2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)

Reducing Forks in the Blockchain via Probabilistic Verification

Bing Liu
College of Computer Science and Technology
Harbin Institute of Technology
Shenzhen, China
liubing@stu.hit.edu.cn

Yang Qin
College of Computer Science and Technology
Harbin Institute of Technology
Shenzhen, China
csyqin@hit.edu.cn

Xiaowen Chu
Department of Computer Science
Hong Kong Baptist University
Hong Kong, China
chxw@comp.hkbu.edu.hk

Abstract— **Blockchain is a disruptive technique that finds many applications in FinTech, IoT, and token economy. Because of the asynchrony of network, the competition of mining, and the nondeterministic block propagation delay, forks in the blockchain occur frequently which not only waste a lot of computing resources but also result in potential security issues. This paper introduces PvScheme, a probabilistic verification scheme that can effectively reduce the block propagation delay and hence reduce the occurrence of blockchain forks. We further enhance the security of PvScheme to provide reliable block delivery. We also analyze the resistance of PvScheme to fake blocks and double spending attacks. The results of several comparative experiments show that our scheme can indeed reduce forks and improve the blockchain performance.**

Keywords- blockchain, fork, security, performance

I. Introduction

Blockchain is the underlying technology of Bitcoin. In the blockchain network, there are thousands of nodes, each of which holds all the blocks that are created by miners and broadcasted to all nodes in the network.

According to the standard protocol of blockchain [1], nodes need to verify the block before accepting it, including the verification of hash value of the block head and transactions contained in the block body. Then the block could be added to its chain and broadcasted to its neighbor nodes. Figure 1 visualizes the process of block propagation. Node M receives a block, verifies it and sends an *inv* message to its neighbors. Suppose node N receives the *inv* message. Since it does not know about the block, it replies a *getdata* message. Upon receiving the *getdata* message, Node M delivers the block to Node N.

As illustrated in Figure 1, at each node or hop in the course of broadcast, the message incurs a propagation delay. The propagation delay is the combination of transmission time and the time of local verification of the block. The transmission time includes the time it takes to announce, request and transmit the block [1].

Due to the delay in the generation and propagation of blocks and the asynchrony of network itself, it is common when two or more miners produce blocks with the same height at roughly the same time. This creates an apparent fork in the blockchain, as shown in Figure 2. A miner M_1 who has successfully solved the proof-of-work puzzle broadcasts $Block_{21}$ to the network. A miner M_2 is far from M_1 and, since he has not received $Block_{21}$ in a period of time, he digs out $Block_{22}$, which has the same height as $Block_{21}$. According to the blockchain protocol [2], when miners produce simultaneous blocks at the end of the blockchain, each node individually chooses which block to accept. And nodes usually use the first block they receive without other considerations. The main chain forks due to different nodes' acceptance of different blocks.

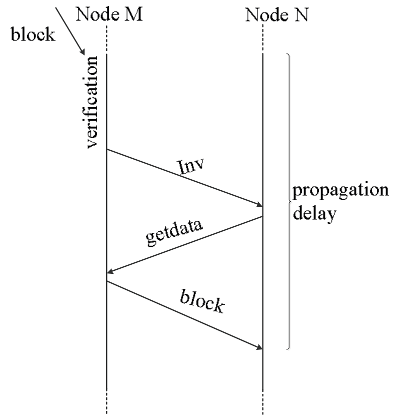

Figure 1. The process of block propagation [1]

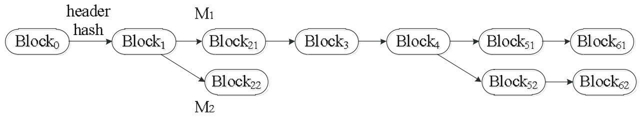

Figure 2. Forks in the blockchain [2]

Fork is a problem that must be solved. The emergence of forks not only wastes the computing power of the network, but also weakens the security of the blockchain.

This paper aims to study how to reduce the frequency of forks in the blockchain. The main contributions are summarized as follows:

- We propose a PvScheme of probabilistic verification for peers receiving blocks. It does not require to verify blocks one by one, so as to reduce the propagation delay and greatly reduce the blockchain forks.
- We enhance the security of PvScheme to provide reliable block delivery for peers. Two preventive measures are adopted against attacks.
- We conduct simulation experiments to verify the performance of PvScheme. Through several comparative tests, the results show that our scheme

2473-3490/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDEW.2019.00-42

indeed reduces forks and improves the blockchain performance.

II. Related Works

The performance and security of blockchain have been a hot topic in recent years. Christian et al. [1] discussed the relationship between the propagation delay and the probability of blockchain forks. They claimed that the blockchain forks were caused by the block propagation delay in the network. Eleftherios et al. [3] designed ByzCoin, a cryptocurrency based on a strong consensus protocol. They introduced a joint signature scheme to reduce the overhead of PBFT rounds and the overhead of lightweight client authentication transaction requests. Eyal and Sirer [4] proposed a selfish mining strategy in the study of Bitcoin. The strategy enables an attacker that owns only about 25% of the system's computing power to gain benefits beyond this ratio, resulting in a lot of computing works of honest nodes wasted. Sapirshtein et al. [5] further studied this attack strategy and pointed out that even if the computing power was less than 25% of the total network, attackers could still find the optimal selfish mining strategy to obtain disproportionate benefits. Another two papers [4][5] overturned the argument that Nakamoto [6] first put forward the 51% attack in 2008. Babaioff et al. [7] proposed that adversaries could publish their private block by controlling most nodes of the system, thereby weaken the proportion computing of other nodes in the network. It makes it easier for an attack's private block to gain recognition from other nodes, thus affecting the security of the blockchain. Heilman et al. [8] proposed that adversaries could control the connections of other nodes, and cut off network communication, thus making it easier to perform selfish mining. Rosenfeld [9] systematically introduced the conditions of double-spending implementation and the probability of success, and gave the concrete calculation of how to make double-spending profitable, and proved it with a strict mathematical reasoning. Ghassan et al. [10] proposed that when the blockchain forked, in the fast Bitcoin payment, there would be a very high probability to perform a successful double-spending attack. Tobias et al. [11] proposed a concept of concretization threshold for fast Bitcoin transactions, and as long as the verification process reaches this threshold, it is ensured that the transaction is validated and the good can be sold. Arthur et al. [12] mainly introduced how an attacker delayed the delivery of blocks and transactions to the victim in double-spending attacks, and proposed several solutions to effectively prevent these situations. Wang et al. [15] measured and analyzed the Bitcoin blockchain network from the mining pools' perspective. Some recent studies proposed verifiable and authenticated queries in blockchain [16] [17].

III. Blockchain Forks

A. Factors for Forks

There are many factors that affect the blockchain forks. Some forks are caused by the block propagation delay. Since blocks are found independently at random by the peers in the network, a block might be found while a conflicting block is being propagated in the network [1]. When selfish mining exists, a malicious miner keeps his mined blocks private, secretly creating a private branch, thus artificially causing the blockchain forks [13]. Malicious miners can also perform double-spending attacks in selfish mining. There are some other factors, such as the block generation rate set by the blockchain system [14], and the size of the propagation rate of network in the blockchain [13].

In the current Bitcoin's blockchain, the average time for a node to receive a block is 12.6 seconds, while its median is 6.5 seconds. And 5% of nodes still do not receive the block after the block has been broadcasted for 40 seconds, which brings the possibility of fork into Bitcoin. According to the current statistics, Bitcoin forks occur every day, with an average frequency about every 60 blocks [1].

Fork is inevitable due to the randomness of mining. In this paper, we consider how to reduce the possibility of forks. Here we mainly focus on the block propagation delay in the network.

B. Block Propagation Delay

The blockchain forks are mainly caused by the block propagation delay in the network [1]. Thus, if the size of blockchain network increases, propagation delays will grow and, consequently, the rate of blockchain forks will also increase.

Blockchain designers ensure the block security by validating blocks continuously. The same verification process is performed when different nodes accept the same block. Each node needs to validate each block and transaction, which may cause an over-verification problem in the blockchain. And it takes a lot of time resources and brings more possibilities to the fork of main chain.

To some degree, it can reduce forks if we can reduce block propagation delay in the blockchain. In addition to the transmission time, it still has a local verification time when nodes receive blocks. And the verification time is the major contributor to the propagation delay in the network [1]. The improvement of the verification time will greatly improve the efficiency for peers to receive blocks, which will reduce the blockchain forks.

IV. Probabilistic Verification

In this section, we propose a PvScheme by using a probabilistic verification scheme for peers when receiving blocks. In our probabilistic verification, a node does not need to verify every block it has received, but selectively verifies it based on a probability. When a node prepares to receive a block, it generates a random probability ε at first; and if the probability ε meets the validation degree υ, the block does not need to be verified and is received directly. Otherwise, the block needs to be verified.

A. PvScheme

In order to carry on our scheme, we give some definitions.

1) Validation degree υ : We define the concept of validation degree, that is, the average block verification ratio of nodes in the blockchain. E.g., when the validation degree is 0.5, a block is verified every two nodes on average across the network. When the validation degree is 1.0, the node needs to verify every block, which is the standard protocol implemented now.

In this scheme, we set α to represent whether a node has verified a block. If the block has been verified, the flag α is set to 1; otherwise the flag α is 0.

The probabilistic verification algorithm is shown below.

Algorithm 1 Probabilistic Verification Algorithm

```
Input: blocks $B_1$,...$B_i$..., $B_s$, nodes $N_1$, ...$N_j$..., $N_r$,
       validation degree $\upsilon$
Output: verification flag $\alpha$
While new block $B_i$ is broadcasted do
    While node $N_j$ prepare to receive $B_i$ do
        Generate a random probability $\varepsilon$, the average value
        of $\varepsilon$ generated by all nodes on this block is $\upsilon$,
        i.e.: $\frac{1}{r}\sum_{k=1}^{r}\varepsilon = \upsilon$
        If probability $\varepsilon$ is less than validation degree $\upsilon$
            Node $N_j$ verifies the block $B_i$, and accepts it.
            verification flag $\alpha_j$ sets to 1
        Else
            Node $N_j$ accepts block $B_i$ directly
            verification flag $\alpha_j$ sets to 1
    End while
End while
```

The process of probability verification is shown in Figure 3. Due to the random probability $\varepsilon_1 < \upsilon$, N_1 needs to verify B_{102}, and accepts it later. On the contrary, $\varepsilon_2 \geq \upsilon$, N_2 can accept B_{102} directly.

2) Synchronization time τ : it is the delay required for all nodes to fully accept blocks in the network. That is, the total time from the first block being dug out until the last block being accepted in the network.

Theoretically, with the increase of validation degree υ, more nodes need to validate the block when they receive it. This results in greater propagation delays, and the synchronization time τ will increase.

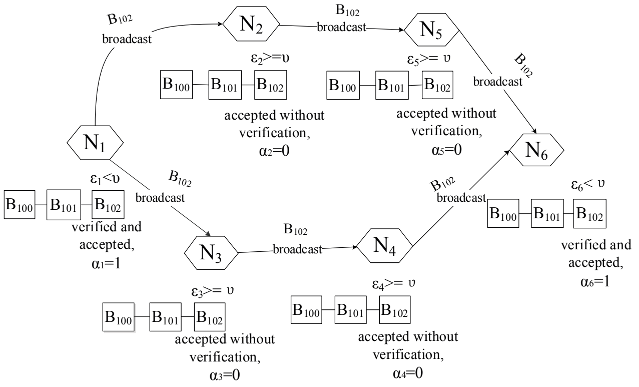
B_{102}
broadcast
N_2
B_{102}
broadcast
N_5
ε_2>=υ
B_{100}
B_{101}
B_{102}
accepted without verification, α_2=0
ε_5>= υ
B_{100}
B_{101}
B_{102}
accepted without verification, α_5=0
B_{102}
broadcast
N_1
N_6
ε_1<υ
B_{102}
broadcast
B_{100}
B_{101}
B_{102}
verified and accepted, α_1=1
B_{102}
broadcast
ε_6< υ
B_{100}
B_{101}
B_{102}
verified and accepted, α_6=1
N_3
B_{102}
broadcast
N_4
ε_3>= υ
B_{100}
B_{101}
B_{102}
accepted without verification, α_3=0
ε_4>= υ
B_{100}
B_{101}
B_{102}
accepted without verification, α_4=0

Figure 3. The process of probablity verification

3) Stale block rate r_s : According to the official definition of Bitcoin [2], once a fork is resolved, the blocks that do not contribute to the main chain are considered stale blocks. The stale block rate in the blockchain is the ratio of stale blocks to total blocks over a given period of time, which is expressed as (1):

$$r_s = \frac{stale\,blocks}{total\,blocks} \tag{1}$$

Theoretically, the greater the validation degree υ, the more nodes are needed to verify the block. Therefore, it costs longer block propagation delays, resulting in a larger stale block rate r_s.

4) Security index ς : We define the security index ς as the level of safety in which blocks are not tampered with at the current validation degree. When $\upsilon_0 = 1$, each block received by the node is verified, so its security index ς_0 should also be 1. Therefore, we define that the security index ς of the validation degree υ is proportional to the security index $\varsigma_0 = 1$ of the verification degree $\upsilon_0 = 1$.

$$\frac{\varsigma}{\varsigma_0} = \frac{\upsilon}{\upsilon_0} \tag{2}$$

Thus, we can get the security index ς of the validation degree υ:

$$\upsilon = \frac{\varsigma}{\varsigma_0}\upsilon_0 \tag{3}$$

It could be concluded that the value of security index is the same as the value of its validation degree. Therefore, greater validation degree υ leads to higher security index ς.

5) Security assessment rate ρ : We define a security assessment rate, which is used to represent a joint analysis of the block security and its performance at various validation degrees. We want to get a lower stale block rate as safely as possible. The security assessment rate is expressed as (4):

$$\rho = \tau\frac{r_s^2}{\varsigma} \tag{4}$$

where τ, r_s, ς are synchronization time, stale block rate and security index, respectively.

In practice, we need a smaller synchronization time τ and stale block rate r_s, as well as a larger security index ς. Therefore, we ask for a smaller ρ in this formula, which indicates that, with as much safety as possible, we get a relatively low fork, and compromise on security and performance.

B. Enhancing PvScheme Security

In our scheme, not all nodes verifies the block when they receive it. When a malicious attacker publishes a fake block, some nodes may accept the fake block directly and broadcast it to the network. This fake block is not discarded until a node verifies it and finds errors.

To ensure the correctness of blocks, we designed a feedback and rebroadcast mechanism. Moreover, in order to detect fake blocks as quickly as possible, we set a threshold θ, which represents a block can be continuously unverified by up to θ nodes.

1) Feedback and rebroadcast: When the validation degree is not 1.0, blocks are not strictly verified by nodes. There may be some fake blocks that have accepted by some nodes. A feedback and rebroadcast mechanism is designed, that is, when a node verifies a block and finds an error, it continues to feedback until it finds a node with the validated block, and requires the node to rebroadcast the block. The feedback design is shown in Figure 4.

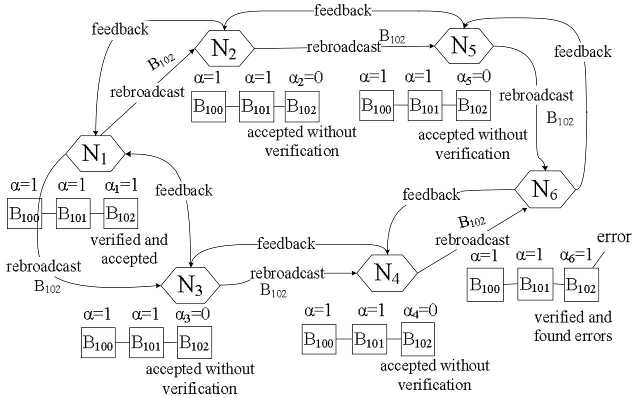

Figure 4. The process of feedback and rebroadcast

In Figure 4, when the node N_6 verifies B_{102} and finds an error, it discards the block and asks its previous nodes to rebroadcast B_{102}. N_4 or N_5 finds its own flag α_4=0 or α_5=0, and continues to feedback. The process continues until it finds α_1=1 on N_1, so the feedback process ends. N_1 should rebroadcast its block B_{102}.

The feedback and rebroadcast algorithm is shown below.

Algorithm 2 Feedback and rebroadcast Algorithm

```
Input: blocks $B_1$,…$B_i$…, $B_s$, nodes $N_1$, …$N_j$…, $N_r$,
Output: verification flag $\alpha$
While new block $B_i$ is broadcasted do
    While node $N_j$ prepare to receive $B_i$ do
        Node $N_j$ verifies the block $B_i$
        If $B_i$ is true
            $N_j$ accepts $B_i$, verification flag $\alpha_j$ sets to 1
        Else
            j=j-1, it feedbacks to previous nodes
            While verification flag $\alpha_{j-1}$ is not equal to 1 do
                j=j-1, it feedbacks to previous nodes again
            End while
            Node $N_j$ rebroadcasts $B_i$
    End while
End while
```

2) Threshold θ : We set the distance between two adjacent nodes to 1, and when nodes adopt probabilistic verification, the average path length of two verification nodes could be expressed as (5):

$$\chi = \frac{1}{\upsilon} \tag{5}$$

where χ is the average path length of two verification nodes at the validation degree υ.

In the worst case, all nodes will not verify a block. If the number of nodes in the network is *n*, the probability of its worst case occurring is in (6)

$$\varphi = \upsilon^n \tag{6}$$

where φ is the probability that all nodes in the network do not verify a block. When υ =0.5, *n*=10, φ is less than one thousandth. Since there are thousands of nodes in the real network, the possibility of the worst case is negligible.

To avoid the worst case, we set a threshold θ for a block, which represents a block can be continuously unverified by up to θ nodes. The value of θ is initialized to zero. If the block is not validated and accepted, the value is incremented by one, and if the block is verified, the value is reset to zero.

We set threshold value $\theta \le 3$, which means three consecutive nodes at most are allowed to accept the block without verification. When the next node receives the block, the block must be verified. In addition, if there is an error, the block can be retransmitted in a shorter path.

C. Defense against Attacks

1) Fake blocks attacks: Here, we set T_{new} to the propagation delay of PvScheme. p is the probability of a block error at the validation degree υ , and T_{sta} is the propagation delay in the standard protocol. We set transmission time t_m, verification time t_v, the feedback time t_f at a hop. The feedback process only takes a little time, so we get $t_v \gg t_m \gg t_f$.

According to our scheme, the transmission time is t_m at a hop, and the average verification time is υt_v at the validation degree υ . However, the block may be wrong, so the feedback time and rebroadcast time should be included. The probability of block error is p , the average feedback path length is $1/\upsilon$ at the validation degree υ, and the feedback time is t_f at a hop, so we get the average feedback time is $p\frac{1}{\upsilon}t_f$, and the average rebroadcast time is $p(\frac{1}{\upsilon}t_m + \upsilon t_v)$. The rebroadcast process may still be wrong, and it may need the second feedback and rebroadcast process, and the time is $p^2(\frac{1}{\upsilon}t_f + \frac{1}{\upsilon}t_m + \upsilon t_v)$, and the third feedback and rebroadcast process and so on. So we get T_{new}:

$$T_{new} = (t_m + \upsilon t_v) + (\frac{1}{\upsilon}t_f + \frac{1}{\upsilon}t_m + \upsilon t_v)\sum_{i=1}^{\infty} p^i \tag{7}$$

However, in the standard protocol, the propagation delay T_{sta} is the combination of the transmission time and verification time.

$$T_{sta} = t_m + t_v \tag{8}$$

We would like to know the range of p , which makes the propagation delay of our scheme is still less than the delay of the standard protocol. Make $T_{new} = T_{sta}$, and we get:

$$p \approx \frac{\upsilon(1-\upsilon)t_v}{\upsilon t_v + t_m + t_f} \tag{9}$$

When $\upsilon = 0.5$, $t_v \gg t_m \gg t_f$,we get $p \approx 0.5$, and $\upsilon = 0.9$, we get $p \approx 0.1$.

It can be concluded that our scheme costs less propagation delay than the standard protocol if less than 50% error blocks occurs in the validation degree of 0.5, or less than 10% error blocks occurs in the validation degree of 0.9. Since the block error probability of the blockchain is currently very small, our scheme is very practical.

2) double-spending attacks: Our scheme is more effective in preventing double-spending attacks, especially for the malicious double-spending in selfish mining proposed in [3]. In our scheme, at each node, it gets the propagation delay T^*_{new} if there is no need to feedback and rebroadcast.

$$T^*_{new} = t_m + \upsilon t_v \tag{10}$$

In the standard protocol, the propagation delay is in (8). We set η to the propagation delay improvement of PvScheme over the standard protocol, and we get:

$$\eta = \frac{T_{sta} - T^*_{new}}{T_{sta}} = \frac{(1-\upsilon)t_v}{t_m + t_v} \tag{11}$$

For the case $\upsilon \le 1$, $t_v \gg t_m$, we get $\eta \approx 1-\upsilon$. So a new block will be transmitted more quickly to the remaining nodes in PvScheme.

When a malicious attacker uses a fork to implement double-spending, our scheme could speed up the broadcast of the honest block, and make the honest block first learnt by most of the nodes in the network. And according to the protocol of the blockchain, miners usually work under the first block they receive [2]. It greatly reduces the likelihood that a malicious block will be accepted by most computing power of the network, which hinders the selfish mining and double-spending attacks.

V. EXPERIMENTS AND RESULTS

In order to consider the tradeoff between security and performance in the blockchain, we set the validation degree υ from 0.5 to 1.0 in our experiments, as shown in TABLE I.

TABLE I. VALIDATION DEGREE AND ITS CORRESPONDING PARAMETERS

validation degree υ	average number of verifications per 10 nodes	average feedback path length χ
0.5	5	2
0.6	6	10/6
0.7	7	10/7
0.8	8	10/8
0.9	9	10/9
1.0	10	1

This experiment draws on a simulation platform that has gained a lot of affirmations [3]. We have also modified the platform to increase the related requirements and parameters for our assessment. Then, we performed 5 simulations with validation degree υ from 0.5 to 1.0, and calculated the average value as our reference data. In this experiment, 1000 blocks were designed to be broadcast between 6000 nodes. And 16 miners were designed in this experiment. The experimental results are shown in the chart below.

Figure 5 shows the synchronization time at different validation degrees. Figure 6 shows the stale block rate at different validation degrees.

In Figure 5 and Figure 6, as validation degree increases, so does the synchronization time and the stale block rate, which is consistent with our previous theoretical analysis.

In contrast, it can be concluded that after enhancing PvScheme security, our scheme has higher synchronization time and stale block rate than before. The security design increases the delay of the system, and more forks appear in the blockchain, which is consistent with theoretical analysis.

Figure 7 shows the propagation delay of blocks received by different proportions of nodes in the network at different validation degrees.

As can be seen from Figure 7, the block propagation delay increases with the increase of the validation degree. It is still in accordance with our analysis. More propagation delay occurs because more nodes verify the block.

We also have captured the longest forks at various validation degree, which is shown in Figure 8. As shown in Figure 8, as the validation degree increases, in addition to the validation degree of 0.9, the longest forks also show an increasing trend. The fork generation is inherently random, so the data of validation degree of 0.9 is normal. It means that PvScheme can indeed reduce the number of continuous forks, thus saving computing power for the network.

Here, we show the experimental results under various validation degrees in TABLE II.

TABLE II. EXPERIMENTAL TEST RESULTS

Validation degree υ	Synchronization time τ	Stale block rate r_s	Security index ς	Security assessment rate ρ
0.5	37543.3	1.64%	0.5	20.195
0.6	38008.6	1.75%	0.6	19.400
0.7	38473.7	1.78%	0.7	17.414
0.8	38700.8	1.80%	0.8	15.673
0.9	39080.1	1.88%	0.9	15.347
1	39878.6	1.98%	1	15.634

Based on the above theoretical analysis, we need to find the minimum security assessment rate ρ . In TABLE II, it meets the requirements when the validation degree υ=0.9. In other words, the validation degree of 0.9 can be used to improve the standard protocol. On the one hand, it can reduce the blockchain forks, and on the other hand, it can ensure the security of the blockchain.

VI. CONCLUSION

In this paper, we analyze the factors that cause the blockchain forks during block propagation. We propose a

PvScheme with probabilistic verification when nodes receive blocks. It does not require each node to verify new blocks, so it can greatly reduce the block propagation delay, and reduce the probability of forks in theory. We further enhance the PvScheme security, and design two mechanisms to guarantee the security of the blockchain. Our simulation results show that we get the best effect when the validation degree is 0.9, where it gets a relatively lower forks and higher security. In practice, the best validation degree can be determined by considering various environments and network structures.

ACKNOWLEDGMENT

This work was supported by the Science and Technology Fundament Research Fund of Shenzhen under grant JCYJ20160318095218091, JCYJ20170307151807788.

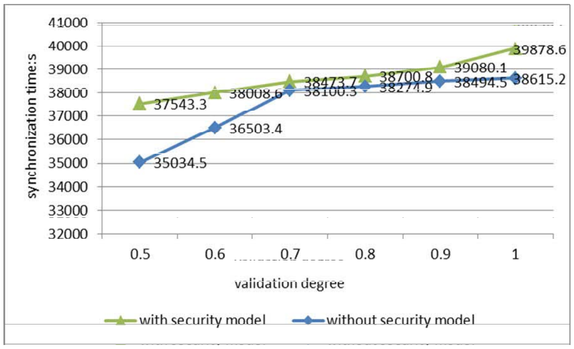

Figure 5. Synchronization time under various validation degrees

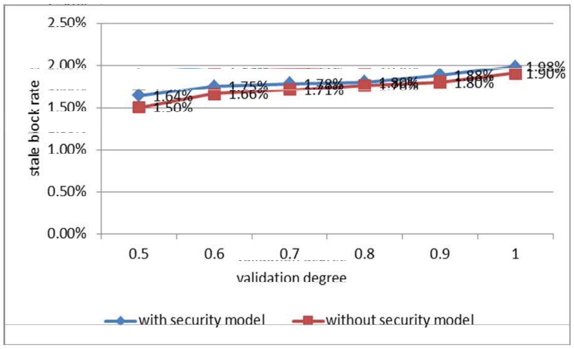

Figure 6. Stale block rate under various validation degrees

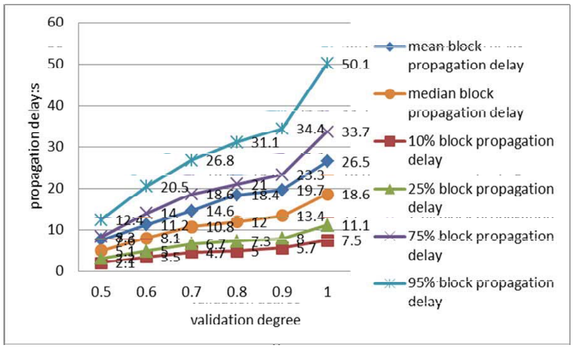

Figure 7. Block propagation delay under various validation degrees

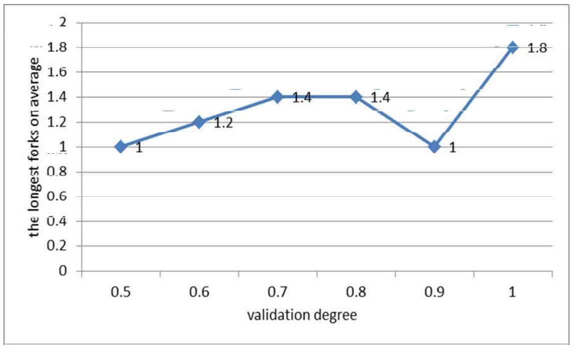

Figure 8. The longest forks under various validation degrees

REFERENCES

[1] C. Decker and R. Wattenhofer, "Information Propagation in the Bitcoin Network," in Proc. of IEEE P2P, 2013.

[2] https://bitcoin.org/en/developer-guide.

[3] E. K. Kogias, et al., "Enhancing bitcoin security and performance with strong consistency via collective signing," in Proc. of USENIX Security, 2016.

[4] E. Ittay and E. G. Sirer, "Majority is not Enough: Bitcoin Mining is Vulnerable," in Proc. of Financial Cryptography and Data Security, 2014.

[5] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in Bitcoin," in Proc. of the 2016 Conference on Financial Crypto (FC), 2016.

[6] Satoshi Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf.

[7] M. Babaioff, S. Dobzinski, S. Oren, and A. Zohar, "On bitcoin and red balloons," In Proc. of ACM EC, 2012.

[8] E. Heilman, A. Kendler, A. Zohar, et al., "Eclipse Attacks on Bitcoin's Peer-to-Peer Network," in Proc. of USENIX Security Symposium, 2015:129-14

[9] M. Rosenfeld, "Analysis of Hashrate-Based Double Spending," arXiv preprint, arXiv:1402.2009, 2014.

[10] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in Proc. of ACM CCS, 2012.

[11] T. Bamert, C. Decker. L. Elsen, S. Welten, and R. Wattenhofer, "Have a snack, pay with bitcoin," in Proc. of IEEE P2P, Trento, Italy, 2013.

[12] A Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in Proc. ACM CCS, 2015.

[13] A. Gervais, et al., "On the Security and Performance of Proof of Work Blockchains," in Proc. of ACM CCS 2016.

[14] F. Tschorsch and B. Scheuermann, "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2084-2123, 2016.

[15] C. Wang, X.-W. Chu, and Y. Qin, "Measurement and Analysis of the Bitcoin Networks: A View from Mining Pools," arXiv preprint, arXiv:1902.07549, 2019.

[16] C. Xu, C. Zhang, and J. Xu, "vChain: Enabling Verifiable Boolean Range Queries over Blockchain Databases," in Proc. of SIGMOD, 2019.

[17] C. Zhang, C. Xu, J. Xu, Y. Tang, and B. Choi, "GEM^2-Tree: A Gas-Efficient Structure for Authenticated Range Queries in Blockchain," in Proc. of ICDE, 2019.