mathematics

MDPI

Article

Delegated Proof of Accessibility (DPoAC): A Novel Consensus Protocol for Blockchain Systems

Manpreet Kaur [1,2,*], Shikha Gupta [1], Deepak Kumar [3], Chaman Verma [4], Bogdan-Constantin Neagu [5,*] and Maria Simona Raboaca [6,*]

[1] Department of Computer Science and Engineering, University Institute of Engineering, Chandigarh University, Gharuan, Mohali 140413, Punjab, India; shikha.g.206@gmail.com or shikha.e8741@cumail.in
[2] Department of Computer Science and Engineering, Guru Nanak Dev Engineering College, Ludhiana 141006, Punjab, India
[3] Apex Institute of Technology, Chandigarh University, Mohali 140413, Punjab, India; deepak.e11296@cumail.in
[4] Department of Media and Educational Informatics, Faculty of Informatics, Eötvös Loránd University, 1053 Budapest, Hungary; chaman@inf.elte.hu
[5] Department of Power Engineering, "Gheorghe Asachi" Technical University of Iasi, 700050 Iasi, Romania
[6] National Research and Development Institute for Cryogenic and Isotopic Technologies—ICSI Rm. Vâlcea, Uzinei Street, No. 4, P.O. Box 7 Râureni, 240050 Râmnicu Vâlcea, Romania
* Correspondence: manpreet_mand@gndec.ac.in or preetmand@gmail.com (M.K.); bogdan.neagu@tuiasi.ro (B.-C.N.); simona.raboaca@icsi.ro (M.S.R.)

Abstract: As the backbone of every blockchain application, the consensus protocol is impacted by numerous risks, namely resource requirements and energy consumption, which limit the usage of blockchain. Applications such as IoT/IIoT cannot use these high-cost consensus methods due to limited resources. Therefore, we introduce Delegated Proof of Accessibility (DPoAC), a new consensus technique that employs secret sharing, PoS with random selection, and an interplanetary file system (IPFS).DPoAC is decomposed into two stages. During the initial stage, a secret is generated by a randomly chosen super node and divided into n shares. These shares are encrypted and stored in different n nodes on the IPFS network. The nodes will compete to access these shareholders to reconstruct the secret. The winning node will be awarded block generation rights. PoS with random selection is used in the second stage to compute the appropriate hash value and construct a block with valid transactions. In this novel approach, a node with few computational resources and small stakes can still obtain block generation rights by providing access to secret shares and reconstructing the secret, making the system reasonably fair. We qualitatively analyze and compare our scheme based on performance parameters against existing mainstream consensus protocols in the context of IoT/IIoT networks.

Keywords: blockchain; consensus; DPoAC; secret sharing; IPFS; IoT; IIoT

MSC: 68w10

Citation: 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* **2022**, *10*, 2336. https://doi.org/10.3390/math10132336

Academic Editor: Jan Lansky

Received: 22 May 2022
Accepted: 27 June 2022
Published: 3 July 2022

1. Introduction

Following the success of Bitcoin, blockchain technologies have been gaining popularity. Blockchain has become an interesting option for academics and researchers due to its intrinsic characteristics, such as the absence of central authority, transparency, and security. Blockchain is among the most revolutionary innovations, with the ability to transform the behavior of many operations and industries [1]. Blockchain is essentially a peer-to-peer network of nodes that facilitates communication across several non trusting nodes in order to add new blocks to the end of an existing blockchain while keeping the previous blocks intact [2]. The blockchain framework comprises various layers. The

network layer, consensus layer, data model layer, execution layer, and application layer are the five essential layers that are relevant to practically any blockchain application [2]. The consensus protocol, which resides at the consensus layer, is critical in determining network security and performance measures. Consensus mechanisms are a core component of the long-term stability of a blockchain system. The potential of a system to verify the accuracy and authenticity of a block without the involvement of a trusted intermediary is a significant benefit of using a blockchain application [3]. Despite the lack of centralized control over transaction validation and confirmation, blockchain claims that every transaction is entirely secure and verified [3]. The existence of a consensus algorithm, as a key component of any blockchain network, is the driving force behind it.

Most modern blockchain solutions are incapable of satisfying the demands of any large-scale real-world application due to significant restrictions imposed by scalability, security, and performance. Many of these constraints are the result of issues generated by the underpinning consensus. The development of more realistic blockchain networks is centered on consensus mechanisms [3]. Consequently, a variety of consensus protocols have been devised to optimize the efficiency of blockchain systems while addressing the unique needs of application domains [2].

Proof of Work (PoW), Proof of Stake (PoS), Proof of Activity (PoA), and Delegated Proof of Stake (DPoS) are some mainstream consensus methods that already exist and have been evaluated in the blockchain context [2]. In PoW, participants are required to solve a computationally intensive puzzle, making the process very hard, expensive, and time-consuming. Because of the high resource requirements of PoW, it is unsuited for low-resource applications such as IoT/IIoT. Thus, many variants of PoW have been implemented; one such alternative is PoS, in which a node has to deposit some amount of cryptocurrency to receive the block generation rights. The core principle is that nodes with higher stakes are more valuable than other nodes, hence the potential of a node being honest is quite high; otherwise, they would lose their stake invested in the network. However, in this approach, nodes with greater stakes would become richer, and they would have the incentive to create more and more blocks than nodes with lower stakes. The stake deposited to participate in the block generation process becomes too high for smaller stake nodes beyond a certain number of blocks being created, prompting small stake-holding nodes to quit the network, limiting scalability and increasing network monopoly.

In addition to these two most common consensus algorithms, there are some other variants also such as DPoS and PoA. DPoS is more effective, faster, and more decentralized as compared to PoS. A set of nodes is selected in the network as delegates by an election mechanism through stakeholders. These delegators will generate a new block according to a predetermined pattern, and if a delegator fails to create a block in its turn, then it will be removed. In comparison to PoW and PoS, DPoS is a more cost-effective and high-performance consensus protocol [2]. However, voting cannot prevent the selection of fraudulent nodes, particularly when the network is small, and may pose certain security threats.

As the Bitcoin reward becomes halved for every 210,000 blocks mined and the time when it becomes obsolete, miners tend to lose their motivation to participate in the mining process. Hence, miners would demand high transaction fees with respect to the computing resources involved in the mining process. Therefore, these conflicts have been addressed in a hybrid approach called Proof of Activity (PoA), which is a combination of PoW and PoS. This approach is decomposed into two stages. During the first stage, it works as a pure PoW to create a block header by solving a computational puzzle. In the next stage, a set of N validators is chosen based on their stake, to validate the transactions present in a block one by one, and after N confirmation, that block is appended to the existing chain. The rewards to create the block are shared among the miner who is the winner of the first stage and N validators on an equal basis. However, this approach also suffers from huge resource requirements and is relatively complex to implement.

Existing IoT devices have limited computational capabilities in terms of processing and capabilities. In addition, the security and scalability concerns of IoT devices could be addressed by integrating blockchain into IoT/IIoT applications. As a decentralized framework, blockchain minimizes the possibility of a single point of failure and enhances security. Data are distributed throughout the network rather than being stored on a single server, which enhances scalability. Therefore, blockchain could be the perfect choice to complement the limitations of IoT devices. However, the existing mainstream consensus mechanisms share several limitations, including limited efficiency, excessive power usage, and high resource requirements. For these constraints, blockchain applications are impracticable, particularly in the IoT/IIoT environment [4].

Such concerns prompt us to introduce a novel consensus protocol that inherits the benefits of existing consensus protocols. Therefore, in this paper, we proposed Delegated Proof of Accessibility (DPoAC), a novel consensus protocol derived from the novel techniques Proof of Accessibility and Proof of Stake with randomized selection.

Our contributions could be summarized as follows:

(a) We propose a novel consensus protocol, DPoAC, based on secret sharing and capable of solving the limitations of high resource requirements and reduced efficiency faced by current consensus methods.
(b) We performed qualitative analysis of this proposed consensus algorithm in terms of important parameters such as fault tolerance, resource saving, computational complexity, and security.

The remaining paper is structured into the following sections. We present some preliminary concepts in Section 2 to enable readers to comprehend the information employed in the proposed approach. Section 3 focuses on some related work in the existing literature. The problem statement is defined in the following section. Section 5 defines a model for the proposed system. Section 6 contains a description and flow diagrams of our proposed consensus algorithm. Section 7 evaluates our novel protocol against various parameters such as security, fault tolerance, and so on. Section 8 provides a discussion on existing consensus algorithms and their comparison with DPoAC, particularly in IoT systems, before concluding our work.

2. Preliminary Concepts

An outline of the basic concepts used in the design of our consensus protocol is included in this part. To begin with, we introduce the consensus formally and its key properties. Then, we discuss threshold cryptography, secret sharing, interplanetary file systems, and distributed hash tables.

2.1. Consensus

Definition 1.(Consensus)**.** *Assume t nodes in an n-node network are dishonest. As illustrated in Figure 1, the consensus is said to be achieved if the under-mentioned characteristics are satisfied.*

(a) Agreement: All honest nodes must reach the same agreement.
(b) Validity: The agreed-upon value is one among many input values held by nodes.
(c) Termination: All honest nodes must terminate within a predetermined time.

The security evaluation of any consensus protocol is often assessed with this characteristic-based approach, which involves the demonstration of the process to satisfy certain criteria that include agreement, validity, and termination. The largest number of dishonest nodes acceptable by a protocol along with the termination time of honest nodes and complexity of communication are some prominent factors to describe the quality of a consensus protocol.

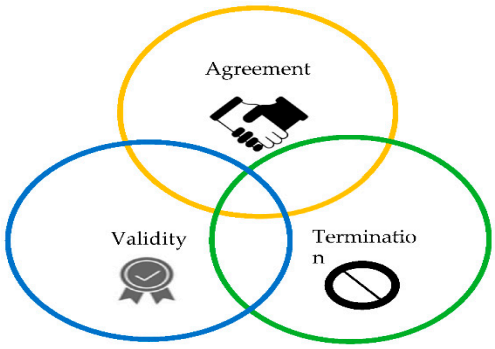

Figure 1. Consensus properties.

2.2. Threshold Cryptography

Threshold cryptography [5,6] allows a secret-owner to share a secret with a set of users. A (k, n)-threshold approach creates and distributes n shares of the secret among the parties [7]. A minimum of k distinct shares must be integrated to reassemble the secret. Therefore, the strengths of threshold cryptography are not only leveraged by distribution of trust but also through fault tolerance [8].

2.3. Secret Sharing

2.3.1. Additive Secret Sharing (ASS)

In this scheme [9], a secret S is segmented into n unique additive shares $[S]_k, k \in [1, n]$such that $\sum_{k=0}^{n}[S]_k = S$ where S∈ℝ (a field). ASS in particular is an (n, n)-threshold scheme that requires knowledge of all the n shares in order to reconstruct the secret S. ASS is a linear secret sharing scheme that performs a linear operation $\sum_{k=0}^{n}(S_k \pm r_k) = S \pm b$, where r_k is a random number chosen locally. Every party must perform $S_k \pm r_k$ locally, and the result in additive form is shared across others [10].

2.3.2. Shamir's Secret Sharing (SSS)

SSS is a distributed mechanism to protect a secret, most commonly to secure other encryption keys. The secret is divided into several parts known as shares. The initial secret is reassembled with the help of these individual shares. A certain amount of shares is required to uncover the secret through SSS. This is referred to as the threshold, and it denotes the minimum necessary of shares required to reveal the secret [5].To better understand the SSS scheme, the secret is some data S that are decomposed into n distinct parts, $S_1, S_2, \ldots S_n$,in such a way that the following conditions satisfy [6]:

1. Knowledge of atleast k number of distinct $S_i$parts is required to determineoriginal data S.
2. If there is knowledge of only ≤ k−1 parts of S_i, then it becomes difficult to reconstruct original data S, leading to S being undetermined.

Here, only k data parts out of n are required to regenerate the original secret; therefore, it is known as the (k, n)—threshold scheme. Two points are required to plot a line, while a parabola requires three points to uniquely identify [11]. Similarly, the basic principle of SSS is built on the Lagrange interpolation theorem, which states that k points are sufficient to uniquely identify a polynomial with a degree less than or equal to k−1.

The SSS scheme must fulfill some useful properties also listed as follows [11]:

- **Secure**: The secret data are generated with information-theoretic security if the adversary does not acquire any extra information while executing the real-world protocol than the information it obtains under ideal settings with central control.
- **Minimal**: The size of the original data must be greater than that of the size of the individual part.

- **Dynamic**: Without altering the secret, security may be readily improved by modifying the polynomial $f(x)$on a regular basis and allocating fresh shares to the parties.
- **Flexible**: Each member could be allocated a variable amount of shares based on their role inside the organization. Because of this weighting technique, higher-ranking participants receive a large amount of shares from the total number of shares available.

Theorem 1.*(Lagrange interpolation). Assume a finite field* $\mathfrak{F}$. *Then, k distinct pairs* (x_i, y_i)*produce a polynomial* $f(x)$ *of degree* ≤ *k–1 such that* $f(x_i) = y_i$*Let k–1* < $|\mathfrak{F}|$ *so that all* x_i*'s are unique. Then,*$f(x)$*is calculated as:*

$$f(x) = \sum_{i=0}^{k} y_i \prod_{\substack{1 \le j \le k \\ i \ne j}} \frac{x - x_j}{x_i - x_j} \quad (1)$$

To share the secret S as S⟶ S_1 , S_2 , ... S_n, the following steps are employed:

- Select a sufficiently big prime number as P and assume $\mathfrak{F} = \mathbb{Z}/p\mathbb{Z}$.
- Select coefficients f_1 , f_2 , ... $f_{k-1} \in \mathfrak{F}$, which are to be the coefficients of degree k–1 polynomial f.
- $f(z) = f_0 + f_1 z + \cdots + f_{k-1} z^{k-1}$, where $f_0 = S$.
- Determine the value of each f_i and assign it to user i.

SSS reconstruction is built on the Lagrange interpolation theorem presented in (1), which can be seen in Figure 2. If k parties are involved and $f_i$represents the ith party, then using k points on the polynomial curve with degree ≤ (k–1) enables calculation of the specific coefficients to a polynomial with (k–1) degree. The coefficient $f_0$in the polynomial defines the secret S. From any collection of k shares, $f_0$could be revealed through an interpolation given below [4]:

$$f(x) = \sum_{i=0}^{k} y_i \prod_{\substack{1 \le j \le k \\ i \ne j}} \frac{x - x_j}{x_i - x_j} \quad \twoheadrightarrow \quad f(0) = \sum_{i=0}^{k} y_i \prod_{\substack{1 \le j \le k \\ i \ne j}} \frac{x_j}{x_j - x_i} \quad (2)$$

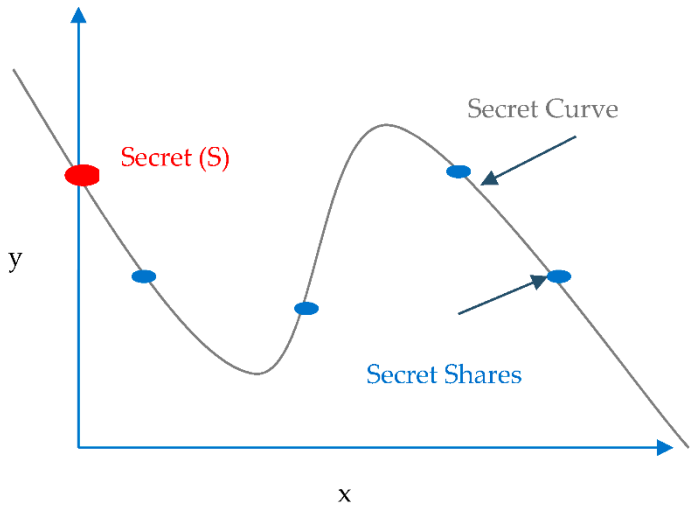

Figure 2. Lagrange interpolation.

2.4. Interplanetary File System (IPFS)

The interplanetary file system (IPFS) is a peer-to-peer distributed file system that is viewed as an alternative to HTTP [12,13]. In contrast to HTTP, content-based indexing is employed in IPFS; whenever a file is added to the system, it is separated into pieces of 256kB. Each of these pieces contains object data and references to be stored in a Merkle DAG. The system provides a single hash known as the base content identifier (CID), which is then used to retrieve the file from IPFS. Such hash creation also ensures network de duplication since the same hash is generated from a file when it is uploaded to IPFS repeatedly, and even a slight variation in the file will result in a totally different base CID hash [14]. Distributed hash tables (DHTs) are used to store data on IPFS; the distributed part allows recent hash tables to be made available in multiple locations [15].

2.5. Distributed Hash Table (DHT)

DHTs are primarily used in P2P networks to record and maintain information. A distributed hash table (DHT) is a decentralized solution to provide a lookup structure similar to a hash table to keep index-value pairs, and individual participants could effectively obtain the value linked to a specific index. The DHT is used by IPFS and other decentralized content systems to allow routing, discovery of content, and peers on the network.

3. Related Work

Naz et al. [16] presented a blockchain-enabled data sharing and digital assets delivery system by making use of IPFS. By executing the authorization functions in an owner-written smart contract, this approach improves security and access control. The proposed system was implemented on a private Ethereum blockchain. Due to the encryption provided by the Shamir secret sharing scheme to IPFS data hashes, access to data is restricted to those clients who have pending digital content payments. As a result, the owner is protected from any kind of hash leaking to an illegitimate user. Moreover, the smart contract for the review system may assist users in looking for and posting reviews. The simulation results for energy consumption and economic evaluation of the proposed system have been performed.

Zhang et al. [17] offered a novel blockchain-enabled hierarchical threshold secret sharing approach. Private shares are allocated among multiple tiers of users in the system, and the secret can be obtained by any permitted subset of those users. Smart contracts were created to detect malicious activities and to maintain the integrity of the secret sharing procedure. If users do not abide by the rules honestly, fraudulent activity may be identified, leading to financial penalties. Finally, without a central authority, participants can recreate the secret in a fair manner.

Kudin et al. [18] proposed a theoretical consensus method named Proof of Accuracy (PoA). In PoA, the work of identifying the block leader is based on solving a problem with a specific computational complexity threshold, similar to Proof of Work, as well as providing proof to access input data necessary to solve the problem. A theoretical concept was provided without implementation details.

Liang et al. [19] explored a secure fabric-based data transport system using the SSS technique to move data across trading hubs using a secret-based system. A data consensus algorithm is used to store data in a flexible linked-based storage solution. However, this technique is intended for modest applications, and the importance of power data security is ignored.

Online data stored on a central server managed by a single organization are vulnerable to a variety of attacks. Masayuki et al. [4] suggested secure storage without a central server. Personal data are protected from malicious parties. Using a secret sharing mechanism, each user's information is divided into segments. These segments are saved on distinct network nodes. By masking most essential traits of data, they are turned into

metadata. For reconstructing the original data, a user may search for network nodes containing the data segments. This suggested system is trustworthy because it allows a user to recognize target data even when peer nodes change. Furthermore, utilizing the majority rule consensus, a fraudulent node may be discovered by other network nodes. However, system security is not assessed in a quantitative manner.

Geng et al. [20] proposed an enhanced consensus system that incorporates a verifiable secret sharing scheme in the context of a large blockchain network. Privacy preservation is ensured by verifiable secret sharing, and secure multiparty computation is used to enhance security, efficiency, and fairness.

Zhou et al. [21] explored the privacy protection features of secure multiparty computation in the permissioned blockchain. This work integrated a secure MPC protocol in Hyperledger Fabric. The proposed protocol used secret sharing Homomorphic encryption and zero-knowledge proof.

Andrian et al. [22] highlighted the use of IPFS to increase data availability and throughput by distributing data onto distinct IPFS nodes. A real-time monitoring system was added to provide data flow and node status. The performance of the proposed system was compared with existing systems, and the experimental results proved that the proposed IPFS-based system can minimize the file replication time and improve throughput.

Hoogerwerf et al. [23] explored an efficient method for generating joint random numbers in a multi-party context. The involved parties independently produced random number via bit-wise sharing and merged them to generate a final secret value.

4. Problem Statement

A consensus mechanism, which serves as the backbone of every BC application, has a significant impact on the functionality and efficiency of the underlying BC. The consensus mechanism is the process of obtaining an agreement on a piece of information from the majority of nodes in a distributed environment. Consensus methods are designed to preserve consistency in a network with several faulty nodes; as a result, certain communications must be assumed unavailable, and the consensus process must be fault-tolerant. These protocols determine which nodes can contribute new blocks to the chain. Blockchain consensus algorithms are divided into two types: proof-based protocols and voting-based protocols. The former needs participants to submit evidence of resources or efforts consumed to execute a computationally intensive activity, while the latter requires a majority of votes from participants to decide whether to add a new block or which node is permitted to produce a new block. Hence, the potential for boosting the effectiveness of the blockchain network is totally dependent on the underpinning consensus process. Therefore, our study addresses the following research question:

RQ: How to improve the performance of blockchain in IoT/IIoT applications?

To answer this question, we proposed a novel consensus protocol (DPoAC), which would be capable of improving the overall performance of the BC network and more suitable to integrate the blockchain into the IoT/IIoT ecosystem.

As the client–server paradigm is used in IoT systems to store and process data across numerous IoT devices, there are always reliability and privacy vulnerabilities in the event of a single point of failure or server collusion. To tackle this, we will employ IPFS, a P2P decentralized version-controlled file system, where each file is content-addressable and indexed with its own hash value. IPFS file hashes may be easily saved on the blockchain as well. Therefore, IPFS could act as an excellent foundation for designing an IoT-based decentralized access control system [24].

5. System Design

We present a design of our planned system model in this part as depicted in Figure 3. The below-listed entities and modules compose the proposed model.

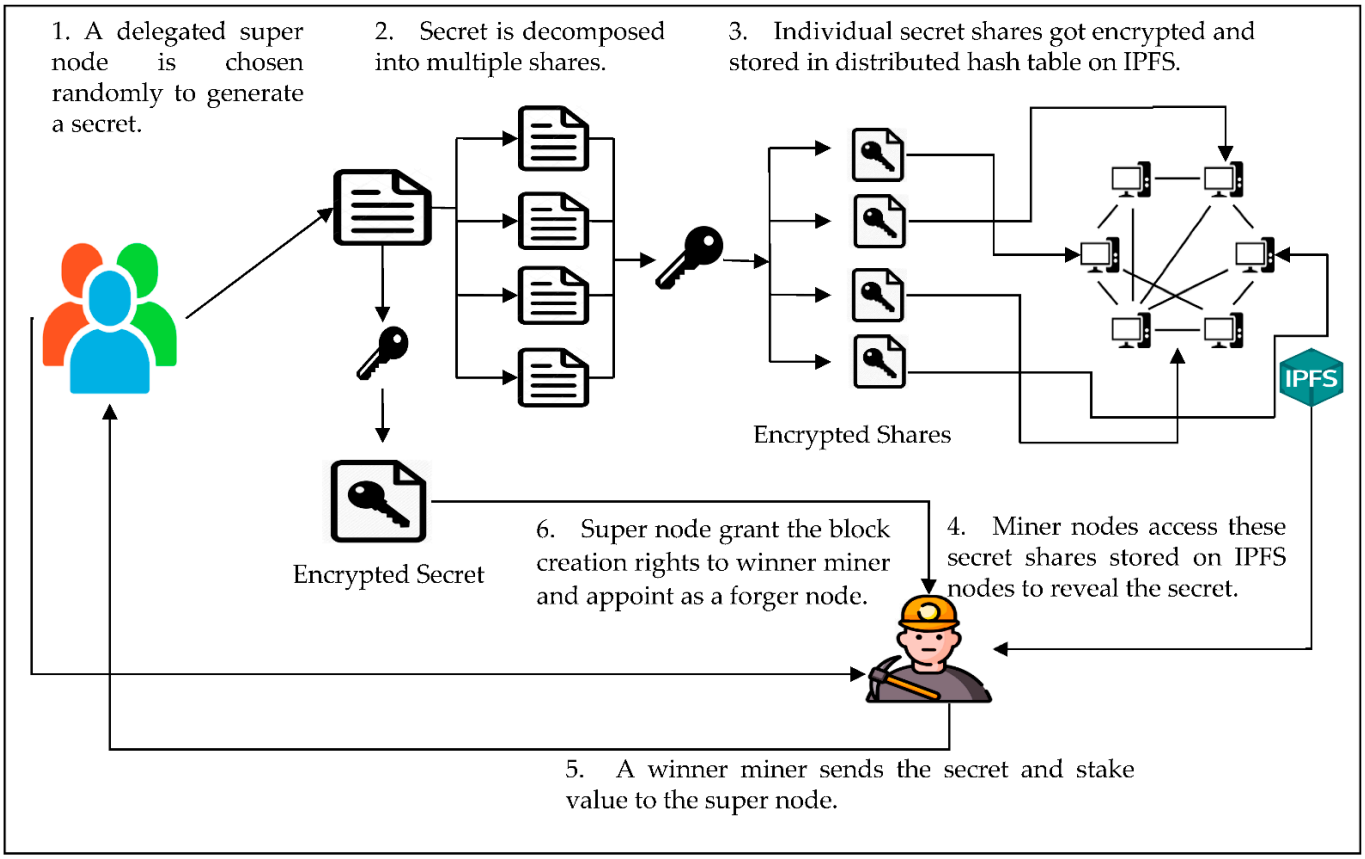

Figure 3. Proposed system design.

5.1. Entities

- Delegated Super Node: A node is chosen randomly from a set of delegated nodes to initiate the process of allocating block creation rights among other nodes.
- P2P Nodes: Each node in a P2P network is a P2P node that has a pair of public and secret keys along with other metadata.
- Miner Nodes: Nodes with specialized capabilities that are capable of revealing the secret number generated by the delegated super node within time constraints.
- Secret Shareholders: P2P nodes that would be used to hold the secret shares generated by the system and are obliged to share among their peers who have been identified as authentic.
- Forger Node: Anode that has successfully revealed the secret number and is assigned as a block creator.

5.2. Modules

1. *Secret Generation and Distribution*: A delegated super node is chosen at random to generate a random secret number, say S, and the hash of this number S, which is unknown to P2P participants, is computed as H(S) and preserved via a (k, n)-threshold cryptographic protocol like the Shamir secret sharing protocol. Following that, the secret number S is divided into n multiple shares or parts. These shares are then encrypted, and utilizing the IPFS protocol, these encrypted shares, together with metadata, are recorded on different shareholders. The value H(S) will be broadcasted to all other nodes.
2. *Secret Shares Retrieval*: A collection of a minimum of k shares from a total of n shares is necessary to reveal a secret. Therefore, IPFS nodes holding the secret shares are accessed by miner nodes or the nodes that need to become the block creator. The first node that has accessed and collected k shares will be able to reveal the secret and

proves this to the delegated super node. All other nodes will verify that the node has accessed the shares and revealed the correct secret by verifying it against the value H(S).

3. *Block Creation and Verification*: Once a node has proved that it has accessed all the shareholders and revealed the correct secret generated by the super node, it will be assigned to create the block creation rights; we named this node forger node. A forger uses its secret key to calculate an encrypted value from the hash of its predecessor block. The hash of this encrypted value is then calculated, and the first 64 bits of the resultant hash value are termed as "hit value". The inclusion of the secret key in the above computation ensures that a unique hit value is derived exclusively from a forger. For the forging, a node producing a hit value less than that of the "target value" is selected [2]. The target value is obtained by using (3).

$$Target\ value = T_b \times L \times S_e \quad (3)$$

where T_b = "base target value" = previous block target value × time consumed to forging that block, L = time elapsed since the last block forged, and S_e = amount of reputation coins accumulated or staked [2].

Once a block has been created, it is disseminated to other P2P nodes for verification. The block will be stored on the blockchain if more than half of the network nodes have verified it.

4. Block Rewards and Penalty: If a forger successfully creates a block and that block is verified by the majority of P2P nodes, then the forger and super node will be entitled to receive rewards in terms of reputation coins with an 80-20 ratio, respectively. On the other hand, if a forger attempts to construct a fraudulent block, then staked reputation coins will be lost, and that forger will have to wait for a specified amount of time to take part in the next block creation round. This type of mechanism will protect the system against malicious attacks.

6. Proposed Consensus Algorithm (Delegated Proof of Accessibility)

Though many consensus algorithms exist in the literature, they all come with various shortcomings such as efficiency, energy consumption, and privacy concerns. We propose a new consensus algorithm, Delegated Proof of Accessibility, based on secret sharing and proof of stake with randomized selection. The algorithm is divided into two phases: (a) Proof of Accessibility and (b) PoS using randomized selection, as shown in Figures 4 and 5, respectively.

6.1. Phase 1 (Proof of Accessibility)

1. If the transaction pool is not empty, then a node is chosen randomly from a set of delegators as a super node.
2. The super node will generate a secret number S using Shamir's secret sharing algorithm.
3. Compute the hash value of this secret number and store it as H(S).
4. Decompose S into N different shares/parts as S→ $\boldsymbol{S_1}$, $\boldsymbol{S_2}$, ... $\boldsymbol{S_n}$.
5. Encrypt each Si share as S_{ie} and compute the hash value of each encrypted share as $\mathbf{CID_i}$ = H($\mathbf{S_{ie}}$), and these shares must be stored on atleast n distinct nodes using IPFS.
6. Broadcast the CID of secret shareholders on IPFS and H(S) to the P2P network.
7. The miner nodes will have to access ≥ k secret shareholders and retrieve the data of $\mathbf{S_{ke}}$shares.
8. Decrypt $\mathbf{S_{ke}}$ encrypted shares to retrieve the secret shares $\mathbf{S_1}$ to $\mathbf{S_k}$ as a (k, n)-threshold is required to reconstruct the secret S'.
9. Compute the hash S' as H(S') and match it with the broadcasted value H(S); if matched, then the miner will forward the secret number S' and staked coins to the super node.

10. The super node verifies the sent secret and, if verified, grants block generation rights to that miner.

6.2. Phase 2 (PoS with Randomized Selection)

1. Calculate the target value for the current forger $\mathbf{F_i}$ by the formula given in (3).
2. Compute a hash value for a new block using the private key of F_i, and the secret number S revealed.
3. Extract the first 64 bits of this hash value as "hit value".
4. Match this hit value to the specified target value.
5. If target value ≥ hit value, then $\mathbf{F_i}$ can successfully create the new block and broadcast this block with all the P2P nodes.
6. If the majority of P2P nodes verify this block, it is added to the current chain, and the transaction fee as block rewards will be released to forger $\mathbf{F_i}$and super node in an80–20 ratio, respectively.
7. If forger $\mathbf{F_i}$ creates a malicious block, then the staked reputation coins will be lost, and that forger $\mathbf{F_i}$will have to wait for a specified amount of time to take part in the next block creation round in the future.

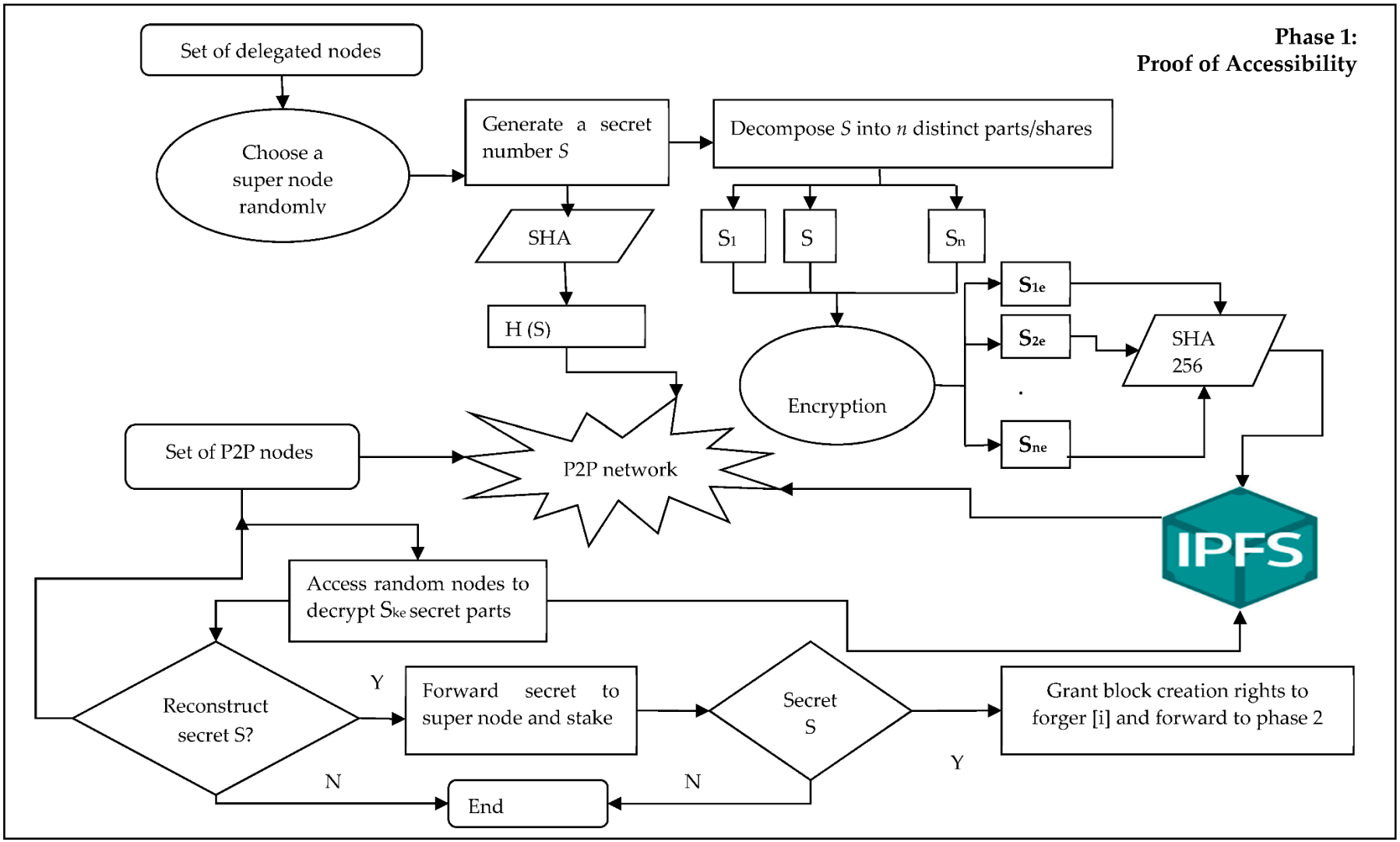

Figure 4. Proposed consensus algorithm (Phase 1).

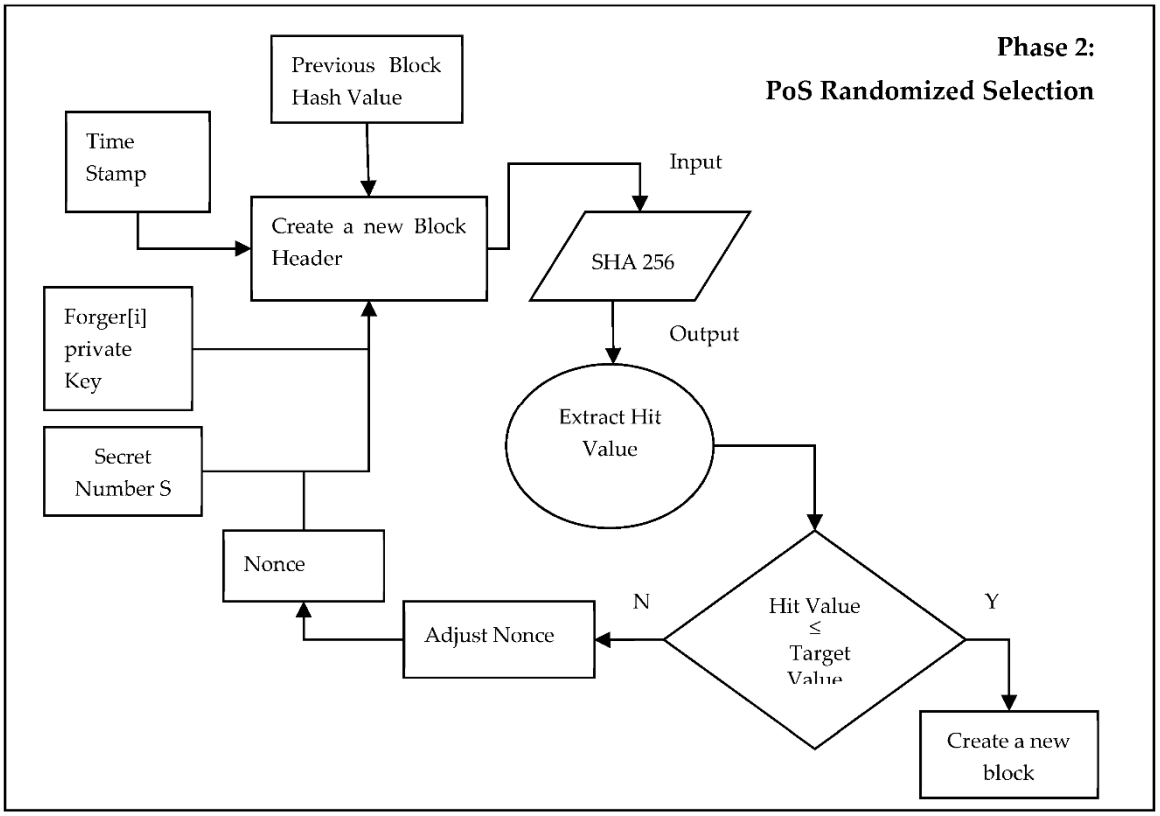

Figure 5. Proposed consensus algorithm (Phase 2).

7. Analysis of Proposed Consensus Algorithm (DPoAC)

An analysis of our novel consensus algorithm DPoAC with respect to fault tolerance, resource saving, computational complexity, and security is examined in this part.

(a) *Fault Tolerance:* A blockchain is, at its core, a distributed, decentralized system supported by a shared ledger. Although numerous consensus methods have been devised to retain consent on a consistent state of the system across all participants, there is still a potential that this consent may not be observed owing to the presence of certain faults. Fault tolerance is defined as the effort to obtain and preserve this consent in the network of potential faults.
A (k, n)—threshold secret sharing scheme has been utilized in our novel protocol that needs atleast k shares to be accessed in order to reveal a secret. Hence, the tolerance rate could be improved by limiting the access to more than k–1 distinct shares. Thus, we can select the relatively large threshold value, and consensus can still be achieved.

(b) *Resource Saving:* Block generation rights are neither directly proportional to computation power as in PoW nor accumulated stake directly as in PoS, hence nodes reduce the wastage of computational resources. In the proposed algorithm, a node has to prove the possession of shares and has to reveal the secret number to become the block generator. Hence, this protocol would not only save significant resources but also reduce the energy required to run a consensus protocol.

(c) *Computational Complexity:* The average CPU time required to recreate a secret is directly proportional to the number of shares. This is due to the fact that the time taken to encrypt each share rises in direct proportion to the amount of shares, resulting in a significant increase in computational complexity.

(d) *Security:* The proposed consensus algorithm will be resistant to various types of malicious attacks, described as below:

1. 51% attack: For the consensus protocol based on proof of effort only that is computational resources in PoW, stake in PoS, activity in PoA, etc., which creates a room for gaining access to more than 50% of these resources for a successful attack. However, in the proposed consensus algorithm, we combined two approaches that would definitely increase the cost of malicious attacks and enhance system security.
2. DDoS attack: The probability of a DDoS attack on a P2P node or delegated super node is insignificant and will not violate the protocol because of the random selection and penalty mechanism for malicious behavior induced in the proposed algorithm.
3. Sybil attack: In the blockchain system, any fraudulent node might pretend to be multiple nodes to seize control of the whole network and engage in undesired activities. Because the proposed algorithm is a combination of secret sharing and proof of stake with randomized selection, the malicious node needs to pay a stake that would be lost due to such activities. Hence, the proposed algorithm is capable of limiting such types of attacks.

8. Discussion

In this section, we analyze the compatibility of the existing consensus algorithm in IoT networks, and a comparison of these existing algorithms with DPoAC is explored.

8.1. Blockchain—IoT Convergence

Blockchain creates a P2P network that shares computing and memory needs across all network devices. Therefore, the overhead of setting up and maintaining centralized clouds, data centers, and networking equipment is minimized. A single point of failure issue can also be addressed by such a communication model based on decentralization [25]. A cryptographic algorithm as a major building block of blockchain allows this structure to possess built-in security and privacy provisions in IoT networks. Furthermore, the immutable nature of a distributed ledger makes blockchain capable of solving the data integrity issues related to IoT devices [26].

Although the built-in features of blockchain ensure security, data integrity, lack of central authority, and many more, it is relatively difficult to deploy blockchain in resource-limited IoT networks because of the reasons listed below.

1. The computational-intensive nature of the existing consensus algorithm restricts the use of blockchain in many applications including IoT/IIoT. IoT devices come with limited computation and storage capabilities and are unable to solve mathematical puzzles (as in PoW) that are required to create a new block.
2. Huge energy and resource consumption requirements of existing consensus algorithms, making the system unsuitable to use with IoT devices with limited resources.
3. IoT networks are composed of a wide range of devices that need continuous communication with other devices and fast responses. Hence, there is a need to create blocks every second, demanding low-latency consensus protocols [27].

As a result, utilizing innovative consensus mechanisms with lower processing needs, network costs, and delay is a more practical alternative for empowering integrated IoT–blockchain networks [28].

8.2. Comparison of DPoAC with Existing Consensus Protocols

Consensus protocols are major elements of any blockchain application because of the provision of the mutual agreement provided by these protocols so as to decide which block should be included in the network. The privacy and reliability offered by a blockchain system can be easily judged by the degree of privacy and reliability offered by its underpinned consensus method [28]. Throughout the years, a plethora of consensus

have been proposed and implemented specific to applications. However, conventional protocols have many shortcomings that make them unsuitable for resource-constrained IoT/IIoT networks. Therefore, a new consensus protocol is suggested in this work to overcome these limitations. In this section, existing consensus algorithms are discussed in brief, and a comparison of these algorithms with DPoAC is provided.

8.2.1. Proof of Work (PoW)

Bitcoin pioneered the use of PoW. It is the most widely used consensus mechanism. It is a computationally expensive method in which participants must solve a cryptographic problem. Many computation resources have been used by participant nodes known as miners in order to solve this problem [29]. The participant who solves this challenge first will be granted block generation privileges. The solved block is then transmitted throughout the network for validation from other participants [2]. A miner must compute a unique nonce (number used only once) value to solve the cryptographic challenge, and that nonce value must be below the specified target. The mechanism for obtaining the appropriate nonce value for a block is a mathematically difficult and time-consuming operation that can only be accomplished by miners using a brute-force approach [30]. Because miners consume a lot of resources, these nodes are also given mining rewards in addition to the transaction fee [2]. It is important to note that the mining difficulty in the network is determined by the target value. The mining difficulty level decreases with an increase in target value, whereas the difficulty level of mining increases when there is a decrease in the target value. In Bitcoin, the difficulty level is controlled in such a way that a new block must be produced every 10 min [31]. Although PoW has been shown as an efficient strategy for Bitcoin in the past, it may not be practicable for IoT networks because of high processing, energy, and bandwidth requirements.

8.2.2. Proof of Stake (PoS)

Proof of Stake (PoS) is the next extensively utilized consensus approach in cryptocurrencies following PoW. A clear distinction between PoW and PoS is that it does not cause participants to compete to solve a computationally intensive problem. This approach selects a node at random to mine the next block depending upon the accumulated stake held by that node in the network. Rather than solving a complicated hash challenge, the chosen node will use a digital signature to confirm its stake ownership. Consequently, it does not need a large amount of computer resources. However, any malicious activity committed by the selected node would cause the loss of staked coins [2]. The necessity of stake in terms of cryptocurrencies, which is not applicable in IoT systems, is the fundamental constraint of this technique to get utilized in IoT networks [28].

8.2.3. Delegated Proof of Stake (DPoS)

Although this strategy is based on PoS, there are substantial differences between the two. This process includes the majority of stakeholders electing a group of representatives known as delegates. These delegates are then in charge of network management. A delegate from the set of delegates is selected in a round-robin manner to mine the block. If a delegate is unable to produce a block within the specified deadline, the next delegate from the set will be selected. The set of delegates will be re-elected after a fixed period of time. A fraudulent delegate can be identified and removed using built-in techniques in DPoS. When compared to PoW and PoS, DPoS is a more cost-effective and high-performance consensus technique but at the cost of decentralization of the blockchain network [2,28]. DPoS is particularly fascinating for IoT systems because of these characteristics. However, the primary constraint for DPoS in IoT environments is its reliance on economic stakes to select delegates [28].

8.2.4. Proof of Activity (PoA)

This hybrid algorithm combines the features of PoW and PoS. This process works in two stages. In the first, it works similarly to PoW to solve the target hash. There are no transactions in the solved block, just the block header and winner miner's address. The transactions are then included in this block so as to move towards the second stage, i.e., PoS. The block header is then signed, and the transactions are validated by a set of validators [2]. This technique is less vulnerable to attacks, but it may result in longer delays, which may be unacceptable for time-sensitive IoT applications [32].

8.2.5. Delegated Proof of Accessibility (DPoAC)

Shamir's secret sharing method, PoS with randomized selection, and IPFS are all used in the proposed approach. This hybrid algorithm is divided into two steps. Initially, a delegated super node will be chosen at random to produce a secret and breakdown a secret into n separate shares. After encryption, these secret shares will be kept on the IPFS network. The miner nodes will compete to gain access to secret shareholders to divulge the secret. To discover the exact secret, a minimum of k shareholders must be accessible. The miner who successfully regenerates the secret is referred to as the forger node. The secret will then be sent to the delegated super node together with the staked reputation coins by the forger node. Upon verification of the correct secret, block generation rights are granted to that forger. In the following stage, PoS with randomization will be used to calculate the correct hash value by the using secret value that was just disclosed. Hash functions are appropriate for usage in several applications, including blockchain due to their lack of collision and ease of computation [33]. The forger and delegated super node will receive rewards in an 80-20 ratio for each valid block. A malicious forger node will lose all of its reputation coins and will be either prohibited from mining or removed from the network. This technique will aid in the prevention of various malicious attempts.

We employ IPFS-a decentralized P2P system idea, to store and retrieve secret shares in an IoT system, eliminating the restrictions of the client–server model and assure data privacy as well as reliable data retrieval with less latency. Hence, this suggested protocol provides a novel combination of secret sharing with IPFS and PoS to address the high resource needs, cryptocurrency reliance, and energy consumption issues encountered by mainstream consensus protocols, making DPoAC a desirable alternative for usage in IoT/IIoT networks.

In Table 1, a comparison of the aforementioned consensus algorithms is provided based on some performance parameters. Our proposed algorithm is expected to overcome the limitations of the existing consensus algorithm to prove its candidature for IoT/IIoT applicability. DPoAC would be preferable to PoW because it does not require a large amount of resources to obtain block creation rights; instead, it would only need to access a small number of IPFS nodes to reveal a secret, which would be quite cost effective. This will eliminate resource and energy consumption issues in IoT systems with limited resources. Furthermore, because it does not require a monetary stake to generate new blocks, our suggested technique is well suited to be used in the IoT as an alternative to PoS and its variations.

Table 1. Comparison of mainstream consensus algorithms against DPoAC [2,28,34–39].

	PoW	**PoS**	**DPoS**	**PoA**	**Proposed Protocol DPoAC**
Access	Public	Public	Public	Permissioned	Public
Assignment of Accounting Rights	Computing power	Stake	Stake votes	Activity based	Access to secret shares
Decentralization Level	High	High	Medium	Low	Medium

Accounting Nodes	Whole network	Whole network	Selected nodes	Selected nodes	Whole network
Delay	High	Medium	Medium	High	Medium
Throughput Capacity	Low	Medium	High	Low	High
Computing Overhead	High	Medium	Medium	High	Low
Network Overhead	Low	Low	N/A	Low	Low
Storage Overhead	High	High	High	High	Low
Scalability	Not scalable	Scalable	Partially Scalable	Partially Scalable	Scalable
IoT Applicability	Not applicable due to high resource requirement	Partially applicable due to monetized stake	Partially applicable due to monetized stake	Not applicable due to monetized stake and high latency	Fully applicable due to the use of reputation value as stake and reduced resource needs to reveal secret
Security	Vulnerable to 51 percent attack, Sybil and DDoS attack.	Less vulnerable to 51 percent attack as compared to PoW, vulnerable to Sybil and DDoS attack	Less vulnerable to 51% attack than PoW, vulnerable to Sybil and DDoS attack	Removes 51 percent attack threat, Sybil and DDoS attack due to permissioned network	Capable of eliminating 51 percent attack, Sybil attack, and DDoS attack to a great extent
Mining Rewards	In monetary terms	In monetary terms	In monetary terms	In monetary terms	In reputation value

9. Conclusions

We proposed a new consensus mechanism, DPoAC, based on Shamir's secret sharing, IPFS, and PoS with random selection in this work. The use of randomization when choosing a node to construct a secret guarantees that every node in the network is treated fairly. Furthermore, the incorporation of IPFS into the intended scheme provides data privacy and encrypted communication with reliable data retrieval in a peer-to-peer decentralized architecture. Detailed system design and algorithm flows were provided. Finally, DPoAC was evaluated on various parameters such as fault tolerance, resource saving, computational complexity, and security. The proposed system imposes less severe implications on the aforementioned parameters as compared to existing mainstream consensus protocols. A thorough discussion of existing mainstream consensus algorithms was provided, and their applicability in IoT systems was accessed. In addition, the proposed algorithm, DPoAC, was compared with these existing consensus algorithms based on a set of important performance parameters. Our proposed method is expected to achieve desired performance and accuracy.

In the future, we plan to undertake large-scale experimentation to prove the validity of our proposed protocol and compare its performance against existing consensus algorithms based on these specified parameters.

Author Contributions: Conceptualization. M.K. and S.G.; methodology, M.K. and S.G.; software, M.K. and S.G.; validation, M.K. and S.G.; formal analysis, M.K. and S.G.; investigation, M.K. and S.G.; resources, M.K. and S.G.; data curation, M.K. and S.G.; writing—original draft preparation, M.K. and S.G.; writing—review and editing, M.K., S.G., D.K., C.V., and M.S.R.; visualization, M.K., S.G., D.K., C.V., and M.S.R., Project Administration, C.V., M.S.R., and B.-C.N., Funding acquisition, C.V., M.S.R., and B.-C.N. All authors have read and agreed to the published version of the manuscript.

Funding: This project is funded by the Ministry of Research, Innovation and Digitization through Program 1-Development of the National Research and Development System, Subprogram 1.1. Institutional Performance—Projects to Finance Excellence in RDI, Contract No. 19PFE/30.12.2021,and a grant of the National Center for Hydrogen and Fuel Cells (CNHPC)—Installations and Special Objectives of National Interest (IOSIN) and also supported by Gheorghe Asachi Technical University of Iasi, Romania.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The work of Chaman Verma was supported by the European Social Fund under the project "Talent Management in Autonomous Vehicle Control Technologies" (EFOP-3.6.3-VEKOP-16-2017-00001). This paper was partially supported by UEFISCDI Romania and MCI through projects AISTOR, FinSESco, CREATE, I-DELTA, DEFRAUDIFY, Hydro3D, FED4FIRE—SO-SHARED, AIPLAN—STORABLE, EREMI, NGI-UAV-AGRO and by European Union's Horizon 2020 research and innovation program under grant agreements No. 872172 (TESTBED2) and No. 777996 (SealedGRID).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kaur, M.; Gupta, S. Blockchain Technology for Convergence: An Overview, Applications, and Challenges. In *Blockchain and AI Technology in the Industrial Internet of Things*; IGI Global: Hershey, PA, USA, 2021; pp. 1–17.
2. 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* **2021**, *9*, 80931–80944.
3. Kaur, M.; Gupta, S. Blockchain Consensus Protocols: State-of-the-art and Future Directions. In Proceedings of the 2021 International Conference on Technological Advancements and Innovations (ICTAI),Tashkent, Uzbekistan, 10–12 November 2021; pp. 446–453. https://doi.org/10.1109/ICTAI53825.2021.9673260.
4. 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 Proceedings of the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), Taiwan, China, 27–29 March 2017.
5. Bacis, E.; Facchinetti, D.; Guarnieri, M.; Rosa, M.; Rossi, M.; Paraboschi, S. I told you tomorrow: Practical time-locked secrets using smart contracts.In Proceedings of the 16th International Conference on Availability, Reliability and Security, Vienna, Austria, 17–20 August 2021.
6. Shamir, A. How to share a secret (1979). In *Ideas That Created the Future*; The MIT Press: Cambridge, MA, USA, 2021; pp. 475–478. ISBN 9780262363174.
7. de Souza, L.F.; Tonkikh, A.; Tucci-Piergiovanni, S.; Sirdey, R.; Stan, O.; Quero, N.; Kuznetsov, P. RandSolomon: Optimally resilient random number generator with deterministic termination. *arXiv* **2021**. https://doi.org/10.4230/LIPIcs.OPODIS.2021.23.
8. Goldreich, O.; Oren, Y. Definitions and properties of zero-knowledge proof systems. *J. Cryptol.* **1994**, *7*, 1–32.
9. Benhamouda, F.; Degwekar, A.; Ishai, Y.; Rabin, T. On the local leakage resilience of linear secret sharing schemes. *J. Cryptol.* **2021**, *34*, 10.
10. Xia, Z.; Gu, Q.; Zhou, W.; Xiong, L.; Weng, J.; Xiong, N. STR: Secure computation on additive shares using the share-transform-reveal strategy. *IEEE Trans. Comput.* **2021**. https://doi.org/10.1109/TC.2021.3073171.
11. Harris, C.G. Consensus-based secret sharing in blockchain smart contracts. In Proceedings of the 2019 International Workshop on Big Data and Information Security (IWBIS), Bali, Indonesia, 11 October 2019.
12. Benet, J. IPFS—Content Addressed, Versioned, P2P File System. *arXiv* **2014**. https://doi.org/10.48550/arXiv.1407.3561.
13. Chen, Y.; Li, H.; Li, K.; Zhang, J. An improved P2P file system scheme based on IPFS and Blockchain. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017.
14. Kumar, S.; Bharti, A.K.; Amin, R. Decentralized secure storage of medical records using Blockchain and IPFS : A comparative analysis with future directions. *Secur. Priv.***2021**, *4*, e162.
15. Athanere, S.; Thakur, R. Blockchain based hierarchical semi-decentralized approach using IPFS for secure and efficient data sharing. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 1523–1534.
16. 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* **2019**, *11*, 7054.
17. Zhang, E.; Li, M.; Yiu, S.-M.; Du, J.; Zhu, J.-Z.; Jin, G.-G. Fair hierarchical secret sharing scheme based on smart contract. *Inf. Sci.* **2021**, *546*, 166–176.
18. Kudin, A.M.; Kovalenko, B.A.; Shvidchenko, I.V. Blockchain technology: Issues of analysis and synthesis. *Cybern. Syst. Anal.* **2019**, *55*, 488–495.

19. 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 Trans. Ind. Inform.* **2019**, *15*, 3582–3592.
20. 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***2022**, *2*, 66–80.
21. Zhou, J.; Feng, Y.; Wang, Z.; Guo, D. Using secure multi-party computation to protect privacy on a permissioned blockchain. *Sensors***2021**, *21*, 1540.
22. Andrian, Y.; Kim, H.; Ju, H. A distributed file-based storage system for improving high availability of space weather data. *Appl. Sci.***2019**, *9*, 5024.
23. Hoogerwerf, E.; van Tetering, D.; Bay, A.; Erkin, Z. Efficient joint random number generation for secure multi-party computation. InProceedings of the 18th International Conference on Security and Cryptography, Paris, France, 6–8 July 2021.
24. Muralidharan, S.; Ko, H.An InterPlanetary File System (IPFS) based IoT framework. In Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 11–13 January 2019.
25. Miglani, A.; Kumar, N.; Chamola, V.; Zeadally, S. Blockchain for Internet of Energy Management: Review, Solutions, and Challenges. *Comput. Commun.* **2020**, *151*, 395–418. https://doi.org/10.1016/j.comcom.2020.01.014.
26. Banafa, A. IoT and Blockchain Convergence: Benefits and Challenges. *IEEE Internet Things* **2017**, *9*.
27. Alam Khan, F.; Asif, M.; Ahmad, A.; Alharbi, M.; Aljuaid, H. Blockchain Technology, Improvement Suggestions, Security Challenges on Smart Grid and Its Application in Healthcare for Sustainable Development. *Sustain. Cities Soc.* **2020**, *55*, 102018. https://doi.org/10.1016/j.scs.2020.102018.
28. Salimitari, M.; Chatterjee, M.; Fallah, Y.P. A Survey on Consensus Methods in Blockchain for Resource-Constrained IoT Networks. *Internet Things* **2020**, *11*, 100212. https://doi.org/10.1016/j.iot.2020.100212.
29. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Decentralized Bus. Rev.* **2008**, 21260.
30. Salimitari, M.; Chatterjee, M.; Yuksel, M.; Pasiliao, E. Profit maximization for bitcoin pool mining: Aprospect theoretic approach.In Proceedings of the Collaboration and Internet Computing (CIC), San Jose, CA, USA, 15–17 October 2017; pp. 267–274 .
31. Eyal, I.; Sirer, E.G. Majority Is Not Enough: Bitcoin Mining Is Vulnerable. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 436–454, ISBN 9783662454718.
32. Debus, J. Consensus Methods in Blockchain Systems. *Tech. Rep*. **2017**, 1–58.
33. Haber, S.; Stornetta, W.S. How to time-stamp a digital document. *J. Cryptol.* **1991**, *3*, 99–111.
34. Hazari, S.S.; Mahmoud, Q.H. Comparative Evaluation of Consensus Mechanisms in Cryptocurrencies. *Internet Technol. Lett.* **2019**, *2*, e100. https://doi.org/10.1002/itl2.100.
35. Yang, F.; Zhou, W.; Wu, Q.; Long, R.; Xiong, N.N.; Zhou, M. Delegated Proof of Stake with Downgrade: A Secure and Efficient Blockchain Consensus Algorithm with Downgrade Mechanism. *IEEE Access* **2019**, *7*, 118541–118555. https://doi.org/10.1109/access.2019.2935149.
36. Wang, Y.; Cai, S.; Lin, C.; Chen, Z.; Wang, T.; Gao, Z.; Zhou, C. Study of Blockchains's Consensus Mechanism Based on Credit. *IEEE Access* **2019**, *7*, 10224–10231.
37. Bamakan, S.M.H.; Motavali, A.; Babaei Bondarti, A. A Survey of Blockchain Consensus Algorithms Performance Evaluation Criteria. *Expert Syst. Appl.* **2020**, *154*, 113385.
38. Alkhazaali, A.H.; Ata, O. Lightweight Fog Based Solution for Privacy preserving in IoT Using Blockchain. In Proceedings of the 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 26–28 June 2020; pp. 1–10.
39. Akbar, N.A.; Muneer, A.; ElHakim, N.; Fati, S.M. Distributed Hybrid Double-Spending Attack Prevention Mechanism for Proof-of-Work and Proof-of-Stake Blockchain Consensuses. *Future Internet* **2021**, *13*, 285. https://doi.org/10.3390/fi13110285.