PoNW: A Secure and Scalable Proof-of-Notarized-Work Based Consensus Mechanism

Mwrwan Abubakar
School of Computing, Edinburgh Napier University.
Edinburgh, UK
m.abubakar@napier.ac.uk

Zakwan Jaroucheh
School of Computing, Edinburgh Napier University.
Edinburgh, UK
z.jaroucheh@napier.ac.uk

Ahmed Al-Dubai
School of Computing, Edinburgh Napier University
Edinburgh, UK
a.al-dubai@napier.ac.uk

Bill Buchanan
School of Computing, Edinburgh Napier University
Edinburgh, UK
b.buchanan@napier.ac.uk

ABSTRACT

The original consensus algorithm - Proof of Work (PoW) has been widely utilized in the blockchain systems and is been adopted by many cryptocurrencies, such as Bitcoin and Ethereum, among many others. Nevertheless, the concept has received criticisms over its high energy consumption. This is induced by the necessity for all nodes in the network to communicate synchronously for consensus over the ledger state to be reached. Additionally, the concept has also shown clear limitations regarding performance and throughput. In trying to rectify this issue, the paper proposes the introduction of a new hybrid consensus protocol known as the Proof of Notarized Work (PoNW). The PoNW concept reduces the number of nodes that need to achieve consensus, thereby reducing the overall energy consumption in the current PoW. In addition, we propose using a decentralized random beacon to select nodes to participate in the mining process randomly. Therefore, our algorithm promises to achieve higher scalability and consistency levels without conceding its decentralization. When this is paired with a Byzantine Fault Tolerance (PBFT) verification, the system gains the ability to replace the probabilistic finality in current PoW with absolute finality in a matter of seconds, solving the issue of scalability. Finally, the study will look into the proposed algorithm's security and provides threats model to insure an acceptable failure probability. Results from the security analysis have shown that our consensus algorithm ensures forks cannot occur, and it remains secure and consistent even amid numerous attacks.

CCS Concepts

•Computing methodologies → Distributed computing methodologies → Distributed algorithms

Keywords

Proof of Work, Proof of Notarized work, Decentralized Random Beacon

ICVISP 2020, December 9–11, 2020, Bangkok, Thailand
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8953-2/20/12…$15.00
https://doi.org/10.1145/3448823.3448875

ACM Reference format:

Mwrwan Abubakar, Zakwan Jaroucheh, Ahmed Al-Dubai and Bill Buchanan. 2020. PoNW: A Secure and Scalable Proof-of-Notarized-Work Based Consensus Mechanism. In Proceedings of *2020 4th International Conference on Vision, Image and Signal Processing (ICVISP 2020), December 9-11, 2020, Bangkok, Thailand.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3448823.3448875

1. INTRODUCTION

It was in 1993 when M. Naor and C. Dwork introduced the Proof of Work concept [1], which would later be applied on a larger scale by Satoshi to allow a distributed and trust-less consensus, at the advent of his Bitcoin Cryptocurrency in 2008 [2]. A major benefit of the POW consensus protocol is the presence of a robust algorithm that can ward off malicious participants. The concept has proved to work under being put under various tests in real-word scenarios and remains the foundation of cryptocurrencies such as Ethereum, Bitcoin, and several other blockchain applications. All the transactions taking place in a PoW based consensus is recorded, verified, and broadcasted among all the participants existing in the decentralized peer-to-peer network. In doing so, the process makes the whole system resistant, stable, and immutable. However, for this to happen, there is a need for half of the computing resources to uphold honesty. While a security property requires an honest majority to work, this can be very costly in terms of scalability, as all the participants need to be kept in the loop of what is happening and agree implicitly [2]. The rapid evolution that blockchain technologies have undergone has resulted in a growing demand for increased quality of services provided by them. This, in turn, has led to the meteoric rise of key challenges that arise during the design phase of blockchain protocols, particularly because the performance posted by the adopted consensus mechanisms will be a significant deciding factor of the blockchain network's performance in terms of network scalability, robustness to arbitrarily behaving nodes, speed of consensus finality and data consistency, etc [3]. The performance of the first generation of blockchain consensus protocols was limited by two factors; transaction throughput and the confirmation latency, which is a result of the consensus used in the blockchains that require synchronous communication for the blocks to be persistent. Therefore, clients have to wait for up to ten minutes before a transaction can be confirmed in Bitcoin and around 15 seconds in Ethereum [4]. The second generation of blockchains has later emerged as a solution to the challenges faced by the first generations of blockchain. This second-generation resulted in using the traditional Byzantine consensus algorithms, which allow for an immediate strong consistency. Since then, there has been an emergence of algorithm alternatives to PoW, such as the Delegated Proof of State (DPoS) [5] and the proof of stake (PoS) [6]. Other alternatives, such as IOTA [7], propose replacing a blockchain data structure with a Directed Acyclic Graph (DAG) data structure. However, the previously proposed approaches cannot provide a considerable throughput improvement without first is conceding with regard to other significant factors [4]. These include security and decentralization since most of the proposed approaches can guarantee maximum performance in an environment where a participant's behaviour is expected. Although the current blockchain systems that relay on Byzantine consensus mechanisms can guarantee stronger consistency in a short time, it does not scale well for a large number of nodes [8].

1.1 Sharding

With previous generations suffering from the issue of scalability, the architecture of the third blockchain generation was geared towards solving this. This generation of blockchain proposed the use of sharding, which is a prominent approach used to overcome the throughput and scalability limitations present in existing blockchain systems [9]. Sharding uses a variety of different methods to assign blockchain nodes to different groups (shards). Nodes that belong to the same shard form a committee and work in parallel to achieve consensus. As a result, this allows blockchain systems to scale to larger networks. Although sharded blockchains proving more potential compared to the traditional BFT, there was still a need to ensure the per-subchain consensus protocol runs across hundreds of participating in adversarial environments [9]. As the number of nodes achieving the consensus is minimized, the probability of an adversary being able to abort the system becomes higher. This, therefore, shows that one cannot avoid the scalability requirement of BFT consensus by simply changing the architecture.

1.2 Problem Statement

From the above, it can be noted that there does not exist a single consensus protocol able to provide all the scalability, consistency, and decentralization properties [10]. Systems based on a PoW consensus architecture fail to guarantee immediate finality due to its major scalability issue. While these systems can prevent arbitrary changes to the state by using validation, it allows for the creation of two or more valid continuations forking. Additionally, there have been known cases where the participants place preference in their own state for such purposes as performing a double-spend attack or earn a block mining reward [11]. In the same way, DPoS faces the challenge of decreased decentralization while the PoS consistency is challenged by the Nothing-at-Stake problem. It can also be noted that PBFT experiences massive network scalability problems, forcing it only to be used for consortium chains. Therefore, in this paper, we are looking to address the previously mentioned issues by developing a secure and scalable consensus mechanism that can preserve the security characteristics of the PoW consensus protocol, while also improving its scalability, and reducing its energy consumption. When making a comparison between pure consensus protocols and hybrid consensus protocols such as ours, the hybrid consensus protocols have shown more potential in terms of increased capability of an optimized decentralization, efficiency, practicality, and security.

2. RELATED WORK AND OUR CONTRIBUTION

A significant of all the researches produced recently on blockchain consensus algorithms have shown focus on addressing throughput limitations, scalability improvement, and reducing the energy consumption of the current PoW consensus protocols. For instance, authors in [12] proposed the use of a hybrid consensus protocol termed the Deterministic Proof of Work (DPoW), which promises to provide impressive consistency and scalability, with no downtime in decentralization. The proposed consensus comes in two major parts. The first part works on solving the PoW cryptographic puzzle while the second part works on verifying the proposed result's correctness. In doing so, the system provides the users with benefits associated with the PBFT and PoW protocols and often referred to as a map-reduced PoW mining algorithm. Another study introduced the Bitcoin-NG consensus protocol [13]. This protocol works on reducing the transaction's processing latency by combining PoW with Byzantine's tolerance. The main idea here is decoupling the miner election's process from transaction verification by using two different types of blocks. These include Micro blocks and Key blocks. The function of the Key blocks is to use PoW in serving as a leader selection. The leader from key blocks then assumes the responsibility of creating Micro blocks, which are crucial for transactions requiring the leader's signature without needing a power-consuming PoW. One key downside, however, is, even with that potential, the Bitcoin-NG houses a number of challenges such as history rewriting and even deliberate forking. The next consensus protocol was highlighted in [14], the paper provided a PoW consensus algorithm that allocates miners randomly into small mining pools called the distributed proof-of-work consensus. According to Cicada white paper [14], this consensus protocol uses a Distributed Hash Table to reduce storage overheads. The system then uses small amounts of energy by reducing the number of nodes being used to achieve mining when compared to the original PoW. However, the system is not without challenges as it has been criticized for experiencing difficulties in implementing the miner's selection process results [15]. Other several research efforts were geared towards finding a solution towards the key challenge of reliance on consensus algorithm by a small group of trusted replicas. One such example is the Entangled proofs of Work and Knowledge (EWoK) [16]. This algorithm divides nodes into shards. Additionally, this algorithm requires workers to store every part of the suggested blockchain data independently. While this algorithm promises to improve issues of sharding, it increases the problem of cross-sharding communication overhead. This is because miners are incentivized to store the shards locally in an attempt to gain an advantage in solving the next PoW hash-based puzzle. Another group of authors introduced the practical Proof of Kernel work (PPoKW) [17] ; it is another leaderless consensus algorithm. This algorithm is based on a low-energy PoW consensus that works to reduce the number of nodes in the PoW cryptographic puzzle and does the selection of nodes randomly to carry out the mining processes. This algorithm makes its node selection in a similar way to the approach in [18], which is based on a cryptographic sortation. However, one key criticism of this algorithm is its storage of the white list into the chain as it gives rise to scalability issues [19]. Additionally, the VRF model has to deal with the Last actor abort. This challenge encompasses a scenario where the last actor can reveal their commitment during the process of generating random value.

2.1 Our Contribution

Through this study, we have worked on developing a blockchain-based consensus protocol model, in addition to its system design and the required set of data structures. In doing so, we aim to formally study its implementation features, security-related primitives, and characteristics, which are crucial in solving the following key challenges: energy consumption, probabilistic confirmation time, scalability, and decentralization. Our contributions include

1. The construction of a new hybrid consensus algorithm that strikes a balance between the PBFT and PoW consensus mechanisms.
2. We proposed a secure random model to select participants to perform PoW to stop an adversary from concentrating its presence in one committee and exceeding the byzantine-tolerance threshold.
3. We proposed a ranking mechanism to resolve chain fork, which is based on the Pseudo-Random Process along with a permutation function to arrange selected committee members into sequential order.
4. We provided security analysis of the model together with a threat model which ensures a certain acceptable probability of failure.

3. BACKGROUND

3.1 Random Beacon

The random beacon is the source of autonomy and unpredictability in the system and is used to produce unpredictable random values. Based on the asymmetric public key cryptography concept, a digital signature produced from the random beacon is a unique and unpredictable value that can be used as a source of randomness to generate random values from it [20]. The centralized random beacon model can be susceptible to manipulation, as the signer will have control of the random beacon process, which is dependent on the signer's private key. This can affect the process of generating random values and makes it vulnerable to manipulation. Furthermore, it can also be a single point of failure. If a signer who selected by the centralized random beacon to generate the next signature is hacked or is offline, it can halt the random value process. In addition, if a malicious adversary controlled a signer node can then send conflicted random values to more than one client. To solve the previously mentioned issue, the BLS threshold signature has been used to provide a decentralized random beacon that can be operated by all the members of the threshold committee. Therefore, the decentralized random beacon can act as a trusted third party. In addition, the produced output does not need to agree on by running a full consensus. The random beacon in our consensus performs as a verifiable random function (VRF) and utilized as a method for randomness-based sharding on top of the PoW consensus protocol. The random beacon in our PoNW algorithm relies on BLS signature as introduced in Dfinity consensus [21]. The output of the VRF cannot be predicted by anyone utile released for all clients.

3.2 The BLS Signature Scheme

BLS is a unique deterministic pairing-based signature scheme introduced by Dan Boneh, Ben Lynn, and Hovav Shacham [22]. This scheme provides properties of uniqueness, non-interactiveness threshold signature, which allows a shorter threshold signature comparing to other similar approaches, where K out of N signature shares are adequate to generate a valid combined threshold signature. Irrespective of which subset is signed, it produces the same threshold signature that will be verified with the group public key. It also provides a friendly distributed key generation mechanism. Algorithms 1-5 defines these methods.

Algorithm 1: BLS Parameters

```
Two elliptic curves: $E_1$ and $E_2$
$E_1$ and $E_2$ have two elements $P_1$ & $P_2$ of prime order $p$
Two groups $G_1$ and $G_2$ of prime order $r$ on two elliptic curves $E_1$ and $E_2$
```

Algorithm 2: Generators

```
$P_1 \in G_1$
$P_2 \in G_2$
Bi-linear and non-degenerate pairing: $G_1 \times G_2 \rightarrow GT$
```

Algorithm 3: Key Generation

```
Secret key is a random bit string between 1 to p − 1 bits: $SK = x$
$SK: x \pmod{p}$
Public key: $PK = {}_x P_2 \in G_2$
```

Algorithm 4: Signature Generation

```
Input: $M$ (Message)
Output: $TS$ - the threshold signature
Sign: $SK = xM$
Message hashed: $H(M) \in G_1$
Signature: $TS = xH(M)$
```

Algorithm 5: Signature Verification

```
Input: $PK, H(M)$ and $TS$
Output: $True / False$
$\hat{}e (TS, P_2) = \hat{}e (H (M), PK)$
```

4. SYSTEM DESIGN

This section describes the proposed PoNW algorithm, which provides an energy-efficient protocol that is very robust and can solve issues of scalability and is suitable for permissioned and open blockchain. However, in this model, we propose our algorithm for permissioned blockchain models that controlled by a single federation or entity as this can be useful for a blockchain system whose applications revolve around reduced energy and faster transactions of the PoW, such as IoT. This is because IoT applications rely on a permission blockchain. However, even with the permissioned nature of the private blockchain, IoT remains prone to attacks, such as device capturing and cloning. Additionally, IoT devices are characterized by a key limitation in hardware resources and are energy-constrained. In the proposed consensus model, nodes will not be involved in the mining and verification until the random beacon mechanism selects it. This will allow IoT devices to perform their application-specific tasks while at the same time, mining blocks. We begin by describing the components of PoNW consensus.

4.1 System Components

4.1.1 Block Structure

The block in our PoNW consensus have the structure of:

$$B = (p, r, z, d, o)$$

where: p: is the previous block
r: is the round number
z: is the notarization of the previous block
d: is the data payload, a set of transactions and state
o: the block creator (owner)

4.1.2 Chain Structure

The chain *C* represents a set of a sequential order of blocks (B_0, B_1, ..., B_r) Whereas *r* is the round number of the block *Br*. The previous block is *H(Bi−1)* for all $i > 0$. The notarization of the previous block represents a valid threshold signature of *Bi−1* for all $i > 0$, Whereas B_0 represent the genesis block; B_1 is the first block after the genesis block and *Br* is the head of the chain *C*. If more than one node submitted a block, which in return produce a fork of more than one chain available: Chain 1 = *C*, Chain 2 = *C'*. Whereas the head of chain *C* is head of the chain *C'*. Then *S* is the set of blocks in the chain and *C(S)* is a chain of a set of blocks *S*, which donate the largest common prefix of chains *C(B)*, where $B \in S$.

4.1.3 Nodes

Nodes in the blockchain network *1, 2, 3, …ni* $\in N$. Each node $i \in U$, where *U* is the set of all nodes in the blockchain system. Each node *i* has a public and private key pair: *PK* indicates the node's public key, and *SK* indicates the node's private key. In a private (permissioned) blockchain model, the set of public keys for all nodes in the blockchain is known for all nodes.

4.1.4 Group

At each round, a group is created. Nodes $i \in U$ in the blockchain network are allocated randomly into a single or multiple portions where is one group forms a committee. We always have a single active group for the current round to agree on a block (notarization) and to drive the randomness process for the following rounds.

4.1.5 Byzantine Nodes

A group is fault-tolerant, and any subset of threshold size can distribute signature shares to be combined into a single threshold signature. Every member in the group can then combine the received signature shares to produce the group signature. This will produce a unique deterministic signature, which will be the same irrespective to which members signed.

4.1.6 Decentralized Notary

The block notarization process in our consensus is decentralized, which generated by all the group members. The notarization in the block is the threshold signature under a block created by a leader who selected by the random beacon from the previous round. The notary members are looking to agree on the correctness of the cryptographically solved block in the current round. The notarization is not a consensus. However, the notarization process can be used to reach consensus about a block during the normal process of the current round. Before it can consider a block as a notarized, a block needs to receive enough signature shares from the notary members. This will reduce the time need it to finalize a block, as the minimum threshold number required to sign a block will act as a Byzantine agreement. Thus, it does not need a separate consensus protocol to achieve this and provides a fast block finality at the same time of generating the random beacon (Figure 1).

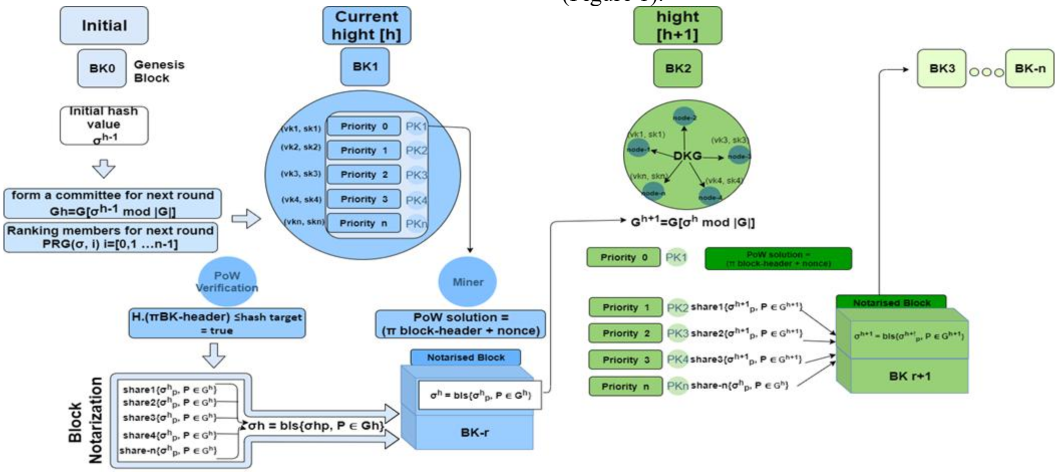

Figure 1. Proof of notarized work system model.

The blockchain system initialized with an initial hash value stored on the genesis block. The produced hash forms a random beacon σ^{h-1}, which is going to be used to select committee members for the first round $G^h = G\ [\sigma^{h-1}\ mod\ |G|]$. The DKG process leaves each member with a public verification vector and its secret key shares *(vk, sk)*. If more than one miner solves the block cryptographic puzzle, preference is given to the highest-ranked node. The notary members at the first round *r* verify that the block *Br* is solved correctly. Then they sign the block and send their secret shares to be combined in a single threshold signature to form the block notarization, which is then be used to select committee for the next round *r+1*. After that, members of the next committee sign the previous threshold signature just after beginning the new round *r+1* to produce new random beacon $G^{h+1} = G\ [\sigma^h\ mod\ |G|]$, which is going to be used to generate the following random beacon and so on.

4.2 PoNW Properties

4.2.1 Faster Block Finality

Finality is a concept that guarantees the previous transactions is irreversible, and can never change. This is a significant property, which measures the time needed to wait before it can guarantee that the transaction written in the blockchain cannot be changed. Therefore, most of the blockchain systems today can provide probabilistic finality, which cannot guarantee immediate finality. Such as in PoW which relies on the longest chain of work. Due to the competition between miners to mine a current block, it is possible to have more than one miner creating more than a block at the same height. As a result, the chain will divide it into more than one fork. Thus, to decide which chain is the valid chain from all other forks, a different fork resolution process used to choose between the forked chains. For an instant, GHOST protocol used in Ethereum [23] , and the longest chain rule is used in Bitcoin

[2] . In our PoNW the highest weight chain based on the ranking of the nodes, which is derived from the threshold signature. The node ranking process represents the weight of the nodes that can to add blocks to the chain. Therefore, this approach provides a valuable solution to select between the competed chains. In case if more than one node submitted a valid block, preference is given to the highest-ranked node.

4.2.2 Block Notarization

Our PoNW provides a fast finality by proposing the use of block notarization process similar to the one that defined in Dfinity [21]. Notarization represented as a threshold signature that generated collectively by all nodes in the notary group. This work differs from the traditional PoW, as in our PoNW, the highest-ranked chain is not based on the longest chain of work. Instead, it relays on the random beacon itself. In PoNW, the list of all active nodes in the network is known. The ranking process is driven from the threshold signature to generate an ordered list of ranked nodes that allowed to add a block to the blockchain. As a result, this will provide a secure mechanism of randomly ranking nodes based on the publicly verifiable ranking process that is driven from the distributed random beacon. Therefore, an adversary cannot interfere with the ranking mechanism, as this requires the majority to contribute to generating the threshold signature. If the notary group receives a block, they first check to see whether the block is valid or not. If the block is not valid, they discard it. The notaries will notarize the highest-ranked block if it is valid by signing it with their secret shares and broadcast it. The valid signature can be generated once the block has received a majority signature that is required for the threshold signature. This signature will represent a notarization for the block so that block can be added to the blockchain. Therefore, notarization will resolve any fork in the network, and the chain will only add the notarized blocks. As a result, this will help to achieve finality in a subsequent normal round.

A valid block proposed at the height *h* must reference a block that was notarized at *h-1*. In the current round *r*, a block *Br* will be finalized and appended to the final chain just after receiving a notarization for *Br+1*. It means that a block can be finalized after two confirmations plus the relay time as the notary can run at the same speed as the *random* beacon. Therefore, notarization will provide a fast finality in a few seconds. The many advantages offered by the BLS would perfectly justify the small degradation of performance when is compared with 10 minutes finality time in Bitcoin and 15 seconds in Ethereum.

4.2.3 How to Relay Between Committees

The unique threshold signature *ξr-1* that produced in the previous round *r-1* will be used to prioritize the nodes that are going to mine a block *Br* at the current round *r*. *ξr* is the threshold signature for the current round *r*. The notary members at the current round *r* that selected by *ξr-1* are going to verify that the block *Br* is solved correctly, and then sign it. Each member sends his signature shares to be combined in a single threshold signature *ξr*. When block *Br* received signature shares from the majority requires for the threshold, the block considered as notarized. The notarization on the block is aggregated signature from previous rounds. After that, members of the next committee sign the previous threshold signature *ξr* just after bringing the new round *r+1* to produce new random beacon output *ξr+1*, which is going to be used to generate the following random beacon. The new produced unique threshold signature *ξr+1* will then rank miners for the coming rounds and so on.

4.2.4 Random Beacon Distributed Key Generation

The random beacon provides a verifiable and friendly distributed key generation process that does not need for a trusted dealer. It allows a set of n parties to collectively generate the secret key shares and the group's public key that required for the scheme. Distributed Key Generation (DKG) algorithms is an integral part of any threshold cryptosystems, as it provides an efficient key pair (private & public) generation process that need it to initialize the threshold cryptosystem. In our PoNW consensus, we proposed using a non-interactive DKG protocol based on Gennaro, Jarecki, Krawczyk and Rabin [GJKR] protocol [24].

4.2.5 Distributed Key Generation Process

The threshold group members will generate a shared secret key without knowing the individuals and public keys. When the number of the threshold group members who agreed to sign on the message is satisfied, a new single threshold signature produced, which is the result of the combination of the signature shares of the threshold group members. Then the threshold signature can be verified by anyone who knows the group public key. As a result, each member of the group can contribute to generating a secret key that needs it for signing the group's messages. Moreover, the DKG process produces a group verification vector, which includes the public key for the group. Each member in the group can combine all the verification vectors that been received from other members to produce a single verification vector that can be used to verify a message signed by the group. Each member of the group will generate a verification vector and advertise it publicly so other members can see it. Each member will generate a secret key contribution share for other members in the group and posted to other members. Members of the group send their secret key contribution shares between each other. For the verification of shares received, each member validates the contribution share that received from other members against the verification vector of the sender who sends it and then saves it. Finally, after all the group's members receive their shares, they contribute to produce the group's secret key. The group verification vector can then, use it to derive any of the member public keys.

4.2.6 Pseudo Random Number Generation

As we discussed earlier, the decentralised random beacon will drive the process of randomly selecting nodes for the next committee. We agree that the random beacon is derived from the unique deterministic threshold signature *ξ*. Therefore, we need a PGR to generate a sequence of random values from the threshold signature *ξ*. Given that *PRG(ξ, i)* for *i = [0, 1, … n]*, the random sequence values *PRG(ξ, i)* can then be inserted as an input for a permutation function, to arrange a set of group's members into a sequential order *1,2, …,* $|U| \rightarrowtail$ *U*. An example of this, the permuted congruential generator (PCG) [25] , which provides an efficient statistical performance with a small state size. This will produce an ordered list of nodes identified by its public keys P_1, P_2, …P_n. To form the current group G_r, we need a seed *ξ* and the group size *n*. The seed will be the previous threshold signature *ξr-1*, and the group size is *n*. Members of the current group G_r for the current round *r* will be derived from the previous threshold signature *ξr-1 (mod n)*. Algorithm 1 represents the process of forming a group.

$$G_r = i_1 , i_2, \dots i_n$$
$$G_r = G_i, \; i = \xi_r (mod\, n) \qquad (1)$$

Therefore, using a pseudo-random number generation process, along with permutation function, will only allow blocks from the highest-ranked nodes to be added to the cryptography chain. The notarization from the highest-ranked node will be valid when signed with the secret shares and then can be broadcasted. In this case, the inclusion of notarization helps resolve any forks in the network, and only a notarized block will be added to the chain. From the evaluation, it is clear that our algorithm offers a higher level of security against chain forks.

5. SECURITY ANALYSIS

This section focuses on carrying out an analysis of the security of our solutions in regards to the model previously highlighted under section 4. In the analysis of the decentralized random beacon, the main assumption is that the model uses a cryptographically strong pseudo-random generator in the system's genesis to generate the initial seed. In the case that a central system authority creates the Genesis Block, the system can be used in creating the requisite seed for generating it from a source characterized by high entropy. However, it should be noted that the model shall not set up a threshold signature scheme by relying on a trusted third party. Therefore, in this case, the group *G* shall set up the group public key and the secret key shares by running a DKG for the BLS, when initializing the blockchain system. Lastly, the signing process shall be repeated in non-interactive mode.

In this section, the focus will be on potential factors that can be used by the adversaries to attack the proposed system together with ways to mitigate the occurrence of such threats. In terms of the security model, the key assumption is that honesty is maintained by at least two-thirds of the nodes. Therefore, if more than a third of the nodes are faulty, the algorithm fails to reach a consensus. For this system, the maximum number of nodes before breaking the consensus is 33% and could be made up of comprised nodes or offline nodes. For example, considering the assumption of the BFT mechanisms, in a network with 1000 nodes, it requires no more than 333 nodes are faulty in order to the blockchain system to be considered as a safe. In the case that the consensus nodes are divided into four shards, the consensus nodes will be divided into the quarter with each group assigned 250 nodes. Achieving a consensus, in this case, will require the group to work in a parallel fashion. Therefore, an adversary will only need 83 nodes to fail a consensus. This shows that sharding reduced the system's fault tolerance from 333 nodes to 83 nodes. However, modern-day technology has made it possible for sharding techniques to rely on a sort of randomness in assigning the nodes to their shards, reducing the probability of all 83 nodes being in one shard. In the case that the adversary controls 250 of all nodes in the system, there is a high possibility that all 83 malicious nodes out of the 250 will be in one shard. The previous assumption requires a higher number of nodes in each shard in reducing this high probability. It is a trade-off between the minimum number of nodes and security per shard. Therefore, while having a large number of shards with a reduced number of nodes improves a system's throughput, it increases the probability of having a shard compromised by malicious nodes.

5.1 Threats Model

In analyzing our protocol, the attack shall be deemed as originating from an adversary that has control over a certain fraction of all participant's machines. The underlying assumption is that the adversary's probability of break cryptographic primitives is negligible. This is because, with the number of nodes undertaking in consensus reduced significantly, the probability associated with aborting the algorithm increase. The good thing is that the number can undergo optimization to strike a balance between reliability and performance. The model showed that the security of the PoNW consensus algorithm is upheld only after the bounds highlighted under equations (5) is upheld. The bounds are maximal, and the network may prove to be much secure when subjected to lesser stringent conditions. We begin assuming *f(G)* is number of Byzantine nodes in a group *G* and *n* is the group size, we have Assumption 2, where *B > 2*:

$$|U| > B\, f\,(U) \tag{2}$$

And assumption 3:

$$n > 2f\,(G) \tag{3}$$

Each group *G* in the system represents a random sample of all the nodes in the blockchain system *U*. Given Assumption 2, each group *G* is honest, and each group has a fixed size of *n*. To calculate the probabilistic of *G* honest we used the formula:

$$X \sim Hypergeometric\,(N, K, n) \tag{4}$$

Formula 4 gives a random variable distributed hypergeometrically with the elements of the population given as *N*, *K* and *n*. It has a probability calculated using Formula 5.

$$\underset{x}{\mathrm{P}}\,(k) = \underset{r}{\mathrm{P}}(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}} \tag{5}$$

Formula 5 signifies the probability function of the hypergeometric distribution, and where:

N: is the population size.

K: is the number of success states in the population.

n: is the number of draws.

k: is the number of observed success.

(*ab*) is a binomial coefficient.

The function is positive when:

$$Max\,(0, n + K - N) \leq K \leq Min\,(K, n) \tag{6}$$

Regarding to the hypergeometric Distribution formula, all items of the population is sampled and the result of the draws is classified. In our example, a group is drawn from the total number of publications without replacement. To demonstrate this, we used the hyper-distributed probability code, a Python program developed by Tari labs available in GitHub [26].

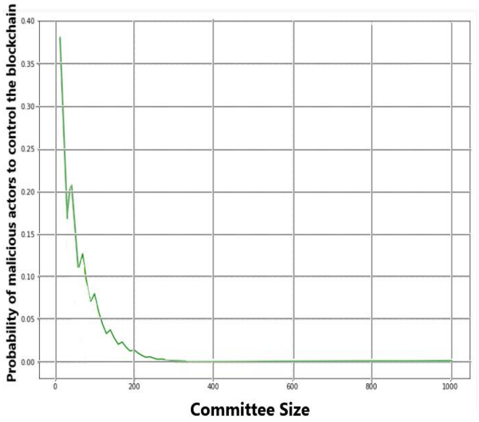

Figure 2. Committee size against probability with the total publication size of 1000.

Figure 2 showed the committee size against the probability of the malicious adversaries to control the blockchain system, with the total publication size of 1000. It demonstrates a lower probabilistic concerning the size of the committee. It can be seen that a lower probabilistic when a committee size is 300 nodes or higher with the elements of the BFT threshold given as:

N: is the population size = 1000

K: is the number of success states in the population is 60.

n: is the number of draws (committee size from 1 to 1000).

k: is the number of observed success. This donates the BFT threshold, which assumes two-thirds of the nodes are honest, which is 67%.

5.2 Possible Attacks

It is crucial that attention is paid to the functioning of the consensus model under both normal and adversarial conditions. For such an environment, the consensus mechanism has to be prepared in dealing with the following attacks.

5.2.1 Randomness Manipulation Attacks

The randomness generation process is one that is prone to frequent attacks. One such attack is the randomness manipulation attack. While using a proof-based consensus protocol to generate randomness, the generated randomness can be manipulated by any insider malicious attacker who can either withhold valid blocks or refuse to mine. This can force the system to rely on a single source in the generation of random beacons. In such a case, the random value process tasked with the generation of random beacons can be halted in the case that a signer selected by the random beacon offline or is hacked. Additionally, a malicious adversary can send conflicting values to various clients when he or she gets control of a signer node. In the attempt to prevent attackers from manipulating values that are generated from the random generation process, the PoNW switched to using a decentralized random beacon to generate randomness. For our model, we decided to rely on a BLS threshold pairing signature scheme as the default random beacon. By using this, the model is guaranteed of a stable decentralized random beacon that is difficult to manipulate, as it requires a minimum number of the threshold members to be generated.

5.2.2 Chain Fork

Our PoNW leverages a permutation function with a Pseudo-Random Process in an attempt to sequentially arrange the selected committee members. In doing so, the algorithm allows for the selection between the competed chains. This is an outstanding breakthrough as the proposed PoNW consensus algorithm provides a solution to issues of forking by proposing the use of a ranking process stated in section 4.2.6. In case more than one node submitted a solution for the block puzzle, the algorithm will add blocks that mined by the highest-ranked nodes in the cryptography chain. The notarization of the highest-ranked nodes done by the notary nodes of the block will be valid when signed with the secret shares. In this case, the inclusion of notarization helps resolve any forks in the network, and only the notarized blocks will be added to the chain. From the evaluation, it is clear that our algorithm offers a higher level of security against the mentioned attacks.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the PoNW consensus algorithm, which is a hybrid approach based on a reduced mining algorithm combined with a PBFT verification. Our protocol has shown the potential to achieve a high level of consistency and security by using a decentralized random beacon, which acts as a Verifiable Random Function (VRF) that requires the contribution of a majority of the group members by sending their signature shares to be used in the production of a unique, unpredictable, and deterministic threshold signature. The system then proceeds to use the threshold signature in carrying out the node selection required for the next group. The study has also provided an analysis of the consensus protocol's security model, together with estimations regarding the probability of an adversary controlling the consensus mechanism. The analysis showed that the PoNW is resistant against the 51% attack and also increased this threshold by 66.6%, which achieves great levels of consistency and greater security in maintaining decentralization. It is our belief that the PoNW is a representation of a major step towards the development of more secure decentralized applications. The low latency achieved by the algorithm allows for a myriad of applications, which were complex or impossible to achieve with previous latency consensus methods. We hope that this study shall be a source of motivation for further research into this field. In our current design, we provided a consensus algorithm of principle. Therefore, for future work, we are planning to implement the PoNW consensus in a subsequent practical system and evaluate it in big data scenarios for large scale networks. Our other future works include a comparative study of various benchmark security solutions in large scale networks, as well as evaluation of different threat attack scenarios.

REFERENCES

[1] C. Dwork and M. Naor, "Pricing via Processing, Or, Combatting Junk Mail, Advances in Cryptology," CRYPTO'92: Lecture Notes in Computer Science, pp. 139-147, 1993.

[2] S. Nakamoto, "A Peer-to-Peer Electronic Cash System," 2008.

[3] C. D. I. E. A. E. G. A. J. A. K. A. M. P. S. E. S. E. G. S. D. S. R. W. Kyle Croman, "On Scaling Decentralized Blockchains," in International conference on financial cryptography and data security, 2016.

[4] N. Z. W. L. Y. T. H. Yang Xiao, "A Survey of Distributed Consensus Protocols for Blockchain Networks," IEEE Communications Surveys and Tutorials, 2020.

[5] I. Rõžakov, "A Modest Comparison of Blockchain Consensus Algorithms," 2019. [Online]. Available: http://essay.utwente.nl/78909/.

[6] S. N. Sunny King, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," August 19th, 2012. [Online]. Available: https://decred.org/research/king2012.pdf.

[7] S. Popov, "The Tangle," 2016.

[8] A. Poelstra, "Distributed Consensus from Proof of Stake is Impossible," 2014.[Online]. Available: https://pdfs.semanticscholar.org/9d1c/b62f7fde9b03882567ee01adba31b7108276.pdf.

[9] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin and B. C. Ooi, "Towards Scaling Blockchain Systems via Sharding," in SIGMOD '19 Proceedings of the 2019 International Conference on Management of Data, Amsterdam, Netherlands, 2019.

[10] L. a. H. M. Ismail, "A Review of Blockchain Architecture and Consensus Protocols: Use Cases, Challenges, and Solutions," Symmetry 11.10 (2019): 1198.

[11] I.-C. L. a. T.-C. Liao, "A Survey of Blockchain Security Issues and Challenges," International Journal of Network Security, Vols. Vol.19, No.5, pp. 653-659, Sept. 2017.

[12] G. W. H. W. M. Z. L. Z. a. Q. C. Zhuan Cheng, "A New Hybrid Consensus Protocol: Deterministic Proof Of Work," Semantic Scholar, 2018.

[13] A. E. G. E. G. S. a. R. v. R. Ittay Eyal, "Bitcoin-NG: A Scalable Blockchain Protocol," in 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16), Santa Clara, CA, USA, March 16–18, 2016.

[14] "Cicada: A Distributed Direct Democracy and Decentralized Application Platform," 28 11 2016. [Online]. Available: http://iamcicada.com/whitepaper/.

[15] T. L. Cedric Sanders, "Knowledge Discovery on Blockchains: Challenges and Opportunities," arXiv preprint arXiv:1904.07104 (cs.DC), 2019.

[16] J.-M. B. G. O. K. W. L. Frederik Armknech, "Sharding PoW-based Blockchains via Proofs of Knowledge," IACR Cryptology ePrint Archive, vol. 2017, p. 1067, 2017.

[17] D. J. B. M. H. S. J. L. K. S. S. Leif-Nissen Lundbæk, "Practical Proof of Kernel Work & Distributed Adaptiveness," Yellow Paper: Version 1.2, 2018.

[18] R. H. S. M. G. V. N. Z. Yossi Gilad, "Algorand: Scaling Byzantine Agreements for Cryptocurrencies," in SOSP '17: Proceedings of the 26th Symposium on Operating Systems Principles, October 2017.

[19] C. S. a. T. Liebig, "Knowledge Discovery on Blockchains: Challenges and Opportunities," arXiv:1904.07104 [cs.DC], 2019.

[20] L. T. A. N. B. R. P. H. B. John Kelsey, "A Reference for Randomness Beacons: Format and Protocol Version 2," NIST Internal or Interagency Report (NISTIR) 8213 (Draft). National Institute of Standards and Technology, 2019.

[21] M. M. D. W. Timo Hanke, "DFINITY Technology Overview Series, Consensus System," arXiv preprint arXiv:1805.04548, 2018.

[22] D. B. L. Shacham, "Short Signatures from the Weil Pairing," in International Conference on the Theory and Application of Cryptology and Information Security, 2001.

[23] Y. S. a. A. Zohar, "Secure High-Rate Transaction Processing in Bitcoin," in International Conference on Financial Cryptography and Data Security, 2015.

[24] S. J. H. K. T. R. Rosario Gennaro, "Revisiting the Distributed Key Generation for Discrete-Log Based Cryptosystems," RSA Security'03, 2002.

[25] M. E. O'NEILL, "PCG: A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation," ACM Transactions on Mathematical Software , 2014.

[26] H. Kevoulee, 2019. [Online]. Available: https://github.com/tarilabs/modelling/blob/master/utils/hyper_dist_prob.py.