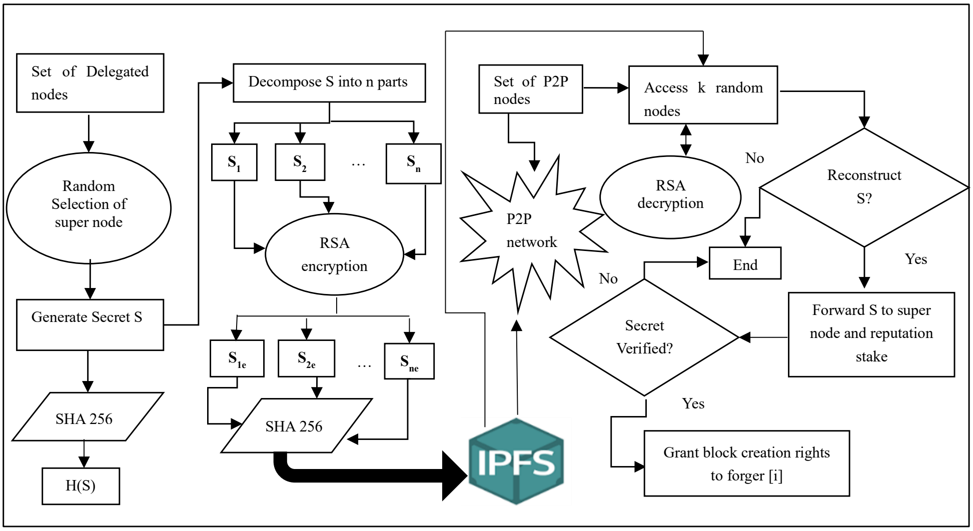

Figure 1. Secret generation and reconstruction

Algorithm 1: Secret generation

```
Data: $F_p$ is a finite field and $p$ is a prime number, Secret $S \in F_p$, threshold
      value $k$, number of secret shares $n$, large prime numbers $p_1$,$p_2$
Result: Secret Shares ($S_1$,$S_2$,...$S_n$), H(S)
$f_0 \leftarrow S$ ;                                   /* S is a Secret value */
$H(S) \leftarrow SHA256(S)$ ;                          /* Apply SHA-256 */
Choose Random coefficients $a_1,a_2,a_3,...a_{k-1} \in F_p$;
Construct polynomial f($x$)=$a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + .... + a_2x^2 + a_1x^1 + S$;
Evaluate the polynomial for $i$= 1 to $n$ do
    Select $x_i$ randomly from $F_p$ s.t. $x_i \neq x_j$ ∀ j ∈ $(1, 2, 3, ...i - 1)$;
    Compute $y_i$ = f($x_i$);
    $S_i$ =($x_i$,$y_i$)
end
```

Algorithm 2 presented the encryption scheme RSA used by our proposed consensus approach to encrypt each S_i to corresponding encrypted value S_{ie}. Algorithm 3 computes hash value of every encrypted share as $CID_i = H(S_{ie})$ and these shares must be stored on at least n distinct nodes using IPFS [16]. In addition, CIDs of secret shareholders nodes are broadcasted to the P2P network with the necessary keys to access these shares.

Algorithm 2: Secret Shares Encryption using RSA algorithm

```
Data: Set of secret shares M = ($S_1$,$S_2$,... $S_n$) and two large prime numbers
      $p_1$,$p_2$.
Result: Encrypted Shares
Compute $Pb_{key_n}$=p × q;
Calculate $\phi(Pb_{key_n})$=($p_1$-1)×($p_2$-1);
Choose an integer e s.t. $1 \leq e \leq \phi$ ($Pb_{key_n}$) and gcd(e, $\phi(Pb_{key_n})$)=1;
Compute d=$e^{-1}$ $\phi$ ($Pb_{key_n}$);
Return ($\phi$ ($Pb_{key_n}$),e,d);
for $i$ =1 to n in M do
    $S_{ie}$=$S_i^e$(mod($Pb_{key_n}$));          /* Get encrypted secret shares */
end
Return($S_{1e}$,$S_{2e}$,...,$S_{ne}$) ;                         /* Encrypted Shares */
```

Algorithm 3: Distribution of encrypted shares on IPFS

```
Data: Set of encrypted secret shares
Result: Set of share-holder IPFS nodes ($CID_i = (x_1, x_2)$,$CID_2$,...$CID_n$)
for $i$ =1 to n in M do
    $CID_i$=SHA256($S_{ie}$);                              /* Apply SHA-256 */
end
shares-ipfs= ($CID_i$=($x_i$,$y_i$));          /* Store each encrypted share on
 randomly chosen IPFS nodes */
for $i$= 1 to n do
    $CID_i$;                              /* Broadcast to all IPFS nodes */
    H(S) ;
    threshold k ;
    $public - key \rightarrow Pb_{key_n} and\ e$
end
```

Algorithm 4 exhibited the process through functions share_retrieval and secret_-recreate. Here, the miner nodes must get access to more than or equal to k secret shareholders and retrieve S_{ke} encrypted secret shares. Therefore, after getting access to k-encrypted secret shares, decryption needs to be performed by keys supplied through the network. Function secret_recreate is utilized to recreate the secret S'. The miner will compute S' as $H(S')$ and compare it to the transmitted value $H(S)$. If it matches, the miner will send both the secret number S' and the staked reputation coins to the super node. The transmitted secret and reputation value are confirmed by the super node, and if confirmed, that miner is granted block-generating privileges and designated as the forger [16].

Algorithm 4: Secret Share Retrieval and Secret Reconstruction

```
Data: Set of share-holder IPFS nodes ($CID_1$,$CID_2$,...$CID_n$), H(S),
      threshold k, super-node-id
Result: Miner-address, Secret S, reputation-stake-coins
/* Function share_retrieval(miner address)                    */
$share - accessed \leftarrow 0$ ;
$S \leftarrow 0$ ;
for $j$= 1 to $k$ do
    i ← 1 if node get-access to IPFS node with $CID_i$ then
        share-accessed= map (($CID_i$) → key.decrypt( $CID_i$ ));
    else
        i ← i+1 ;
    end
    Return share-accessed ($x_i$,$y_i$)
end
/* Function secret_recreate (share-accessed ($x_1$,$y_1$))     */
S ← 0 $\epsilon F_p$ ;
for $i$ = 1 to $k$ do
    $\delta_i = 1 \epsilon F_p$ for $j$ = 1 to $k$ do
        if $i \neq j$ then
            $\delta_i = \delta_i \ . \ \frac{-x_j}{x_i - x_j}$;
        end
    end
    S=S+$\delta_i.y_i$;
end
H'(S) ← SHA256(S) ;
if H'(S)==H(S)) then
    Return 1;
end
```

The process of block creation and verification is shown in Algorithm 5. If the status of the current miner is forger, then calculate the target value as calculated from Equation (1). Hash value with taking as input Secret S, the private key of the forger, previous block hash is then calculated and initial 64 bits are then extracted and termed as “hit value”. Target value and “hit value” are compared and if target value ≥ hit value then forger can successfully produce the block [16]. The newly created block is then broadcasted to all P2P nodes that can easily verify the hash of the revealed secret value S against the value $H(S)$ supplied by the super node.

The block rewards and penalty system is explained using Algorithm 6. This block is included in the existing chain if a substantial percentage of P2P nodes approve it and reputation coins are awarded to the forger and super node in an 80 : 20 mix respectively as block rewards. If a forger creates a false block, the staked reputation coins are destroyed, and the forger must wait a certain length of time before engaging in the next round of block production [16]. The entire process of block generation is depicted in Figure 2.

Algorithm 5: Block Creation and Verification

```
Data: Set of share-holder IPFS nodes ($CID_1$,$CID_2$,...$CID_n$), H(S), threshold k,
      super-node-id,Miner-address,reputation-stake-coins
Result: Block_header (prev_block_hash, nonce, timestamp), set-of-transaction,
        target-value, hit-value, Secret S
/* Declare retrieved-shares and status */
retrieved-shares = call function share_retrieval(miner-address)
status=call function secret_recreate (retrieved-shares ($x_i$,$y_i$ ))
if status==1 & miner-address-is-valid==true & reputation-coin(miner-address) ≠
  0 & time-out(miner-address) is inactive)) then
    Grant block-generation-rights (miner-address)
    forger = miner-address
    Calculate target value according to Equation 1
    Label : Compute H = SHA256 (prev_block_hash, pri-key(forger), S,
      timestamp, nonce)
    hit-val = substring (H , 0, 64) ;  /* extract initial 64 bits as hit value
      */
    if hit_value ≤ target-value then
        generate block B
    else
        nonce ← nonce+1 ;
        goto Label
    end
    Return(new-block-created, forger, reputation-coins, super-node-id)
end
```

Algorithm 6: Block Rewards and Penalty

```
Data: new-block-created, reputation-coins, forger, super-node-id
Result: reputation-coins
if new-block-created ==valid ;                 /* if got n confirmations */
 then
    Set reputation-coins(forger) = reputation-coins + 8
    Set reputation-coins (super-node-id) = reputation-coins + 2
else
    Set reputation-coins(forger) = 0
    Set time-out (forger, x) ;      /* put forger in wait state for x time */
end
```

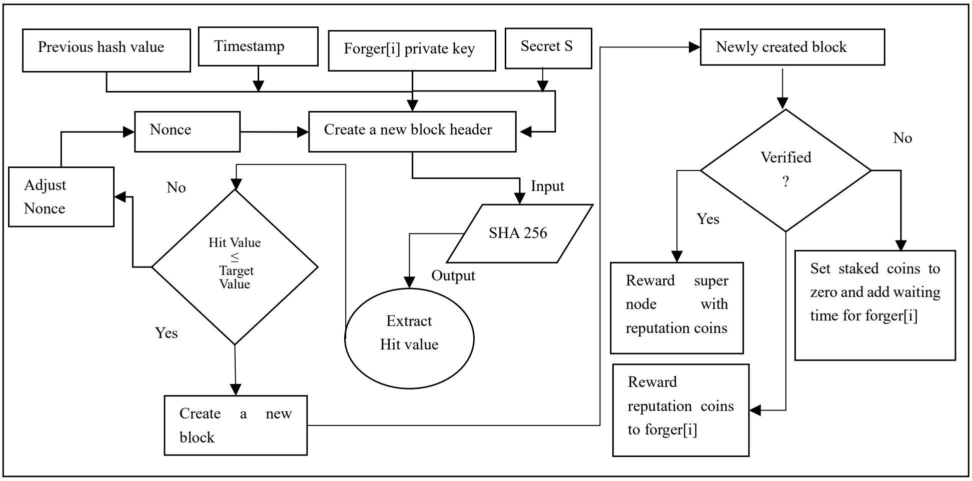

Figure 2. Block generation process

4. Analysis and performance evaluation of DPoAC

Performance evaluation framework

The performance evaluation framework is composed of various parameters that are important in designing consensus protocols. The following aspects are critical in evaluating the performance of these mainstream protocols: applicability, the basis for awarding accounting rights, degree of decentralization, accounting nodes, latency, throughput, fault tolerance rate, energy efficiency, overhead, adversary tolerance, scalability, security, and penalty mechanism. These criteria have been presented thoroughly to demonstrate their importance in consensus protocol design. We have subdivided these performance parameters into three dimensions namely efficiency, structure and security. The efficiency dimension includes all performance-related parameters such as latency, throughput, energy consumption, overhead in terms of computing, network and storage, and scalability. The structural dimension covers applicability, the basis of assigning accounting rights, degree of decentralization, accounting nodes, mining rewards, and IoT suitability [16, 18]. Finally, the security dimension contains the parameters that would ensure the resistance towards several attacks such as 51 percent attack, Sybil attack, and penalty mechanism [16].

4.1. Evaluation of DPoAC against performance parameters

In this section, DPoAC has been evaluated with a comprehensive performance evaluation matrix designed. The performance parameters are divided into three important dimensions namely, efficiency, structural, and security.

4.1.1. Efficiency

We evaluated our prototype using a variety of performance metrics from the efficiency dimension as depicted in Table 2. With a prototype created on a system with an Intel Core i3-3110M CPU, producing a Langrage polynomial with 64 degrees and 256 degrees takes about 15 ms and 46 ms, respectively, as illustrated in Figure 3a. This experiment revealed that even with low-end processors, IoT devices can efficiently produce a Langrage polynomial. The average amount of time needed to upload or download a file on IPFS nodes is shown in Figure 3b. We experimented with file sizes ranging from 1 MB to 1000 MB to see how long it took to load and receive the files.

Table 2

Efficiency comparison of DPoAC against mainstream consensus protocols

Consensus	Latency	Throughput	Energy consumption	Computing overhead	Network overhead	Storage overhead	Scalability
PoW	10 min	$\geq$ 7 TPS (bitcoin) and $\geq$ 15 TPS (Ethereum)	high	high	low	high	not scalable
PoS	1 min	$\geq$ 300 TPS	low	medium	low	high	scalable
DPoS	3 s	$\geq$ 500 TPS	low	medium	N/A	high	partially scalable
PoA	5 min	$\geq$ 14 TPS	medium	high	low	high	partially scalable
DPoAC	42 s (variable)	24–122 TPS	low	low	low	low	scalable

A minimum of 0.98 ms and 1.7 ms are needed to upload and download a file of 1 MB and seems to be quite fast as we have to store the secret shares on an IPFS node with a size much less than that of 1MB. Therefore, the resultant method would be reasonably efficient and fast for secret share distribution and retrieval on IPFS nodes that could vary up to thousands of such nodes. Figure 3c and Figure 3d both represent the computation time used to generate and distribute secret shares as well as their retrieval from IPFS nodes. The computational times grow with the number of shares created and retrieved, as seen in the graphs. In this experiment, we fixed $k = n$, with the highest computation time being recorded as 0.356 s and 0.003 s if $k = 11$ and the lowest computation time as 0.029 s and 0.001 s if $k = 1$ for secret generation and secret share retrieval phases respectively. Hence, the difficulty level of a block being generated could be directly proportional to the number of secret shares being generated. By modifying the block and transaction sizes under optimal conditions, the average latency of our system is measured at 42 s, and TPS in the range of 24 to 122 has been observed. Energy consumption of DPoAC is quite low as compared to other mainstream consensus protocols due to the lack of complex computations. Due to the usage of IPFS and the absence of complex mathematical puzzles, the overhead incurred for storage, network, and computing is greatly reduced. Additionally, the scalability of the proposed system is high as we can easily expand the system to thousands of IoT devices without significantly affecting performance. From these observations, it is implicit that our system is capable to address the limitations of existing consensus protocols in the context of IoT networks.

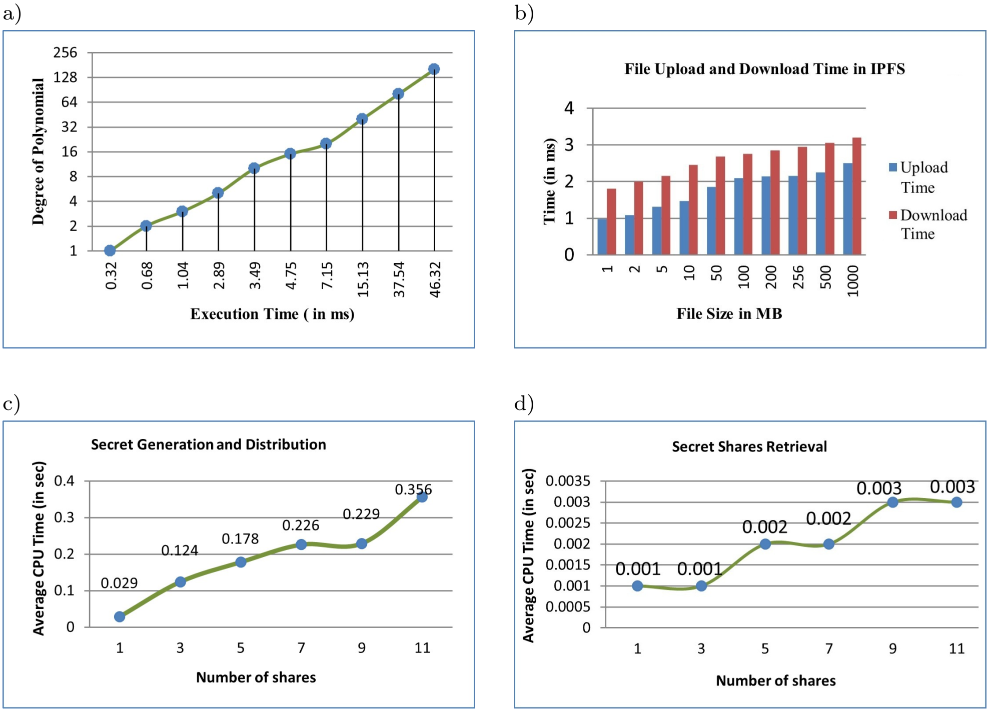

Figure 3. Performance evaluation of DPoAC: a) execution time to build Langrage polynomial; b) average time taken to upload and download a file on IPFS; c) average CPU Time taken for secret generation and distribution; d) average CPU time taken to retrieve the secret shares

4.1.2. Structural

The DPoAC structure is open to the public, and anybody with the required reputation coins can participate in the consensus process. This strategy would be extremely useful in IoT systems due to the usage of a reputation mechanism rather than a financial stake. The degree of decentralization is medium owing to the election of delegated nodes to select a super node to produce a secret and start the block generation process. Access to secret shareholders to expose the secret would provide a foundation for granting block creation rights that would extend from accounting nodes across the entire network [16]. A structural comparison of DPoAC against other consensus protocols is shown in Table 3.

Table 3
Structural comparison of DPoAC against mainstream consensus protocols

Consensus	Basis of assigning accounting rights	Degree of decentralization	Mining rewards	IoT suitability
PoW	Computing Power	high	Monetary rewards for successful miner only	Not applicable due to high resource requirement
PoS	Stake	high	Monetary rewards for successful miner only	Partially applicable due to monetized stake
DPoS	Stake Votes	medium	Monetary rewards for all the delegates distributed equally	Partially applicable due to monetized stake
PoA	Activity Based	low	Monetary rewards for all the stakeholders, and winning miner distributed equally	Not applicable due to monetized stake and high latency
DPoAC	Access to secret Shares	medium	In reputation coins to super node and forger node in 80:20 ratio	Due to the usage of reputation value as a stake and the minimized resource requirements to expose the secret, the method is fully applicable

4.1.3. Security

The proposed consensus mechanism has been demonstrated to be resistant to a number of harmful attacks. Accessibility to more than 50% of these resources, i.e. computing resources in PoW, monetary investment in PoS, and activities in PoA is required for a useful attempt in the consensus process based only on proof of effort [16]. In the intended consensus approach, however, we have merged two methods that will undoubtedly increase the price of destructive efforts while optimizing network security against a 51 percent hazard [16]. Due to the random selection and punishment mechanism for harmful conduct offered in DPoAC [16] the possibility of a DDoS attack on a P2P node or delegated super-node is extremely low and will not breach the protocol. In a Sybil attack, any fraudulent node in the blockchain network might pose as several nodes in order to gain control over the entire network and indulge in undesirable actions. This approach requires the malicious node to pay a stake that would have been forfeited as a result of such activities because it integrates secret sharing and PoS with randomized selection [16]. As a result, the implemented method is capable of preventing such attempts.

Additionally, the use of finite field arithmetic contributes to the security of the SSS scheme because the scattering of the polynomial graph that results from projecting a function on a sufficiently large finite field prohibits an attacker from discovering any aspect of this underlying function. A detailed comparison of DPoAC against security parameters is shown in Table 4.

Table 4
Security comparison of DPoAC against mainstream consensus protocols

Consensus	51 percent attack	Sybil attack	Penalty mechanism
PoW	vulnerable	vulnerable	Does not exist
PoS	less vulnerable	vulnerable	Penalty mechanism would seize the staked coins for the malicious block
DPoS	less vulnerable	vulnerable	The faulty node would be eliminated immediately
PoA	eliminates 51 percent attack	less vulnerable	The faulty node would be eliminated immediately
DPoAC	less vulnerable	less vulnerable	Reputation coins would be lost due to faulty behavior of the forger

5. Conclusions

Through this work, we have implemented and evaluated a lightweight consensus method, DPoAC, that has been derived from classic Proof of Accuracy and PoS consensus protocols. This approach included the SSS scheme to generate and distribute secret values, and IPFS was used to store the individual secret partitions. Although there have been a few papers based on proof-of-accuracy techniques in the past, they were all without concrete implementations. Through this work, we have not only provided detailed algorithms to justify the consensus process but also implemented and evaluated the results. A comprehensive performance matrix has been designed to evaluate and compare the performance of DPoAC. We implemented a prototype, and experimental results show that our protocol works reasonably well in comparison to other mainstream consensus protocols, resolving the issues of energy consumption, heavy resource requirements, and reliance on monetary stakes to become an optimal choice in the IoT context. In future, we would like to test the validity of our protocol at a large scale with real-time IoT data. In addition, we will test the performance of our protocol by including different secret-sharing schemes.

References

[1] Andrian Y., Kim H., Ju H.: A Distributed File-Based Storage System for Improving High Availability of Space Weather Data, *Applied Sciences*, vol. 9(23), 5024, 2019. doi: 10.3390/app9235024.

[2] Aponte-Novoa F.A., Villanueva-Polanco R.: On Proof-of-Accuracy Consensus Protocols, *Mathematics*, vol. 10(14), 2504, 2022. doi: 10.3390/math10142504.

[3] Athanere S., Thakur R.: Blockchain based hierarchical semi-decentralized approach using IPFS for secure and efficient data sharing, *Journal of King Saud University – Computer and Information Sciences*, vol. 34(4), pp. 1523–1534, 2022. doi: 10.1016/j.jksuci.2022.01.019.

[4] Banafa A.: IoT and blockchain convergence: benefits and challenges, IEEE Internet of Things, 2017. https://iot.ieee.org/standards/37-newsletter/january-2017/208-iot-and-blockchain-convergence-benefits-and-challenges.html.

[5] Chen Y., Li H., Li K., Zhang J.: An improved P2P file system scheme based on IPFS and Blockchain. In: *2017 IEEE International Conference on Big Data (Big Data)*, pp. 2652–2657, IEEE, 2017. doi: 10.1109/bigdata.2017.8258226.

[6] Debus J.: *Consensus methods in blockchain systems*, Frankfurt School of Finance & Management, Blockchain Center, Technical Report, 2017.

[7] Farahani B., Firouzi F., Luecking M.: The convergence of IoT and distributed ledger technologies (DLT): Opportunities, challenges, and solutions, *Journal of Network and Computer Applications*, vol. 177, 102936, 2021. doi: 10.1016/j.jnca.2020.102936.

[8] Fukumitsu M., Hasegawa S., Iwazaki J.y., Sakai M., Takahashi D.: A proposal of a secure P2P-type storage scheme by using the secret sharing and the blockchain. In: *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, pp. 803–810, IEEE, 2017. doi: 10.1109/aina.2017.11.

[9] Geng T., Njilla L., Huang C.T.: Delegated Proof of Secret Sharing: A Privacy-Preserving Consensus Protocol Based on Secure Multiparty Computation for IoT Environment, *Network*, vol. 2(1), pp. 66–80, 2022. doi: 10.3390/network2010005.

[10] Harris C.G.: Consensus-based secret sharing in blockchain smart contracts. In: *2019 International Workshop on Big Data and Information Security (IWBIS)*, pp. 79–84, IEEE, 2019. doi: 10.1109/iwbis.2019.8935853.

[11] Jennath H., Asharaf S.: Survey on blockchain consensus strategies. In: *ICDSMLA 2019*, pp. 637–654, Springer, 2020. doi: 10.1007/978-981-15-1420-3_68.

[12] Kara M., Laouid A., Hammoudeh M., AlShaikh M., Bounceur A.: Proof of Chance: A Lightweight Consensus Algorithm for the Internet of Things, *IEEE Transactions on Industrial Informatics*, vol. 18(11), pp. 8336–8345, 2022. doi: 10.1109/tii.2022.3168747.

[13] Kaur M., Gupta S.: Blockchain Consensus Protocols: State-of-the-art and Future Directions. In: *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, pp. 446–453, IEEE, 2021. doi: 10.1109/ictai53825.2021.9673260.

[14] Kaur M., Gupta S.: Blockchain technology for convergence: An overview, applications, and challenges, *Blockchain and AI Technology in the Industrial Internet of Things*, pp. 1–17, 2021. doi: 10.4018/978-1-7998-6694-7.ch001.

[15] Kaur M., Gupta S., Kumar D., Raboaca M.S., Goyal S.B., Verma C.: IPFS: An Off-Chain Storage Solution for Blockchain. In: *Proceedings of International Conference on Recent Innovations in Computing: ICRIC 2022, Volume 1*, pp. 513–525, Springer, 2023. doi: 10.1007/978-981-19-9876-8_39.

[16] Kaur M., Gupta S., Kumar D., Verma C., Neagu B.C., Raboaca M.S.: Delegated Proof of Accessibility (DPoAC): A Novel Consensus Protocol for Blockchain Systems, *Mathematics*, vol. 10(13), 2336, 2022. doi: 10.3390/math10132336.

[17] Kaur M., Khan M.Z., Gupta S., Alsaeedi A.: Adoption of Blockchain With 5G Networks for Industrial IoT: Recent Advances, Challenges, and Potential Solutions, *IEEE Access*, pp. 981–997, 2021. doi: 10.1109/access.2021.3138754.

[18] Kaur M., Khan M.Z., Gupta S., Noorwali A., Chakraborty C., Pani S.K.: MBCP: Performance analysis of large scale mainstream blockchain consensus protocols, *IEEE Access*, vol. 9, pp. 80931–80944, 2021. doi: 10.1109/access.2021.3085187.

[19] Kudin A.M., Kovalenko B.A., Shvidchenko I.V.: Blockchain technology: Issues of analysis and synthesis, *Cybernetics and Systems Analysis*, vol. 55(3), pp. 488–495, 2019. doi: 10.1007/s10559-019-00156-1.

[20] Li X., Jiang P., Chen T., Luo X., Wen Q.: A survey on the security of blockchain systems, *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020. doi: 10.1016/j.future.2017.08.020.

[21] Liang W., Tang M., Long J., Peng X., Xu J., Li K.C.: A secure fabric blockchain-based data transmission technique for industrial Internet-of-Things, *IEEE Transactions on Industrial Informatics*, vol. 15(6), pp. 3582–3592, 2019. doi: 10.1109/tii.2019.2907092.

[22] Miglani A., Kumar N., Chamola V., Zeadally S.: Blockchain for Internet of Energy management: Review, solutions, and challenges, *Computer Communications*, vol. 151, pp. 395–418, 2020. doi: 10.1016/j.comcom.2020.01.014.

[23] Naz M., Al-zahrani F.A., Khalid R., Javaid N., Qamar A.M., Afzal M.K., Shafiq M.: A secure data sharing platform using blockchain and interplanetary file system, *Sustainability*, vol. 11(24), 7054, 2019. doi: 10.3390/su11247054.

[24] Salimitari M., Chatterjee M., Yuksel M., Pasiliao E.: Profit maximization for bitcoin pool mining: A prospect theoretic approach. In: *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*, pp. 267–274, IEEE, 2017. doi: 10.1109/cic.2017.00043.

[25] Shamir A.: How to Share a Secret (1979). In: H.R. Lewis (ed.), *Ideas That Created the Future*, pp. 475–478, MIT Press, 2021. doi: 10.7551/mitpress/12274.003.0048.

[26] Ullah Z., Raza B., Shah H., Khan S., Waheed A.: Towards Blockchain-Based Secure Storage and Trusted Data Sharing Scheme for IoT Environment, *IEEE Access*, vol. 10, pp. 36978–36994, 2022. doi: 10.1109/access.2022.3164081.

[27] Yaga D., Mell P., Roby N., Scarfone K.: Blockchain technology overview, *arXiv preprint arXiv:190611078*, 2019. doi: 10.6028/NIST.IR.8202.

[28] Zhang E., Li M., Yiu S.M., Du J., Zhu J.Z., Jin G.G.: Fair hierarchical secret sharing scheme based on smart contract, *Information Sciences*, vol. 546, pp. 166–176, 2021. doi: 10.1016/j.ins.2020.07.032.

[29] Zhou J., Feng Y., Wang Z., Guo D.: Using secure multi-party computation to protect privacy on a permissioned blockchain, *Sensors*, vol. 21(4), 1540, 2021. doi: 10.3390/s21041540.

Affiliations

Manpreet Kaur
Guru Nanak Dev Engineering College, Department of Computer Science and Engineering, Ludhiana, India
Chandigarh University, Department of Computer Science and Engineering, Gharaun, India, preetmand@gmail.com

Shikha Gupta
Chandigarh University, Department of Computer Science and Engineering, Gharaun, India, shikha.g.206@gmail.com

Received: 15.05.2023
Revised: 29.02.2024
Accepted: 4.04.2024