Preventing Sybil Attack in Blockchain using Distributed Behavior Monitoring of Miners

Swathi P
Dept. of CSE
NIT Goa, India
swathip2693@gmail.com

Chirag Modi
Dept. of CSE
NIT Goa, India
cnmodi@nitgoa.ac.in

Dhiren Patel
VJTI Mumbai
Maharashtra, India
dhiren29p@gmail.com

***Abstract*— Blockchain technology is useful with the record keeping of digital transactions, IoT, supply chain management etc. However, we have observed that the traditional attacks are possible on blockchain due to lack of robust identity management. We found that Sybil attack can cause severe impact in public/permissionless blockchain, in which an attacker can subvert the blockchain by creating a large number of pseudonymous identities (i.e. Fake user accounts) and push legitimate entities in the minority. Such virtual nodes can act like genuine nodes to create disproportionately large influence on the network. This may lead to several other attacks like DoS, DDoS etc. In this paper, a Sybil attack is demonstrated on a blockchain test bed with its impact on the throughput of the system. We propose a solution directive, in which each node monitors the behavior of other nodes and checks for the nodes which are forwarding the blocks of only particular user. Such nodes are quickly identified, blacklisted and notified to other nodes, and thus the Sybil attack can be restricted. We analyze experimental results of the proposed solution.**

***Keywords*— Blockchain, Digital transaction, Security, Sybil attack, Behavior monitoring**

I. INTRODUCTION

In today's world, transaction systems need to be decentralized, transparent and incorruptible. Digital currency allows for instantaneous transactions and border-less transfer of ownerships. For example, bitcoin, a crypto-currency [1] distributes the business of creating money around the Internet. It uses computer algorithms to ensure that payment makes its way securely from buyer to seller. Bitcoin uses blockchain as an underlying technology to offer transparency on transactions with distributed verification. Here, a network of computers uses bitcoin blockchain which maintains the collective public database [2]. Whenever Bitcoin is added into blockchain, the transaction is locked for all information about it. As shown in Fig. 1, consider a user who initiates a transaction about sending money to the receiver. This transaction request is broadcasted in the peer to peer network of computers (nodes). These nodes verify and validate the transactions and user's status by using known algorithms. Here, the user's identity is not revealed instead his/her wallet address is considered to achieve anonymity (pseudo-anonymity). A digital wallet is created on the user's node. Each wallet has a unique address which is considered as an effective identity of an entity in the network. A verified transaction may involve crypto-currency, contracts/ agreements, records or any other data. The verified transaction is then combined with the other validated transactions and a new block of data is added in a ledger by satisfying the pre-requisites of difficulty (e.g. Proof of work) of the block. This block is then added into the existing blockchain in such a way that it can become permanent and immutable. Finally, the transaction is accepted by the community and the receiver gets crypto-currency in his/her digital wallet. Here, If anyone tries to corrupt the transaction, then that transaction is refused by node to continue in the blockchain.

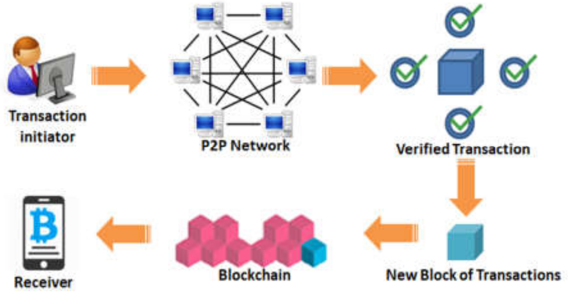

Fig. 1. Workflow of bitcoin blockchain.

As shown in Fig. 2, a blockchain consists of multiple blocks in a ledger, where each block has attributes viz; hash, timestamp, other information and main data. When a user executes a transaction, it is hashed and then broadcasted to each node in a network. Each node's block can contain many transaction records. Blockchain uses Merkle tree structure to generate a final hash value (Merkle tree root) which is recorded in the block header (hash of current block). Timestamp presents the time of generation of block. Other information may include signature of block, Nonce, or any other data as defined by the user as per requirement. Main data include transaction records.

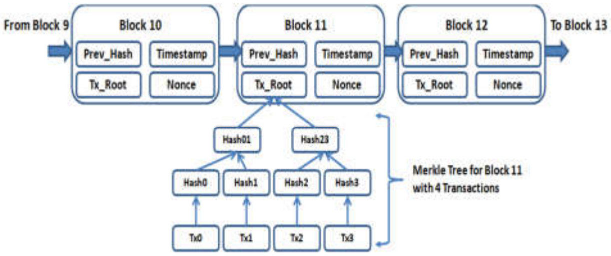

Fig. 2. Blockchain structure.

Blockchain offers many significant advantages [3]: It is fully decentralized; as it is eliminating the dependency on the centralized system for transaction verification, validation and related data updates. Using a consensus mechanism, it offers autonomy as each node in the blockchain network to perform transaction safely. In addition, data updated on blockchain are immutable. It offers anonymity of data as transactions performed among users require only wallet address. Consider a scenario, a company issuing health care allowance, and it wants to ensure that the issued money is used only for a medical purpose. An advantage of using blockchain here is that the issued money can be returned automatically to the issuer if it is not being used for a certain period of time for the specified purpose. Since blockchain is a decentralized system, peer-to-peer transaction can be done smoothly and in a transparent manner.

However, there are some security concerns which may hinder the wide adoption of blockchain. A Sybil attack [4] in blockchain can cause severe impact and can lead to several other attacks like DoS, DDoS, majority attack, mining pool attack etc [5][6]. In addition, existing proposals like CoinShuffle++/DiceMix [7], CoinShuffle [8], ValueShuffle [9], Dandelion [10] and SecureCoin [11] are proven to be vulnerable to Sybil and DoS attacks. Here, an attacker tries to fool the blockchain network by generating and controlling multiple identities which are considered as genuine in blockchain network. Whenever other nodes attempt to connect with the network, they are likely to see controlled majority fork of forged block connected to an attacker node. In addition, C. Decker [12] have analyzed that the dependency on blocks creates the delays in clearing the transactions and also causes the network threats. Thus, an attacker can control and manage whole network and as a result, genuine miners are no longer able to process blockchain transactions appropriately. This can decrease the throughput of the system drastically.

To prevent the effect of Sybil attack in blockchain, we propose a solution, in which each participating node monitors the behavior of other nodes and checks whether any node is forwarding blocks of only particular user over a period of time. Such behavior is considered as malicious and there is a possibility of Sybil attack which can affect the throughput of the system as genuine user's blocks cannot be verified. Such nodes are blacklisted and notified to other nodes to prevent the propagation of Sybil node's transaction block. We demonstrate the impact of Sybil attack on the blockchain using experimental setup and derive the throughput results of the proposed solution.

Rest of this paper is organized as follows: Section II discusses the Sybil attack in blockchain and related solutions existing for other networks. Section III presents the proposed solution directive to prevent Sybil attack in the blockchain, while section IV discusses the performance results of the proposed solution. Section V concludes our research work with references at the end.

II. Literature Review

A. *Sybil Attack in Blockchain*

In blockchain network, an adversary can create multiple virtual identities to take control over the whole network. Such nodes with virtual identities are called as Sybil nodes. An attacker can disconnect the genuine nodes from the blockchain network. In blockchain, miners join the network pool (mining pool) to get shared mining rewards. In addition, an attacker can create a large amount of IDs in a network. A malicious pool operator can also perform Sybil attack. As shown in Fig. 3, an attacker adds a significant number of zero power miners in the blockchain network. Such virtual miners participate in data dissemination, but cannot mine new block [6]. These virtual nodes forward the block of only attacker in the network and stops propagation of the genuine user's blocks. As a result, only an attacker's block propagates in the network and added into the blockchain. Thus, an attacker may get more rewards and reduces the overall throughput of the system.

Consider an example: Let *g* is the block generated by a genuine miner and *a* is the block generated by an attacker. Both broadcast their blocks in the blockchain network. Here, each miner broadcasts the received block to its neighbor nodes, and subsequently neighbor nodes broadcast to their neighbor nodes and so on. At the end, both the blocks should reach to all the nodes in a network. However, an attacker's virtual nodes are forwarding only an attacker's block *a* and prevents the propagation of block *g*. Although, random peer to peer network structure eventually propagates block *g*, but it is very slower than the propagation of block *a* due to a large number of malicious virtual nodes. As a result, an attacker is rewarded frequently than the genuine miners. In addition, genuine miner's computing power is wasted in processing block of *g*. It is also to note that majority bar is pushed high due to attacker's behavior; which in turn leads to further wastage of computing power. The impact of such attack is very high in the permissionless/public blockchain and it is difficult to detect Sybil nodes. However, such attacks can be easily detected in permissioned/private blockchain.

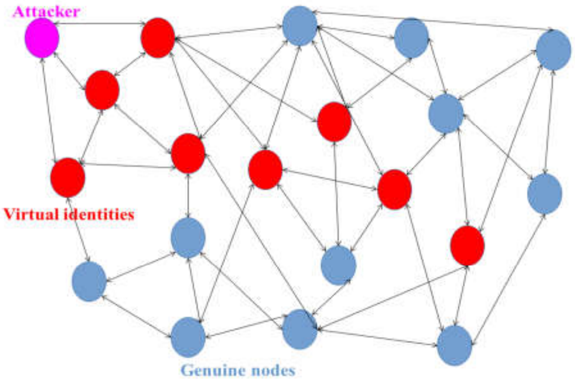

Fig. 3. Sybil attack scenario in blockchain.

B. *Related Work*

From the literature, we have observed that an attacker gets connected to a network pool having enough computation power to perform Sybil attack. Although it requires high computational power for an effective Sybil attack, it is worth to provide solution directive to prevent such vulnerability in blockchain.

We have surveyed Sybil attack detection methods for traditional networks. In radio resource testing method [13], a node verifies its neighbors by assigning each neighbor in a channel. It can then choose a channel randomly for listening. If the neighbor is legitimate, it should hear the message. If there are not enough channels to assign, a node can only test some subset of neighbors at a time. The key assumptions of this approach is that any physical device has only one radio which is not capable of transmitting and receiving messages on more than one channel at a time. Resource tests have been recommended for the same. Similarly, in blockchain context, behavior monitoring for each node is required in a distributed manner to reduce the security risk.

Registration based method [14] performs identity registration to prevent Sybil attack. Here, any node can check the list of "known-good" identities to validate another node. However, a list of known identities are required to be protected from any malicious modification. If an attacker adds malicious identities in such list, he/she can add Sybil nodes as genuine nodes.

Position verification method [15] considers static network. The network verifies the physical position of each node. Here, Sybil node can appear exactly at the same position of malicious nodes. By putting the limit on density of network, in-region based verification can tightly bind the number of Sybil identities that a malicious node can create and thus limit the attack.

Xu *et al.* [16] have proposed an algorithm to resist Sybil attack using network clustering. Here, honest peers are prevented from communicating with Sybil peers by separating them into two clusters connected by attack edges. The algorithm explicitly detects the attack edges, then prohibits communication over two stages viz; aggregation and re-aggregation. In aggregation, the weight for each edge is calculated and higher weighted edge is considered as suspect edge.

For vehicular ad-hoc network, Triki *et al.* [17] have used radio frequency identification device (RFID) to prevent Sybil attack. Here, RFID with the vehicle identification number is embedded with the vehicle to authenticate itself to the RSU (road side unit) and obtain short lifetime certificates.

G. Bissias *et al.* [18] have proposed Sybil resistant mixing named as Xim for bitcoin. They have used two party mixing approach. Consider if Alice wants to mix a coin stored in address *A*. Initially, Alice commits a transaction to advertise her willingness to mix with a partner tipping *t/2* coin from *A* to the miners. She chooses one who tips *t* coin to the miner, and thus no one else can recognize their partnership. Later, Alice confirms the partnership by releasing another *t/2* coin to the miners. Let the partner is Bob with address *B*. Here, Alice and Bob can swap funds. So, *d* coins from *A* transferred to *B* can be controlled by Bob and *d* coins from *B* to *A* can be controlled by Alice. Then, Alice can find new partner for further transaction. It requires higher mixing time.

In neighborhood similarity method [19], Sybil attack is detected by monitoring the hitting events of suspect node over the period of time. Here, each node observes the hitting event in neighbor list and repeats the process to calculate the eigen gap. If the hitting event is greater than the threshold then it is said to be Sybil. In case of blockchain, the position of the node is dynamic.

MixCoin approach [20] applies third-party mixing protocol for the anonymous transactions of Bitcoins. Here, a user applies some coins with a third-party and gets the same amount of coins from the mix which is applied by other users. It uses a reputation-based cryptographic technique to prevent coin theft. However, the user coins can be stolen by mix as mix knows the linking between the users and outputs.

BlindCoin [21] extends MixCoin techniques by applying blind signatures to generate user inputs and outputs which are cryptographically blinded. To remove linking of user input and output, it requires publishing extra transactions and receiving the blindCoins. It has additional costs and delays on mixing the coins.

TumbleBit [22] offers unidirectional unlinkable transaction hub which allows to perform fast and off-blockchain transactions using Tumbler, an untrusted party. In this approach, anyone, including Tumbler cannot link a transaction with sender and receiver.

C. Research Scope

As per our observation, very limited work has been done towards detecting Sybil attack in blockchain. The existing approaches suffer from high computational cost which makes them infeasible to use in highly dynamic and distributed blockchain network. There is a need of monitoring and analyzing the behavior of each node in a distributed manner and blacklisting the suspected nodes publicly. P. Winter [23], have discussed different Sybil attacks in Tor network and as per their analysis, it is found that manual verification is required for analyzing the Sybil frequently. In addition, there is a need of improving the throughput of the system in the presence of Sybil attack and achieving high accuracy and low error rate in detecting the Sybil attack in blockchain. As Sybil attack can occur due to forging of identity of genuine users, it is required to monitor the behavior of each node to reduce the impact of Sybil attack in working network instead of detecting the forged identity. In this paper, the proposed solution attempt to monitor the behavior of each node in a distributed manner and announce the suspect nodes with the improved throughput and accuracy.

III. Proposed Solution

The objective of the proposed work is to prevent the Sybil attack in permissionless and public blockchain, while improving the throughput on mining of genuine node's block. In addition, it should achieve high accuracy and low error rate while detecting Sybil nodes in the blockchain network.

We propose to include wallet generated address in each block of user. Each node in blockchain network has a unique address which does not reveal the identity of the user. This address is wallet generated. The processing steps for generating a unique address for each user is given in Fig. 4.

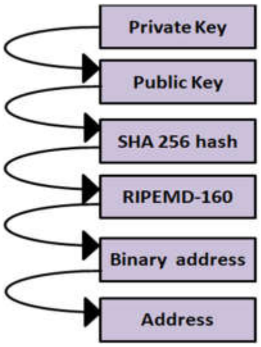

Fig. 4. Steps of digital wallet address generation.

For generating a wallet address, a user has to produce his identity like email address etc to the wallet. Here, wallet generates a pseudo-random number and converts it into the wallet import format. It takes initial affix 0x80 of the random number and applies a SHA256 algorithm on it. Then it takes first 4 bytes from the result of SHA256 algorithm and adds it at the end of the initial random number. The resultant output is converted into base28 string which is considered as a wallet import format private key [24]. To produce public key, an elliptic curve cryptography [25] is used. Consider, *k* is a private key and *G* is a generator point on elliptic curve, then *K*= *k.G* is the user's public key. This public key is hashed using the SHA256 algorithm and the resultant output is again hashed using RIPEMD-160. The final output is converted into binary value which is considered as a user's unique address.

In the proposed solution, we are extending the existing block structure in the blockchain with the user's unique address, as shown in Fig. 5. When a miner node creates a block for broadcasting in a blockchain network, his/her address is added in the block header. Accordingly, every miners' blocks contain their unique address in the generated blocks and broadcasted in the peer to peer network. Each of the receiving nodes monitors the behavior of sending/forwarding nodes by maintaining a monitoring table as shown in Table I.

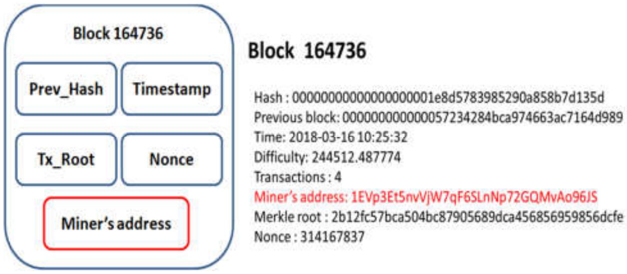
Block 164736
Prev_Hash
Timestamp
Tx_Root
Nonce
Miner's address
Block 164736
Hash : 00000000000000000001e8d5783985290a858b7d135d
Previous block: 0000000000000057234284bca974663ac7164d989
Time: 2018-03-16 10:25:32
Difficulty: 244512.487774
Transactions : 4
Miner's address: 1EVp3Et5nvVjW7qF6SLnNp72GQMvAo96JS
Merkle root : 2b12fc57bca504bc87905689dca456856959856dcfe
Nonce : 314167837

Fig. 5. An example of a block in the proposed solution directive.

TABLE I. NODE BAHAVIOUR MONITORING TABLE AT EACH NODE

Physical address of the forwarding node	Miner's address in the block header	Block number	Count
00-10-5A-44-12-B5	164at26WgqVgkSveET6nPzHhC8sysHJDTp	Block #510387, Block #510385	2
00-50-56-09-00-01	1EVp3Et5nvVjW7qF6SLnNp72GQMvAo96JS	Block #510386	1

The monitoring table (at each node) includes the receiving block's attributes such as block-number, address of a miner (who has generated the block), the physical address of the node (from where the block is received) and the count. The block number and address of miner are collected from the block header, while the physical address of the forwarding node is collected by analyzing the network packets. This address is permanent address and cannot be changed by any miner. The count represents the number of times a forwarding node has sent different blocks for a particular address.

In case of Sybil attack, the receiving node receives higher number of blocks (with an attacker's address) from Sybil nodes and very few blocks of genuine user are received. This happens due to the fact that Sybil nodes forward only an attacker's block and drop genuine user's blocks. As a result, genuine nodes innocently forwards only attacker's block in whole blockchain network. Thus, the propagation of attacker's block can be faster than the genuine block. In this case, monitoring table at each node can have higher counts for Sybil nodes and an attacker's address in the block header. Over the period of time, every node can observe this effect by looking into their monitoring table.

To prevent such attacks, if count at any node's monitoring table crosses some threshold *T*, that node puts the corresponding physical address in publicaly suspected list, as shown in Table II. This list represents that the given physical address is suspected as a Sybil node at number of nodes. This list is distributed among all the nodes in the blockchain network. Here, once a miner adds any physical address in this list, he cannot add the same address again. Once the suspected miners perform fair forwarding of blocks from genuine user, its effect will be reflected in the monitoring table of each node. If the count reduces below the threshold T, such suspected miner's count is reduced in the public suspect list. However, each node can decrease count only once for a given suspected address and also that node should have an increased count for same address once previously.

TABLE II. PUBLIC SUSPECT LIST OF SYBIL NODES

S. No	Suspected physical address	No of nodes have suspected
1	00-10-5A-44-12-B5	6
2	00-17-57-44-12-B6	7
3	00-51-56-0A-00-01	8
4	00-11-3A-42-10-A5	9

Over the period of time, each node can have a complete monitoring table with enough number of entries of Sybil nodes and a public suspected list. Now, if any node receives a block, it checks the miner's address in block header and physical address in public available list to take decision about forwarding that block to other nodes. If a physical address for the block with higher count is found in pubic list, receiving node considers that such block is coming from Sybil nodes and drops such block. Here, the suspected miner can escape from the public list by performing a fair forwarding of blocks of the genuine users. Thus, even if the genuine miner is added in the public list, he can escape from the public list by doing further fair forwarding.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

For the performance evaluation and validation of the proposed solution, we have created a private blockchain network at NIT Goa using Ethereumjs/testrpc [26][27][28][29] installed in 64-bit Ubuntu operating system. Ethereumjs/testrpc can be used for testing and development. As discussed earlier, block structure is updated by including the miner's address. As shown in Fig. 6, a Sybil attack is simulated on Ethereumjs/testrpc with an attacker node, different number of virtual identities and genuine nodes.

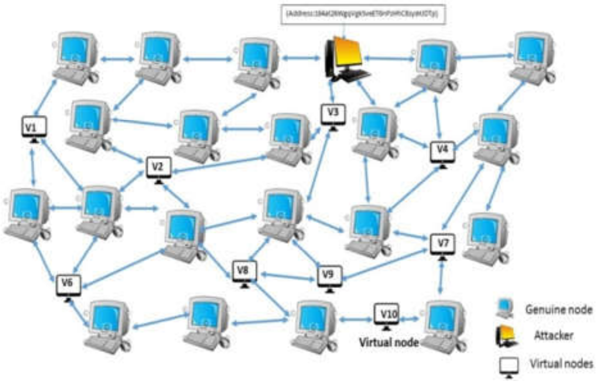

Fig. 6. Experimental setup of blockchain with Sybil nodes and genuine nodes.

Fig. 7 shows the throughput of the blockchain network before and after applying the proposed solution. We have considered the throughput in terms of number of blocks added into the blockchain by the genuine nodes with respect to number of blocks created by genuine nodes. With the help of the proposed solution, the throughput of the blockchain is increased even if Sybil nodes are present. Before applying the proposed solution, the throughput is decreased with the increasing number of Sybil nodes. Initially, one Sybil node's behavior is considered as fair, and thus throughput is decreased. This happens due to the count value not reaching to pre-defined threshold. However, after some time, throughput is increased with the increasing number of Sybil nodes as the genuine nodes have blocked the Sybil nodes to forward an attacker's block.

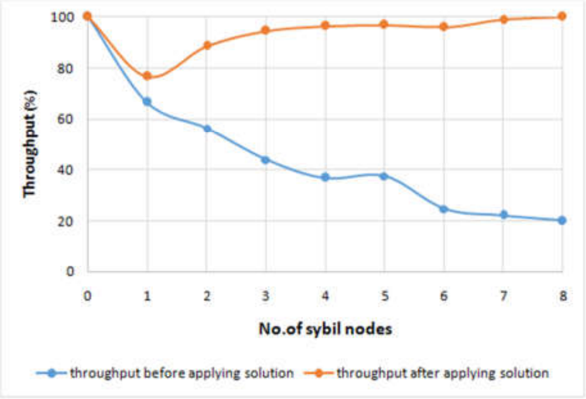

Fig. 7. Throughput results of the proposed solution in blockchain.

For testing the Sybil node detection capability of the proposed solution, different experiments with the increased number of Sybil and genuine nodes are performed, as shown in Table III.

TABLE III. NUMBER OF SYBIL NODES AND GENUINE NODES CONSIDERED

Sr. no.	Number of genuine nodes	Number of Sybil nodes
1	20	8
2	22	10
3	27	14
4	32	18
5	36	21
6	40	23

The performance results of the proposed solution are given in Table IV.

TABLE IV. PERFORMANCE RESULTS OF THE PROPOSED SOLUTION DIRECTIVE

Sr. no.	TPR (%)	FPR (%)	TNR (%)	FNR (%)	Accuracy (%)
1	62.5	0	100	37.5	81.45
2	70	10	90	30	80.4
3	78	14.9	85.1	21.4	81.83
4	83.3	15.7	84.3	16.6	83.88
5	90.4	19.45	80.55	9.5	85.53
6	95.6	20	80	4.3	87.84

True positive rate (TPR) indicates that the percentage of Sybil nodes are detected as Sybil nodes by the proposed solution. It is derived using (1).

$$TPR = {S\_D}/{S} \tag{1}$$

Where, S_D is the number of Sybil nodes detected by the proposed solution and S is the number of Sybil nodes present in the network. The proposed solution has more than 95% detection capability which can be increased with the increasing number of nodes. Therefore, false negative rate is decreased to 4.6%. False negative rate indicates the percentage of Sybil nodes are considered as genuine nodes by the proposed solution, as given by (2).

$$FNR = {G\_C}/{S} \tag{2}$$

Where, G_C is the number of nodes considered as genuine nodes.

True negative rate (TNR) indicates that the percentage of genuine nodes are considered as genuine nodes. It is derived using (3).

$$TNR = {G\_C}/{G} \tag{3}$$

Where, G is the number of genuine nodes present in the blockchain network. It is decreasing with the increasing number of Sybil nodes since genuine nodes are innocently forwarding the attacker's blocks, and thus the false positive rate (FPR) is increasing as it represents that the percentage of genuine nodes are considered as Sybil nodes. FPR indicates the percentage of genuine nodes are detected as Sybil nodes and it is given by (4).

$$FPR = {S\ \_D}/{G} \tag{4}$$

The proposed solution achieves higher accuracy (87.84%) with the increasing number of Sybil nodes. The accuracy of the proposed solution is derived using (5).

$$Accuracy = {S\_D + G\_C}/{N} \tag{5}$$

Where, N is the number of nodes in the blockchain network. However, genuine nodes are forwarding the blocks of other genuine nodes frequently, and thus they are considered as Sybil nodes (as false positives). This effect can be reduced in large network, where the genuine nodes forward the high number of genuine nodes' blocks, while Sybil node forwards only attacker's blocks. In this case, false positives will be very low. Above experiments are conducted to test the feasibility and performance validation of the proposed solution. In the experiments, we have used random threshold to count in the monitoring table at each node, threshold play a major role to consider Sybil nodes. With the help of careful selection of threshold, accuracy of the proposed solution can be increased in large network. In addition, the proposed solution works better with the increased number of transaction requests. However, it may not be able to detect Sybil attack during the initial transaction requests as it is working based on the frequency of the transaction requests from the nodes.

V. Conclusion and Future Work

Sybil attack in the current implementations of public blockchain can cause severe impact. In this paper, we have shown an effect of the Sybil attack on throughput of blockchain network and proposed a solution directive to restrict it. The proposed solution performs distributed behavior monitoring of each node and finds suspect nodes which are not performing the fair forwarding of blocks. It increases the throughput of the system even if Sybil nodes are created in blockchain. It has good accuracy in detecting the Sybil nodes in blockchain. The experimental results are very encouraging and can be improved on large scale blockchain network by considering a large number of nodes which can cause a serious impact on blockchain. In future, the error rate of the proposed solution can be reduced by incorporating trust score for each suspected node. In addition, suspect list prevention can be considered.

References

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” [Online] Available: https://bitcoin.org/bitcoin.pdf

[2] Q. K. Nguyen, “Blockchain - A Financial Technology for Future Sustainable Development,” 3rd International Conference on Green Technology and Sustainable Development (GTSD), Kaohsiung, 2016, pp. 51-54.

[3] I-C. Lin, and T-C. Liao, “A Survey of Blockchain Security Issues and Challenges,” International Journal of Network Security, vol. 19, no. 5, 2017, pp. 653-659.

[4] J. R. Douceur, “The sybil attack,” in the First International Workshop on Peer-to-Peer Systems, ser. IPTPS ’01. London, UK: SpringerVerlag, 2002, pp. 251–260.

[5] M. Conti, S. K. E, C. Lal, and S. Ruj, “A Survey on Security and Privacy Issues of Bitcoin,” in IEEE Communications Surveys & Tutorials, 2018. DOI: 10.1109/COMST.2018.2842460

[6] I. Eyal and E. G. Sirer, “Majority is not enough: Bitcoin mining is vulnerable," Financial Cryptography, 2014, pp. 1-18.

[7] T. Ruffing, P. Moreno-Sanchez and A. Kate, “P2p mixing and unlinkable bitcoin transactions,” NDSS Symposium, 2017, pp. 1-15.

[8] T. Ruffing, P. Moreno-Sanchez, A. Kate, “CoinShuffle: Practical decentralized coin mixing for Bitcoin,” In: Proc. of the 19th European Symposium on Research in Computer Security (ESORICS’14), Springer, 2014, pp. 345–364.

[9] T. Ruffing and P. Moreno-Sanchez, “Valueshuffle: Mixing confidential transactions for comprehensive transaction privacy in bitcoin,” International Workshops on Financial Cryptography and Data Security: FC 2017, 2017, pp. 133–154.

[10] S. Bojja Venkatakrishnan, G. Fanti and P. Viswanath, “Dandelion: Redesigning the bitcoin network for anonymity,” Proc. ACM Meas. Anal. Comput. Syst., vol. 1, no. 22, 2017, pp. 1–34.

[11] M. H. Ibrahim, “Securecoin: A robust secure and efficient protocol for anonymous bitcoin ecosystem,” International Journal of Network Security, vol. 19, 2017, pp. 295–312.

[12] C. Decker, and R. Wattenhofer, “Information Propagation in the Bitcoin Network,” 13-th IEEE International Conference on Peer-to-Peer Computing, 2013.

[13] L. Xiao, L. J. Greenstein, N. B. Mandayam and W. Trappe, "Channel-Based Detection of Sybil Attacks in Wireless Networks,” in IEEE Transactions on Information Forensics and Security, vol. 4, no. 3, 2009, pp. 492-503.

[14] J. Newsome, E. Shi, D. Song and A. Perrig, “The sybil attack in sensor networks: Analysis and defenses,” Proc. Int. Symp. Information Processing in Sensor Networks (IPSN), 2004, pp. 259-268.

[15] C. Yu , S. Gupta and A. Agrawal, “Location based Technique to prevent Sybil attack in wireless sensor networks,” International Journal of Reliable Information and Assurance, vol. 5, no. 1, 2017, pp. 1-8.

[16] L. Xu, S. Chainan, H. Takizawa and H. Kobayashi, “Resisting Sybil Attack By Social Network and Network Clustering,” 10th IEEE/IPSJ International Symposium on Applications and the Internet, Seoul, 2010, pp. 15-21.

[17] B. Triki, S. Rekhis, M. Chammem and N. Boudriga, “A privacy preserving solution for the protection against sybil attacks in vehicular ad hoc networks,” 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC), Dubai, 2013, pp. 1-8.

[18] G. Bissias, A. Pinar, O. Brian, N. Levine and M. Liberatore, "Sybil-Resistant Mixing for Bitcoin,” Workshop of privacy in the electronic society, 2014, pp. 149-158.

[19] S. J. Samuel and B. Dhivya, “An efficient technique to detect and prevent Sybil attacks in social network applications,” 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, 2015, pp. 1-3.

[20] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll and E. W. Felten, “Mixcoin: Anonymity for bitcoin with accountable mixes,” in 18th International Conference, FC 2014. Springer Berlin Heidelberg, 2014, pp. 486–504.

[21] L. Valenta and B. Rowan, “Blindcoin: Blinded, accountable mixes for bitcoin,” in Financial Cryptography Workshops, 2015, pp. 112-126.

[22] E. Heilman, L. Alshenibr, F. Baldimtsi, A. Scafuro and S. Goldberg, “Tumblebit: An untrusted bitcoin-compatible anonymous payment hub,” NDSS Symposium, 2017, pp. 1-36. http://eprint.iacr.org/2016/575.

[23] P. Winter, R. Ensafi, K. Loesing, and N. Feamster, “Identifying and characterizing Sybils in the Tor network,” 25th USENIX Security Symposium, 2016, pp. 1169-1185.

[24] J. Baczuk, “How to Generate a Bitcoin Address—Step by Step,” 2018. [Online.] Avaiable: https://medium.com/coinmonks/how-to-generate-a-bitcoin-address-step-by-step-9d7fcbf1ad0b

[25] N. Koblitz, “Elliptic curve cryptosystems,” in Mathematics of Computation, 1987, pp. 203-209.

[26] How to Set Up a Bitcoin Miner. [Online.] Available: http://www.coindesk.com/information/how-to-set-up-a-miner

[27] G. Nash, “Build the Tiniest Blockchain,” 2017. [Online.] Available: https://medium.com/crypto-currently/lets-build-the-tiniest-blockchain-e70965a248b

[28] D. V. Flymen, “Learn Blockchains by Building One,” 2017. https://hackernoon.com/learn-blockchains-by-building-one-117428612f46

[29] Build Your Own Blockchain: A Python Tutorial. http://ecomunsing.com/build-your-own-blockchain