THE TEMTUM CONSENSUS ALGORITHM – A LOW ENERGY REPLACEMENT TO PROOF OF WORK

Richard Dennis and Gareth Owenson

Department of Computing, University of Portsmouth,
Portsmouth, United Kingdom

ABSTRACT

This paper presents a novel consensus algorithm deployed within the Temtum cryptocurrency network. An overview of the proof of work consensus algorithm is presented, and gaps in the research are outlined. The Temtum consensus algorithm's unique components, including the Node Participation Document (NPD) and the use of the NIST randomness beacon, are outlined and explained. Comparisons on the cost to attack the consensus algorithm and energy consumption between the Temtum consensus algorithm and Bitcoin's proof of work is presented and evaluated. We conclude this paper summarising the findings of the research and presenting future work to be conducted.

KEYWORDS

Blockchain, Peer-to-Peer Networks,Cryptocurrencies, Consensus, Byzantine Fault Tolerance

1. INTRODUCTION

A person under the pseudonym Satoshi Nakamoto emailed a cryptography mailing list; a self-pushed paper titled, Bitcoin: A Peer-to-Peer Electronic Cash System [1]. The paper contained a novel approach to a digital currency without a third party or centralized entity requirement. Through the implementation of the blockchain and the proof-of-work consensus algorithm, the previous double-spend attacks were solved.

The proof of work algorithm is arguably the most innovative component outlined in the Bitcoin whitepaper. This algorithm enabled users on the network to be confident the token received from another user has not been previously spent. This was achieved through a globally agreed state of all transactions called the blockchain.

Bitcoin is the decentralized peer-to-peer network that was created from the Bitcoin whitepaper. Further, Bitcoin can also refer to the token, which is transferred between users on the Bitcoin network.

Due to the peer-to-peer architecture of the Bitcoin, there are no centralized components to the network [2]. Instead, users can participate in the network by downloading the Bitcoin client and donating resources to the network. A machine participating in the network running the Bitcoin client is defined as a Bitcoin node.

David C. Wyld et al. (Eds): BIBC, AISO, EDTECH - 2020
pp. 01-15, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101301

The node architecture contains four components. A node can run none, one, or multiple instances of each component. A full node can be defined as a node that runs at least one instance of each component [3]. These components are; Routing mechanisms for the Bitcoin protocol, A wallet, A complete blockchain since the network launch, and the Bitcoin miner.

The blockchain can be defined as a globally agreed network state of transactions on the network since it was deployed. The blockchain achieves a global state by limiting the ability to amend transactions to the blockchain to a single node that has completed the proof-of-work algorithm before the rest of the network's nodes.

Each node on the network confirms transactions that have occurred on the network. Validtransactions are combined into a data set called blocks. A block contains all valid transactions on the network since the publication of the previous block. In addition to the valid transactions, each block contains a previous block's hash to prevent modification of previously confirmed blocks.

The blockchain, also with the proof of work algorithm, prevents the double-spend attack. This attack is when an adversary attempts to spend previously spent tokens. In addition to the inability of an adversary to rewrite previously confirmed blocks, the globally agreed state of the blockchain ensures as long as a majority of miners are not malicious, the network is considered secure.

The mining algorithm is a full node component that completes a brute force calculation to find the solution to a preset mathematical problem [4]. This problem is defined as the SHA256 hash of the block to be confirmed, which, when combined, a random value results in a value lower than a target value [5]. The target value changes every two weeks to ensure the generation of blocks occurs on average every 10 minutes.

Each miner is conducting this algorithm and competing against all other miners on the network to be the first with a correct solution [6]. This brute force method and competition between miners incentives nodes to add more powerful CPUs or ASICs to the network. The increased performance of the CPUs and ASICs also increases the energy consumption of such devices.

Furthermore, incentives miners to donate their CPU resources to the mining process by rewarding the node that finds the valid value first a reward of newly minted Bitcoin and all the transaction fees collected during the block.

Once the block is valid, and proof-of-work is successfully found, the block is defined as mined. The block then is propagated through the network to enable each node on the network to receive this confirmed block and update the nodes' locally stored blockchain.

2. LITERATURE REVIEW

2.1. The Byzantine Generals Problem

The Byzantine Generals Problem is a description of a known problem in computer science. A situation where all involved participants must agree on a single version of an event to prevent complete failure [7]. There is an assumption that half or less of the involved participants are malicious and attempting to disrupt the event by propagating false information or not propagating data. Furthermore, participants do not know if the messages are authentic or follow the correct procedure.

This problem can be demonstrated in a decentralized peer-to-peer network where a file is propagated through the network. Users participating in the network cannot trust the received file due to their inability to determine which node is malicious and sharing malicious files. Due to the lack of guarantee that nodes on the network are participating following the protocol rules, malicious nodes may participate in the network and attack the network.

Previously described digital currency peer-to-peer networks such as e-cash were not secure against this type of attack; therefore, users could spend the same currency multiple times [8]. This problem was overcome with the inclusion of centralized ledgers.

The Bitcoin network is the first known peer-to-peer decentralized digital currency, which prevents the Byzantine generals problem without a third party's requirement. This was achieved due to the novel components of the blockchain and proof-of-work algorithm. All blocks are cryptographically verified by each node on the network, validating the contained hash and a nonce.

While the hashcash network previously detailed the requirement of hashing data on the network, Nakamoto expanded on implementing such a method to create the proof-of-work algorithm [9][1].

Each full node on the network competes against all other nodes on the network to find a random nonce value that results in a hash below a presettarge value when combined with the block hash. Each block contains the previously confirmed block hash within the body of data. Therefore it is impossible to modify a previously confirmed block without changing its hash [10]. Therefore for any modification of a previous block, the hashes for all blocks since this block would also be required to be recalculated. Failure to do so would alert the node of such an attack.

Since the nodes conducting the mining algorithm compete against each other, the network is considered secure so long as more than 50% of nodes correctly follow the protocol rules [11]. Therefore we can conclude Bitcoin is secure against the Byzantine generals problem.

2.2. Mining

As previously discussed, the mining algorithm is used within the Bitcoin protocol to ensure that computational resources must be donated to the network before a block of transactions can be considered valid.

The mining algorithm ensures the generation of average every 10 minutes by modifying the pre-determined value the block hash, and the nonce, when combined, must be lower than. This process ensures only one block of data is valid and accepted by the network [12]. A globally agreed state of all transactions that have occurred on the network prevents an attacker from spending the same Bitcoin more than once.

The mining algorithm's brute force process requires a random number defined as a nonce to be calculated and combined with the block hash until the hash value contains a pre-determined number of zeros at the start of the hash. This requires each node to calculate billions of nonces, and a node which can calculate nonces quicker than other nodes on the network increases their probability of finding a valid result.

As of 2017, the requirement was for a valid hash to start with 17 zeros; this results in a probability of $1.4X10^{20}$ to find a successful nonce [13].

This requirement of the number of zeros at the start of the hash is defined as the network difficulty. This value is adjusted every 2016 blocks to keep blocks being confirmed on average every 10 minutes. This is required due to increasing resources added to the network by nodes wishing to obtain a greater advantage over other nodes, which results in blocks being computed quicker than this timeframe.

A target 256-bit number is encoded in the block header's nBits field, and it has a maximum value of 0x1d00ffff (≈ 2224). The difficulty can be summarized at the ratio of the maximum target over the current target. D ≈ 2224/target.

Due to the SHA-256 hashing algorithm properties, results are truly random and expensive to compute but are deterministic outputs and cheap to validate [14].

The mining process can be conducted without additional data from other nodes. Therefore this can be considered a genuinely decentralized component of the Bitcoin network, requiring only a valid blockchain to participate.

The computational resources of a node are directly proportional to the time required to find a correct solution. The more computational resources the node processes, the more nonces per second the node can test.

While nodes on the network are adding more and more computational resources to give them a competitive advantage over other network nodes, the consistent publication of blocks every 10 minutes has led to a situation where nodes are required to add computation resources in order to maintain their competitive constantly. This, in turn, increases each node's energy consumption for no greater performance on the network.

The miner responsible for finding a valid nonce to the proof-of-work problem is compensated with a block reward and all fees from the confirmed block's transactions. The block reward was originally 50 BTC per confirmed block; this reward is reduced by half roughly every four years.
Due to the increased probability of finding a valid nonce being directly proportional to a node's computational resources, nodes are combined their resources. This merger of computation resources has created so-called mining pools where thousands of nodes combine their resources to have a greater probability of finding the solution to the nonce and receiving the rewards [15]. Due to this shift to centralized mining pools, it is now statistically impossible for a single node to participate in the mining process and expect to find a valid block.

Therefore, even though the mining algorithm was intended to be a decentralized method to confirm transactions on the network, it has morphed into a centralized confirmation mechanism. Furthermore, due to the financial incentives and limited space within a block, transactions that pay the highest transaction fee are more likely to be included.

This demand for resources has impacted the mining process's energy consumption and excluded the home user from participating without joining a mining pool.

2.3. Mining resource consumption

Due to the direct link between computational resources and the probability of success and the hashing algorithm being deployed at the hardware level, Bitcoin ASICs miners have created. These are hardware devices whose only function is to calculate the required nonce for the proof-of-work algorithm.

The ASICs miners are optimized for the proof-of-work algorithm and can conduct more nonce attempts than the average home computer initially used for the mining process [16]. However, the ability to conduct more proof-of-work algorithm attempts results in more energy being required for the device to function.

A Bitcoin ASIC miner typically continuously runs until the cost of electricity and block reward makes it uneconomically viable or the hardware malfunctions.

As users add more ASICs to the network to increase their probability of finding a valid solution, the network difficulty increases, and the network's energy consumption.

This has resulted in a catch-21 situation, where a user must donate more computing resources to the network to stay competitive, which in turn consumes more energy for no greater performance on the network. ASIC development focuses on increasing nonces per second each device can achieve rather than energy efficiency, resulting in significant growth of the Bitcoin network's energy consumed.

In 2016, the Bitcoin network consumed 0.08% (67.86 TWh) of electricity consumed globally per year. Compared to the Visa network, which in 2016 consumed 674,922 Gigajoules of energy and processed 111.2 Billion transactions, averaging 8,000 transactions per second compared to Bitcoin’s five transactions per second [16][17]. This demonstrates a single Bitcoin transaction is compared in energy to 100,000 transactions conducted on the Visa network.

Due to this significant energy consumption for such a low-performance network, law and policymakers are currently debating regulation on how to reduce Bitcoin’s energy consumption[18].

While it can be argued that users' ability to send transactions without a third party is an approximate use of the energy consumed by the network, in a world focused on reducing carbon emissions and energy consumption, it identifies the mining algorithm requires reinvention to a lower energy consumption algorithm.

2.4. 51% Attack

The foundations of a double-spend attack require a malicious adversary to control 51% or more of the network's hashing power [19]. This is an advancement to the Sybil attack, which decentralized peer-to-peer networks are vulnerable to. This is due to a lack of restrictions on users, which can participate in the network. Furthermore, even is a malicious adversary is detected by other nodes on the network, the lack of centralized authority to remove the nodes means they will always be able to participate in the network.

A malicious adversary controlling 50% of the networks hashing power would produce valid blocks simultaneously as an honest network. This would cause a situation where two valid versions of the blockchain exist simultaneously, a problem known as a network fork. Therefore, one version of the blockchain can contain transactions that are not contained in the other blockchain, enabling the attacker to spend the same Bitcoin on both blockchains.

The greater the computational resources under the control of a malicious adversary, the greater they can modify previously confirmed blocks. With more than 50% of the network hashing power, the attacker would be able to modify a previously confirmed block, for example, by removing a transaction within it and then recalculating the proof of work for the new block and all subsequent blocks [20].

Furthermore, an attacker could also disrupt the network by refusing to forward transactions or blocks to the rest of the network and flooding the network with invalid data.

The 51% attacks have been demonstrated to occur in the real world, with two Ethereum based networks, Krypton, and shift, coming under attack during 2016. Furthermore, several other networks have been attacked since 18 million USD of Bitcoin Gold currency was double-spent during May 2018.

3. Temtum components

This section will outline the core components of the novel consensus algorithm deployed in the Temtum network. We will detail the node participating document, including how it interacts with the nodes on the network and prevents malicious modification. Furthermore, the consensus algorithm itself is described in this section.

3.1. The node participation document (NPD)

Within the Temtum network, there is a subset of nodes classified as authority nodes. These nodes participate in the network as normal nodes while also monitoring nodes participating in the network.

To enable the monitoring of nodes on the network, all nodes must announce themselves to the authority nodes when they first join the network. The authority nodes would request data from the nodes to identify them and add them within the NPD.

The requirement of DNS nodes within the network is no longer required due to the authority nodes replacing these nodes and providing newly joining nodes with a global view of the network through the NPD publication.

The NPD can be defined as a document updated on an hourly rate to provide all nodes on the network a global state of nodes participating. All nodes that have been connected to by the authority nodes are defined as currently actively participating in the network and therefore are listed in this document.

The ability of nodes to have a global view of the network participation will reduce the probability of a network-level attack such as a partition attack. The partition attack excludes nodes from participating in the honest network allowing for attacks such as delaying or not forwarding blocks and transactions. Furthermore, it would be possible for the partition node to receive a different version of the blockchain, allowing a double-spend attack. This attack and resistance to attack are detailed later in this paper.

Multiple authority nodes are deployed on the network to prevent censorship or attack caused by a single malicious adversary in control of the authority node, each operated by a separate entity.
For a single NPD to be published, and agreed NPD state must be agreed amongst a majority of authority nodes. They achieve this by voting on the inclusion of each update of the NPD. While the authority nodes can add nodes to the NPD, they may only remove offline nodes from the NPD and not nodes that they suspect to be malicious to prevent censorship.

Data such as the IP address, the amount of blockchain history locally stored, public identity, and the node's role is stored within the NPD.

Figure 1 is an example of the data fields that are collected and stored in the NPD for each node

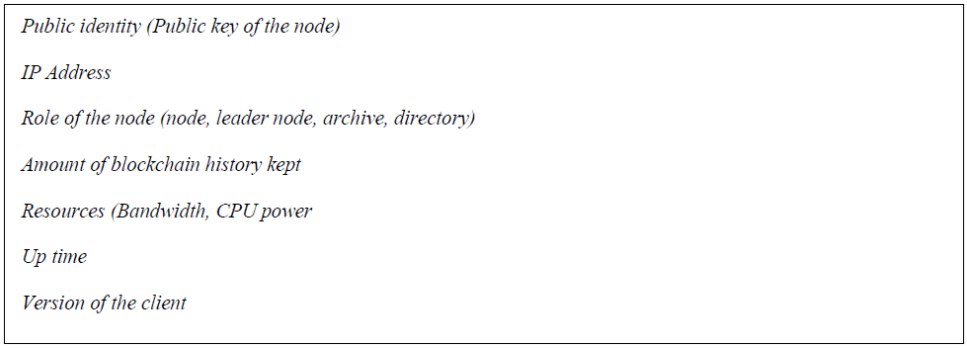

Figure 1. Data stored within the NPD for each node

Due to the components of the temporal blockchain deployed within the Temtum network in which a finite amount of the blockchain history is stored locally, the NPD provides information to nodes on the network to enable the querying of nodes for data which they do not hold locally. Utilizing the NPD nodes that require data can quickly and accurately determine the correct node to query to prevent wasted resources querying all nodes randomly on the network, as is Bitcoin's case.

During the bootstrap process, where a node is attempting to join the network, they must initialize contact with a minimum of one authority node. An “announcement” message is sent from the node bootstrapping to one or more authority node to achieve this. This message contains data to enable the authority nodes to connect back and further query the node.

Table 1. Annoucement message.

Size	Field	Description
32 bytes	Public identity	The public part of the node’s public/private key pair
4 bytes	IPV4 Address	The IP address of the node
2 bytes	Port number	Port number the node can be connected to
4 bytes	Timestamp	Time this block was created (seconds from Unix Epoch)

When an authority node receives this message, they attempt to connect to the node using the provided information and return an “ACK” message. Furthermore, the authority node also returns the last valid NPD; however, the bootstrapping node would not be included in the previous NPD. Within the header of the NPD, a valid time period from and until is defined. This prevents nodes from participating in the network using an out of date NPD. A node attempting to use an older NPD may have a conflicting view of the network, which could be exploited. The NPD is also signed by the authority nodes' private keys to prove the NPD publication source.
Figure 2 displays an example NPD

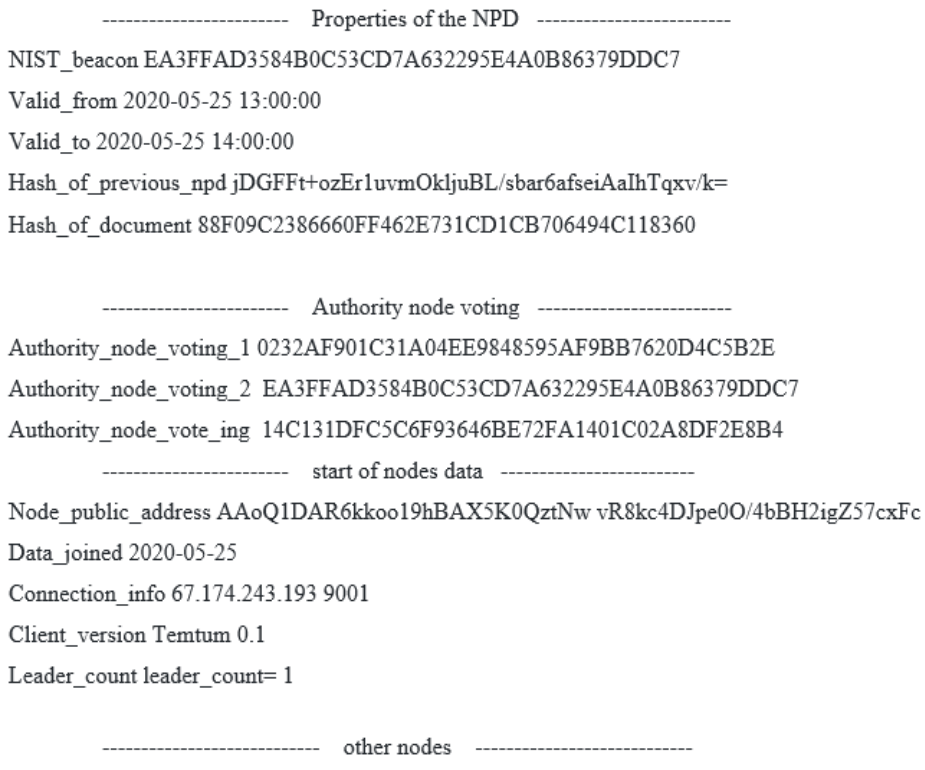

Figure 2. An example NPD document

The NPD is published on an hourly basis. This was calculated as appropriate publication time due to the +4 and -3 network churn, which was observed hourly on the Bitcoin network between 01/05/2015 to 01/05/2018. Furthermore, this approach also is in use on the Tor node document.
A MacBook Pro with 16GB RAM, 150mbit/s available bandwidth, and a 2.8 GHz Quad-Core Intel Core i7 CPU was used during our simulated experiments.

A node entry within the NPD uses 4.49bk of storage, and it was simulated an average authority node would be able to store 481,737 nodes within a single NPD.

Bitcoin currently uses a DNS method to provide bootstrapping nodes with known nodes on the network. It was currently observed 32 nodes per request were received; however, the maximum theoretical IP address to be received per request is 65535.

A vital requirement of the Temtum network is lower-resourced users' ability to participate fully in the network. Therefore a simulated NPD was created and calculated the number of nodes able to be contained within the NPD for a node to download the NPD within a 5-minute window.

A node with an average bandwidth of 1,103 kb/s would require an NPD to be no larger than 0.3309GB, which would, in turn, be able to contain 73,697 node records.

These numbers demonstrate that the NPD structure scale is far in excess of the current Bitcoin network size.

3.2. The consensus algorithm

The proof of work algorithm deployed within the Bitcoin network can be summarised as an expensive operation to determine the node which will be permitted to append a block onto the blockchain.

It was outlined that the resources processed by a node are directly proportionate to the probability of the node finding a correct solution to the proof-of-work algorithm. However, this opens a vector of an attacker where a malicious adversary can pool network resources to enable them to generate blocks at the same speed, or faster than the honest network. This is known as the 51% attack. The ability to generate blocks at the same speed as the honest network would result in two valid blockchains existing on the network at the same time.

A 51% attack does not impact the consensus algorithm deployed in the Temtum network. The Temtum consensus algorithm's unique property is that each node can locally determine which node will be confirming the next block utilizing the NPD data already stored. Since all nodes on the network have the same NPD and conduct the same algorithm, all nodes on the network would select the same node for confirming the next block.

Therefore there can be no fork of the blockchain existing on the Temtum blockchain, making a double-spend attack impossible on Temtum.

The removal of the competition between nodes and the single calculation process reduces the number of computational resources required. Furthermore, since additional resources do not impact the probability of the node being selected, there would be no situation in Bitcoin currency where more computational resources are required to maintain competitiveness.

This property also removes the incentives for nodes to pool their resources together, making the block confirmation process more decentralized than Bitcoin.

The consensus algorithm proposed here will enable the node responsible for confirming the block pre-event rather than Bitcoin's post-event model.

The consensus algorithm:

- Each node can determine the node responsible for confirming the next block using local data but reach a global consensus.
- Two nodes cannot publish blocks at the same time, preventing a fork in the blockchain.
- A single calculation process is conducted,which requires low computation resources.
- Use of a randomness beacon.

The randomness beacon is present to the network from NIST every 60 seconds. Each node would be required to ensure they have downloaded the latest beacon. Each node then performs the same calculations on their NPD to determine the next node to confirm a block.

The NIST beacon 512-bit value is subtracted from the public identity of a node contained within the NPD. This is repeated until every node’s value contained within the NPD has been calculated. The node that results from a closet to zero would be the node responsible for confirming the next block.

```
512 Bit Randomness beacon broadcast to the internet
All nodes retrieve the beacon value
Nodes check the beacon signature to ensure source is valid
Loop through NPD
        Each public id – NIST value = closer to zero than
previous stored value?
                If yes – public node id now potential leader
                If no – Discard public node id
        Loop until all NPD has been queried
```

Figure 3. Pseudo-code for the leader selection algorithm

The probability of two nodes having the same value once the calculation has been conducted is 1.4×10^{77}. While this is a small figure, should this occur, the node with the highest uptime would be selected.

Because each node has an equal probability of confirming the next block irrespectively of their resources, nodes are not disincentivized to pool together. Therefore our consensus algorithm provides more decentralization than Bitcoin.

4. CONSENSUS ENERGY AND RESOURCE CONSUMPTION - BITCOIN AND TEMTUM SIMULATIONS

This section will compare the resource consumption, including energy, of the proof-of-work algorithm deployed within Bitcoin compared to the consensus algorithm used in Temtum.

We first analyzed the energy consumption of the Bitcoin mining method.

We calculated the network hash rate by obtaining the network difficulty contained within the blockchain block headers. To decode the blockchain, we created a blockchain parser which decoded the blockchain into clear text.

We are using data from commercially available products such as Bitcoin miners and energy cost per kilowatt to determine Bitcoin's proof-of-work cost.

Due to the inability to locate miner's physical location, we will use a static value for the electricity cost due to a full node having one, more than one, or zero miners attached to them. Furthermore, since the miners themselves are unable to be queried, it is impossible to determine which model is being operated. Therefore we will use a static value for miners hash rate and energy consumption obtained from the AntMiner S9 ASIC miner.

Using the below formula, we calculated the hash rate using the difficulty we obtained from the block headers. (Where D is the difficulty):

D * 2**256 / (0xffff * 2**208)

Figure 4 shows the estimated terahashes per second on the network since the genesis.

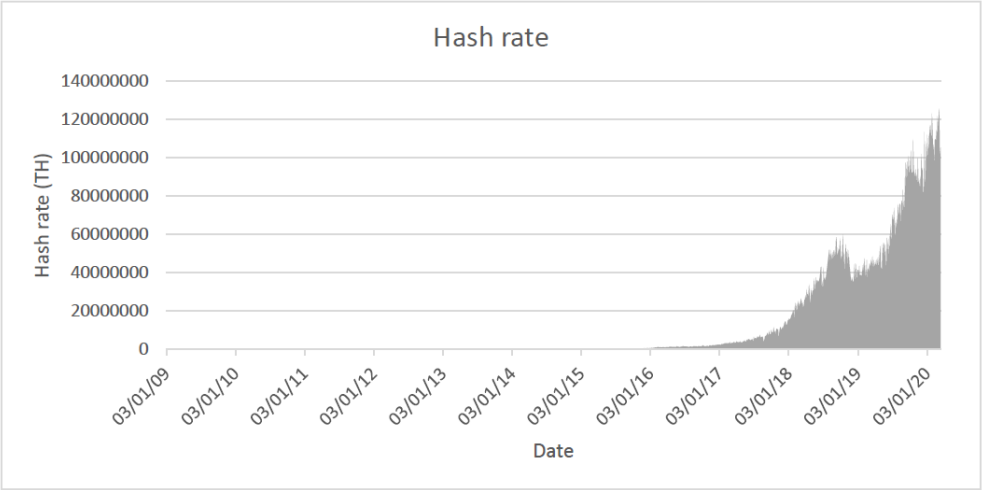

Figure 4. The calculated hash rate of the Bitcoin network since launch

We create models to compare and contrast Bitcoins' proof-of-work energy consumption compared to Temtum's consensus algorithm, which will be evaluated during this section.

The difficulty was observed to increase by 5% every 14 days between February 2016 and August 2017. A fixed cost of electricity of $0.15 per kWh will be utilized for the calculations. This was calculated as the average cost of a US person during the observed period.

Parameters of the experiment (Bitcoin)

Starting Difficulty:	150000000000 Gh/s
Growth (%) 14 days	5
Hash rate per miner (Gh/s)	13500
Power consumption (W)	1300
Cost per kWh ($)	0.15

Our simulations concluded that a single miner conducting the proof-of-work with the current network difficulty would take an average of 148.9 years to find a solution to the proof-of-work.

Expanding the simulation, the average Bitcoin miner consumes 11,388 kWh of electricity yearly, and excluding any profits from block rewards operates a loss of -$3,508.93 during the same period.

The same experiment was conducted on the Temtum network. It has been observed that the average Temtum node consumes 0.05 kWh of electricity. This is due to dedicated ASICs and high-resource computers not participating in Temtum; instead, a basic home laptop can be used.

It was simulated the average Temtum node uses 438 kWh of electricity over the same period, which results in a -$65.76 loss to the node operator.

We can conclude that the Temtum node block confirmation process is 53.4 times cheaper than the Bitcoin network in comparison.

Bitcoin's yearly energy consumption is equivalent to Tajikistan, the world's 84th most energy-consuming country in 2014, with the network consuming 4.56×10^{16} joules of energy. This

country has a population of 8,330,946, which shows how inefficient the consensus algorithm of Bitcoin is.

A Temtum network that operates with the same number of nodes as Bitcoin miners and nodes would consume 8.53×10^{14} joules over the same yearly period, making temtum equivalent to Gibraltar, a country with a population of 29,328.

We calculated Bitcoin to currently has a minimum of 1,100,000 ASICs miners operating at the network.

To enable an accurate comparison, we simulated a Temtum network that contains 1,100,000 nodes. This would result in a node being selected for a to confirm a block on average every 10.57 years

We can conclude that a Temtum node is picked on average 14.09 more often for confirming a block than a single Bitcoin miner. Furthermore

5. ATTACK COMPARISON – BITCOIN COMPARED TO TEMTUM

5.1. DNS Poison Attack

As was previously demonstrated, a node conducting the bootstrap process with Bitcoins DNS nodes would stage would receive an average of 28 nodes,which equates to a 0.38% view of the network

A simulated attack was conducted where an adversary could query the DNS nodes from nodes under the adversary's control. The DNS nodes within Bitcoin relay the most recently queried nodes to nodes, which are bootstrapping on the network. When a node queries the DNS node, they are likely to receive nodes under the adversary's control.

A bootstrapping node attempts to connect to each node received from the DNS. Once eight connections are established to nodes on the Bitcoin network, the download of the blockchain begins.

Due to blockchain properties, if a single honest node is collected to out of a pool of malicious nodes, the honest node will supply the victim node will correct data, even if there are significantly more malicious nodes connected to. This is due to the assumption that the attacker does not process a blockchain with a higher proof-of-work than the honest network.

A graphical representation in figure 5 demonstrates this problem. Where the yellow circle represents the node joining, blue the malicious nodes, and the green nodes represent the honest network.

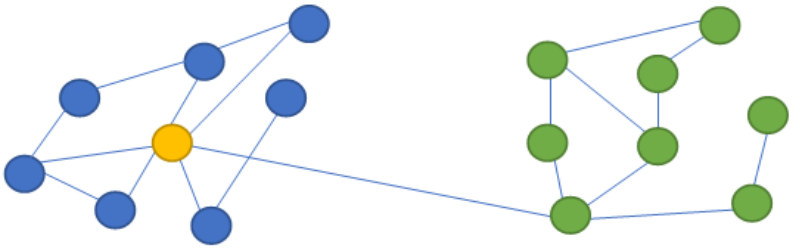

Figure 5. Bootstrap partition attack failure on Bitcoin

For an attacker with 100 US dollars to spend to conduct this attack, they would be able to host 30 malicious nodes on the Bitcoin network. With this small number of nodes, a simulated success rate of 92% in which a bootstrapping node would receive a response containing all nodes under the adversary's control.

When an attacker controls all the nodes the victim node is connected to, they can provide the victim node with a valid but potentially different from the honest network. This would enable double-spend attacks to be conducted against this node.

Due to the global view of the network provided to each bootstrapping node on the Temtum network throughout the NPD, the known list of nodes to the bootstrapping node rises from 28 in Bitcoin to the total network size in Temtum.

To enable a direct comparison, we simulated a 10,000 node Temtum network, of which a malicious adversary controlled 30 of them. Like bitcoin, the bootstrapping node would select eight nodes at random to connect to begin the bootstrapping process. Our simulations showed a 5.210508e-24 probability of this attack succeeding.

We formulated this to be:

Probability of success = ${}^{Y}C_{8} / {}^{x}C_{8.}$

Where y is the number of malicious nodes, and X is the total number of nodes on the network. 8 is the total nodes selected by the bootstrapping node.

This demonstratesthat the consensus algorithm's NPD element also significantly reduces the probability of success from a network partition attack. Furthermore, due to the random selection of nodes from the NPD, the adversary cannot exploit any vulnerability in the nodes' positioning to gain an advantage.

We can conclude that the NPD implementation as a component of the consensus algorithm deployed within the Temtum network provides greater resistance to network-level attacks during the bootstrapping phase. Furthermore, we have demonstrated that the Temtum network is more resistant to Sybil attacks than Bitcoin due to the NPD.

5.2. Sybil Rewrite Attack

We expanded the Sybil attack to demonstrate how an attacker with majority control could potentially rewrite historical data. We assume a malicious adversary has successfully partitioned a node away from the bootstrap's honest network during this section.

There are hardcoded block hashes inserted into the Bitcoin core source code. This is known as checkpoints. The use of checkpoints makes it impossible to alter data before this date due to the invalid hashes that would result from the blocks, even if the blocks are valid.

The last checkpoint to be implemented in the source code occurred at block 295000 and added to Bitcoin Core 0.9.3. This checkpoint was added on April 9, 2014, with a block difficult of 6,119,726,089.

An Antminer S9 uses 0.1 Joule per 10^9 hashes it computes. For a difficultly 6,119,726,089, t was modeled, the miner would need to complete $2.62 * 10^{19}$ hashes costing 73 USD in electricity. The attacker would be able to generate subsequent blocks the same rate until the difficulty adjusted.

This demonstrates the impact of the DNS poison attack on Bitcoin. When we conducted this simulation on the Temtum network, the attack failed.

The failure of this attack on the Temtum network was due to the consensus algorithm and the NPD.

The block validation process is more complicated than Bitcoin due to the block architecture of blocks stored within the temporal blockchain deployed on the Temtum network.

Each block header contains the NIST timestamp of when the block was published and signed by the node that confirmed it. The timestamp's inclusion within the block header demonstrates the block could not be computed ahead of time and had to be generated at that point in time or later. This prevents an attacker from making a longer chain and presenting this to the network as valid can be achieved on Bitcoin.

Furthermore, since the NPD history is made public and the NIST random beacon history, a node can randomly select a block to validate. By reversing the consensus algorithm, the node would compare the node results that should have been responsible for signing to the node, which did sign the block.

This further demonstrates the consensus algorithm deployed within the Temtum network as being more resistant to attack than the Bitcoin proof-of-work model.

The attack could succeed if the NPD publications could be rewritten; however, this assumes a majority of directory nodes being malicious and coordinating to alter previously agreed NPDs. This attack vector is considered unlikely and out of the scope of this section.

6. CONCLUSION

This paper proposed a novel consensus algorithm that was more energy-efficient while maintaining the Bitcoin proof-of-work algorithm's security properties.We outlined the algorithm's consensus algorithm and critical components, such as the NPD within this paper.

We demonstrated a series of attacks on the protocols to simulate a well-resourced adversary through live data collection and simulation.

During each simulation, we demonstrated a measurably way the consensus algorithm of Temtum is either more efficient or more secure when compared under the same constraints as the Bitcoin proof-of-work algorithm.

Therefore, we can conclude the current proof-of-work is inefficient and vulnerable to attacks, which can be easily solved with a solution like a consensus algorithm proposed here.

REFERENCES

[1] S. Nakamoto, "Bitcoin P2P e-cash paper," 2008 October 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[2] P. Cuccuru, "Beyond bitcoin: an early overview on smart contracts," International Journal of Law and Information Technology, Volume 25, Issue 3, p. 179–195, 2017.

[3] J. Bohr and M. Bashir, "Who Uses Bitcoin? An exploration of the Bitcoin community," in 2014 Twelfth Annual International Conference on Privacy, Security and Trust, 2014.

[4] A. Biryukov, D. Khovratovich and I. Pustogarov, "Deanonymisation of Clients in Bitcoin P2P Network," in CCS '14: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, 2014.

[5] J. Soria and V. Savolainen, "Too Big to Cheat: Mining Pools' Incentives to Double Spend in Blockchain Based Cryptocurrencies," in SSRN Electronic Journal, 2019.

[6] M. Romiti, A. Judmayer, A. Zamyatin and B. Haslhofer, "A Deep Dive into Bitcoin Mining Pools: An Empirical Analysis of Mining Shares," 2019.

[7] L. Lamport, R. Shostak and M. Pease, "The Byzantine generals problem.," Concurrency: the Works of Leslie Lamport., p. 203–226, 2019.

[8] Y. Chen and J.-S. Chou, "ID-Based Certificateless Electronic Cash on Smart Card against Identity Theft and Financial Card Fraud," in The International Conference on Digital Security and Forensics, 2014.

[9] A. Back, "Hashcash - Amortizable Publicly Auditable Cost-Functions," 2003.

[10] D. M. A. Cortez, A. M. Sison and R. P. Medina, "Cryptographic Randomness Test of the Modified Hashing Function of SHA256 to Address Length Extension Attack," 8th International Conference on Communications and Broadband Networking, pp. 24-28, 2020.

[11] D. Bradbury, "The problem with Bitcoin," Computer Fraud & Security, pp. 5-8, 2013.

[12] A. Lamiri, K. Gueraoui and G. Zeggwagh, "Bitcoin Difficulty, A Security Feature," Information Systems and Technologies to Support Learning, pp. 367-372, 2018.

[13] S. M. Werner, D. I. Ilie, I. Stewart and W. J. Knottenbelt, "Unstable Throughput: When the Difficulty Algorithm," 2020.

[14] E. Budish, "The Economic Limits of Bitcoin and the Blockchain," NBER Working Paper, 2018.

[15] B. Kaiser, M. Jurado and A. Ledger, "The Looming Threat of China: An Analysis of Chinese Influence on Bitcoin," 2018.

[16] A. Vries, "Bitcoin's Growing Energy Problem," Joule, pp. 801-805, 2018.

[17] Visa, "Annual report 2019," Visa, 2019.

[18] A. l. o. o. panelJonTruby, "Decarbonizing Bitcoin: Law and policy choices for reducing the energy consumption of Blockchain technologies and digital currencies," Energy Research & Social Science, pp. 399-410, 2018.

[19] C. Ye, G. Li, H. Cai, Y. Gu and A. Fukuda, "Analysis of Security in Blockchain: Case Study in 51%-Attack Detecting," in 2018 5th International Conference on Dependable Systems and Their Applications (DSA), 2018.

[20] N. Shi, "A new proof-of-work mechanism for bitcoin," 2016.

[21] H. Chena, T. N. Conga, W. Yang, C. Tan, Y. Li and Y. Ding, "Progress in electrical energy storage system: A critical review," Progress in Natural Science, pp. 291-312, 2009.