Security Analysis of Sharding in the Blockchain System

Dachao Yu*, Hao Xu*, Lei Zhang*, Bin Cao†, Muhammad Ali Imran*

* School of Engineering, University of Glasgow, Glasgow, G12 8QQ, UK
† Beijing University of Posts and Telecommunications, Beijing, China

Email: d.yu.2@research.gla.ac.uk, h.xu.2@research.gla.ac.uk,
lei.zhang@glasgow.ac.uk, caobin65@163.com, muhammad.imran@glasgow.ac.uk

***Abstract*—The design of sharding aims to solve the scalability challenge in a blockchain network. Typically, by splitting the whole blockchain network into smaller shards, the transaction throughput can be significantly improved. However, distributing fewer attesting nodes for transactions in a shard may cause higher security risks. This paper analyzes the security level of transaction verification in different types of shards and transactions. The analyzed result indicates that the size of shards and validating nodes number may influence the transaction security in shards. And the random distribution of attesting nodes can reduce such influence and improve the reliability of consensus in shards.**

***Index Terms*—Blockchain, sharding, security, random distribution**

I. Introduction

In the last decade, Bitcoin [1], Ethereum [2], and other public blockchain networks have drawn public attention to blockchain technology. Public chain with consensus mechanism such as Proof of Work (PoW) has built decentralized databases or ledgers to store tamper-proof transaction records and smart contracts. The principle of consensus mechanisms in blockchain transaction verification calls all validating nodes that engage in the blockchain network to process the verification of transactions. For example, in the PoW, once the validating nodes that have more than half of the computing power can confirm the transaction block, the transaction will be recorded in the decentralized ledger permanently, which means the trust in those transactions is distributed to all validating nodes. The transaction in blockchain with PoW will always be secure unless validating nodes who have over 51% computing power are malicious, which could cause a Sybil attack. With numerous validating nodes in the blockchain system, the probability that the attacker takes control of over 51% validating power in a transaction will be negligible. Therefore, the transaction would be reliable in the public chain networks with large validating node sets.

However, most large-scale blockchain networks have a common problem with scalability and efficiency. In PoW, nodes need to consume time and computing power to solve the complex hash function. The block broadcasting is also time-costing because a completed transaction needs verification and state storage from every miner node. Therefore, the transaction throughput in a public blockchain network is much lower than traditional centralized transaction processors such as VISA, which can process thousands of transactions per second. In order to increase the transaction throughput in the blockchain system, the developers in public blockchain initially try to extend the size of the block to verify more transaction information in one block. But the growing block size overloads the communication throughput and storage space in the database, which has less benefit in efficiency improvement.

To solve the scalability and throughput issue together, a concept called sharding, which is derived from a distributed database will be implemented in new generation blockchain networks such as Ethereum 2.0 [3]. The sharding in the blockchain system aims to separate nodes into different groups called shards, which can process transactions in parallel. Sharding can be categorized in ways of transactions or ledger storage. The transactions in sharding includes non-cross-shard or cross-shard transaction [4]. Non-cross-shard transaction means the transaction happens in a single shard and validating nodes of this transaction are all the nodes in this single shard. According to the principle of blockchain transaction verification, the size of this single shard is closely related to the security level of non-cross shard transactions in it. Cross-shard transaction refers to the transactions that happen between different shards, and the validating nodes are selected randomly from all shards even though most of them are unrelated to this transaction. Moreover, the type of sharding can be defined as transaction sharding or state sharding based on the way of ledger storage. Transaction sharding means every node of all shards will store a completed ledger that contains all verified transactions like traditional blockchain networks. In the state sharding, nodes in a shard will only store the verified transactions that have been processed by nodes in this shard instead of all shards. Therefore, it may require less storage space. Shards from the same blockchain network can implement different consensus mechanisms to fit the requirement of the decentralized application (dApp), which solves the scalability problem in a traditional blockchain system. In wireless blockchain system [5], the sharding can reduce the communication complexity of different consensus protocols

[6] [7], which save the cost of corresponding communication resources in the blockchain network [8].

Several sharding protocols have been developed in a new blockchain network. They are making efforts to improve the scalability of systems even though there are still drawbacks to these protocols. Elastico [9] is the first sharding protocol design for a public blockchain network. It has combined Bitcoin PoW protocol and standard BFT but only concerns about the way of transactions and network sharding. The formation of committees highly improves transaction throughput in the blockchain network, which is approximately proportional to the number of committees (shards). However, the transaction latency in Elastico is not affordable, even if the sharding number is small. Additionally, the normal committees in Elastico tend to have a limited number of nodes. This feature leads to the fact that after several transaction epochs, the probability of transaction failure could be tremendous. Even though Elastico has many drawbacks as a sharding protocol, it still has pointed out a direction to advanced sharding development for later design.

OmniLedger [10] is deployed as the Decentralized Ledger (DL) with a sharding structure. It is based on ByzCoin [11] and Hybrid consensus to select representative attesting nodes via scalable collective signing [12] [13]. RandHound [14] is implemented in OmniLedger to distribute attesters to shard securely and ensure that shards are large enough to resist potential attacks. A two-phase client-driven lock/unlock protocol Atomix to let transactions commit or abort atomically during cross-sharding transactions. OmniLedger supports trust-but-verify validation to reduce transaction latency in low-value payment cases, and it allows the validators to switch between different shards securely and efficiently. However, the OmniLedger system epochs are time-consuming, and it requires advanced anti-censorship to detect unfairly censored transactions.

Ethereum, as the first decentralized Blockchain platform, implements a Turing-complete programming language for smart contracts development. The sharding in Ethereum 2.0 (ETH 2.0) aims to solve the severe issue of low scalability and transaction throughput in the ETH 1.0 network. Shards in ETH 2.0 may use different consensus mechanisms to reach the requirements of their own scenarios. The beacon chain, as the essential structure of ETH 2.0, will be implemented in Stage 0 of development. The primary function of the beacon chain is assigning attesting committee randomly to verify the transactions or smart contracts in 1024 shards. The communication between the beacon chain and shards, which is through crosslinks, will generally be cross-shard communication. Beacon chain and other shards will use Casper FFG [15] to determine the canonical chain with Proof of Stake. Even though the development plan of Phase 0 in ETH 2.0 is explicit, the details of protocols have not been finalized [16].

All these references above only talk about the performance of their unique design even though they use similar random nodes distributed mechanisms. Therefore, it lacks a kind of general method to analyze the performance of sharding with random nodes distribution. The main contribution in this study is building a randomly nodes distributed sharding model to analyze the security performance with the conditions of cross-shard/non-cross-shard transactions and transaction/state sharding. The model initially calculates the probability P of the assigned number of nodes in any shard. With this probability, the probability of secure transaction in any shard can be determined, which represent the security level of the shard. Then the transaction throughput and communication throughput, which is related to the probability of secure transaction, will be conducted. The analysis is indicated in section III. This shard analyzing model provides a general pattern to figure out the security performance of a specific sharding design and improve the reliability of it.

Section II introduces the mathematical principle of the model to analyze the security level in both non-cross-shard and cross-shard transactions of sharding. Section III shows the simulation result of models that given in section II. Section IV concludes this model and what aspect needs to be improved in sharding for practical usage.

II. System Model of Sharding

The property of Decentralized Network (DN) indicates that the security level of DN is positively correlated to the number of validating nodes. Yet, the sharding divided the whole blockchain network into several smaller pieces, which will cause an inevitable decrease of security in transactions because in sharding, the nodes engage in transaction validating is less than the original blockchain system. In order to reduce the influence from less validating nodes, most sharding designs use Verifiable Random Function (VRF) [17] to provide randomness for validating nodes distribution. Before distributing the validating nodes randomly, the nodes will be categorized according to the specific bits of their hash value, which can be adjusted to change the difficulty of assigning function. Some researchers have already optimized the nodes distribution methods in sharding to improve the secure transaction rate, such as Game-theoretic analysis [18] and Trust-Based shard distribution [19].

The analysis of this model depends on the types of sharding (transaction/state) and transaction (cross-shard/non-cross-shard). These types of sharding designs are presented in Fig. 1 and Fig. 2. The model assumes that N nodes with the same computing power are randomly distributed into M shards initially, and the total N nodes contain H malicious nodes in them, which will violate the transaction verification. If the malicious nodes number h reaches 50% of the total nodes number k in a shard, the transaction validated by these nodes will fail, and it would not be recorded in the ledger. Normally, in distributed systems, the malicious node ratio R can influence the reliability of transaction verification in blockchain, which is represented in equation 1

$$R = H/N. \tag{1}$$

This section investigates the security performance in the sharding system when the malicious nodes ratio R change. Table I shows all parameters set in this model.

TABLE I: Parameter setting in sharding security analysis

Notation	Definition
N	Total number of nodes in the network
M	Number of shards
H	Total number of malicious nodes in the network
R	The ratio of malicious nodes in total nodes
k	The number of nodes distributed in any shards
h	The number of malicious nodes distributed in any shards
$P(k)$	Probability of k nodes distributed in a shard
$P(m)$	Probability of m nodes distributed in validating set
P_h	Probability of h malicious nodes in a shard
P_c	Probability of secure transaction in a shard
$P_H(k)$	Probability of h malicious nodes in a k nodes shard
$P_s(k)$	Probability of secure transaction in a k nodes shard

A. Non-cross-shard transaction

Fig. 1 presents the main stages of the non-cross-shard transaction. The first stage is the node distribution: All nodes from the original network are randomly distributed in several shards. During consensus, nodes cannot alter the shards they have been assigned to. The second stage is transaction verification: transactions can only happen between nodes that belong to the single shard, and the transaction can only be verified by nodes in this specific shard. The last stage is ledger storage, and it could be different in the transaction or state sharding. If all transaction records are still stored in every node like a traditional blockchain system, it will be defined as a transaction or network sharding. But if nodes in a shard only store the transaction processed in this specific shard, it will be state sharding. State sharding may require less memory space. However, it may conflict with the purpose of decentralization in the blockchain network. So this trade-off needs to be optimized in innovative sharding designs. Before

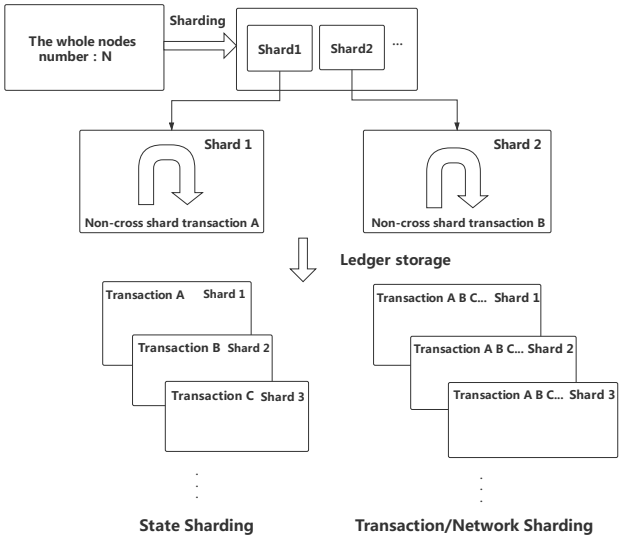

Fig. 1: The non-cross-shard transaction

analyzing the security performance of sharding, it is necessary to figure out the random process of nodes distribution to shards. The random distribution in sharding is similar to the dice tossing problem. Each shard represents a side of dice that has an equivalent probability of selecting an individual node. Therefore, the probability $P(k)$ that k nodes are randomly distributed to a shard can be given by a binomial distribution, which is implied in equation 2, which is influenced by shards number M and total nodes number N

$$P(k) = \binom{N}{k} \cdot (\frac{1}{M})^k \cdot (\frac{M-1}{M})^{N-k}. \tag{2}$$

The equation 2 indicates that the probability $P(k)$ is influenced by shards number M and total nodes number N.

With the theoretical probability $P(k)$ in every shard, the security level of sharding transactions can be analyzed. The probability $P_h(k)$ that N nodes within H malicious nodes have randomly distributed k nodes to a shard within h malicious nodes in this shard. $P_h(k)$ is similar to the problem of product sample check. There are total N products within H poor product. k products are selected from them, and $P_h(k)$ is the probability of h poor products from these k products, which follows the principle of Hyper-geometric Distribution. The equation 3 indicates this probability distribution

$$P_h(k) = f(h,k,H,N) = \frac{\binom{H}{h}\binom{N-H}{k-h}}{\binom{N}{k}}. \tag{3}$$

Once $P_h(k)$ is determined, the successful transaction rate in this k nodes shard could be calculated. With the consensus of PoW, if we assume the node number k in a shard is set and each node has identical computing power, the probability P_c of secure transaction processed in the shard can be accumulated by $P_h(k)$ until h reaches 50% of k

$$P_c(k) = \sum_{h=1}^{\frac{k}{2}} P_h(k) = \sum_{h=1}^{\frac{k}{2}} \frac{\binom{H}{h}\binom{N-H}{k-h}}{\binom{N}{k}}. \tag{4}$$

To obtain the eventual successful transaction rate $P_s(k)$, it's necessary to consider both of the randomly nodes distribution k and malicious nodes distribution h in the shard. Therefore, for the probability that a shard has k nodes within h malicious nodes $P_H(k)$ is the production of $P(k)$ and $P_h(k)$ because these probabilities are independent to each other

$$P_H(k) = P(k) \cdot P_h(k). \tag{5}$$

If h is over 50% of k, the transaction verification will be controlled by malicious users, and the transaction in this shard will be insecure. The probability of secure transactions in one shard with specific nodes number k is $P_s(k)$, which is accumulated by the probability $P_H(k)$ that h is less than half of k while h and k are both uncertain

$$P_s(k) = \sum_{h=1}^{\frac{k}{2}} P_H(k) = P_c(k) \cdot P(k). \tag{6}$$

According to equation 4, 5, 6, $P_s(k)$ is only depends on the amount of k and ratio of malicious nodes R.

B. Cross-shard transaction

The main stages of cross-shard transaction in Fig. 2 also include sharding and ledger storage. The difference from a non-cross-shard transaction is that the transaction can happen between nodes from different shards. It could be quite complicated in the stage of ledger storage if cross-shard transactions records are stored in the way of state sharding because the sharding system can hardly determine which parts of nodes are responsible for storing the records. The mainstream idea tends to let all nodes that are related to the transaction keep the ledger consistent, including transaction nodes and validating nodes. However, this mechanism may cause the issue that some nodes from the same shard may have different ledger contents, which means the blockchain system may lose state consistency. In a cross-shard transaction, the probability $P(k)$

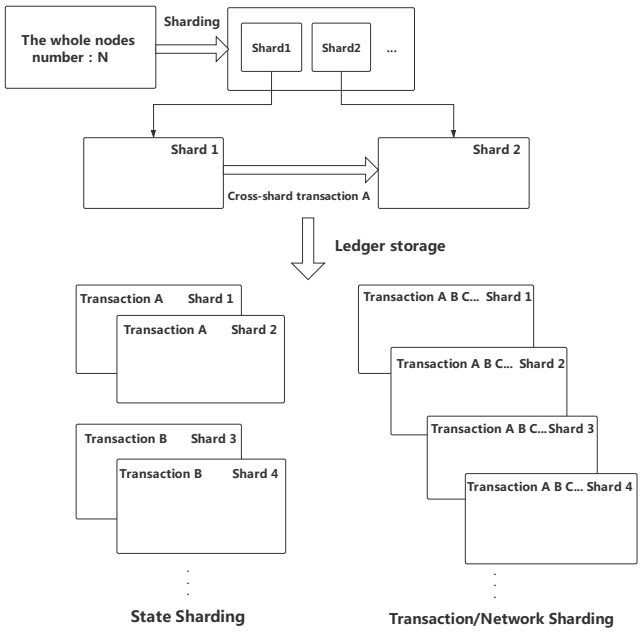

Fig. 2: The cross-shard transaction

that k nodes are assigned to a shard is the same as equation 2 in the non-cross-shard transaction because the number of total nodes N and shards M do not change and the nodes are randomly assigned to shards. However, the security analysis in cross-shard is different from non-cross-shard. The nodes participants in transaction verification are not only from the transaction-relevant shards but also randomly assigned nodes from the whole blockchain network. In the case of validating, it assumes the probability that any node participant validating is 20%. Therefore, the probability $P(m)$ that m nodes are chosen to be validators in a transaction is

$$P(m) = \binom{N}{m}(\frac{1}{5})^m(\frac{5-1}{5})^{N-m}. \tag{7}$$

In cross-shard transaction, the probability that random x nodes within h malicious nodes engage in transaction validating from the blockchain network is P_H:

$$P_h(m) = f(h, m, H, N) = \frac{\binom{H}{m}\binom{N-H}{m-h}}{\binom{N}{m}}, \tag{8}$$

$$P_H(m) = P(m) \cdot P_h(m). \tag{9}$$

When h is less than 50% of m, the transaction will be secure. The probability of secured transaction with m validators P_s will be accumulated by P_H as malicious nodes number h is less than half of validating nodes number m

$$P_s(m) = \sum_{h=1}^{\frac{m}{2}} P_H(m) = \sum_{h=1}^{\frac{m}{2}} P(m) \cdot P_h(m). \tag{10}$$

Compared with the non-cross-shard transaction, the security of cross sharding transaction is influenced by malicious nodes rate in randomly selected validating nodes set instead of the malicious nodes number k in the shard. If the size of validating nodes set is large enough, the difficulty in accomplishing Sybil attacks in a cross-shard transaction will be much more incredible than a non-cross-shard transaction. Therefore, the security of cross-shard transactions can be improved. However, the cross-shard transaction may require more resources to support the communication for a massive validating network and solve the problem of ledger consistency.

III. Simulation Results

In the simulation of section III, to calculate the probability, the number of nodes N is set as 1000, and the number of shards M is set as 10. Through comparing the tendency curves of $P(k)$ in theory and simulation while the nodes number k in a shard changing, the correctness of equation 1 will be determined. The random distribution progress in simulation is repeated 10^5 times to get a mean of probability $P(k)$ and then compared with the theoretical value of $P(k)$ in equation 2.

The comparison between analytical and simulated results is shown in Fig. 3, which concurs the analytical value in equation 2 because the simulation points are overlapped to the analytical curve of $P(k)$. The result presents that $P(k)$ is mainly distributed at $k = \frac{N}{M}$, which means the size of shards will be close if the nodes are random distributed into the shards before the consensus progress.

A. Non-cross-shard transaction

Subsection III-A presents the crucial probabilities that represent the security level of non-cross-shard transactions in sharding. In Fig. 4, the probabilities P_c is indicated that it can be influenced by R. When R is less than 50%, the probability curve of P_c will converge to 100% as h increasing. The Subsection III-A presents the crucial probabilities that represent the security level of non-cross-shard transaction in sharding. In Fig. 4, the probabilities P_c is indicated that it can be influenced by R. When R is less than 50%, the probability curve of P_c will converge to 100% as h increasing. The simulation result reveals that the probability P_H of the ratio of malicious nodes in a shard depends on the ratio of malicious

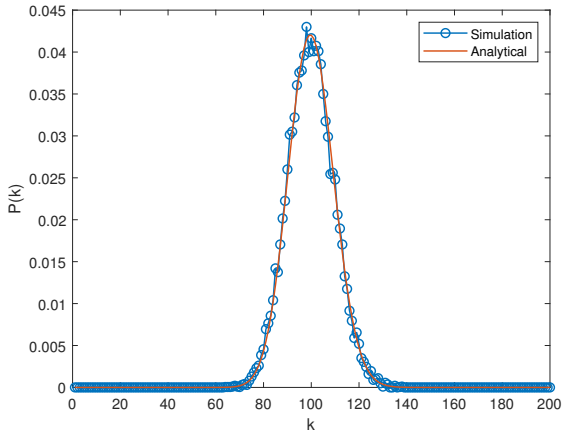

Fig. 3: Probability $P(k)$ of nodes number k distributed in a shard

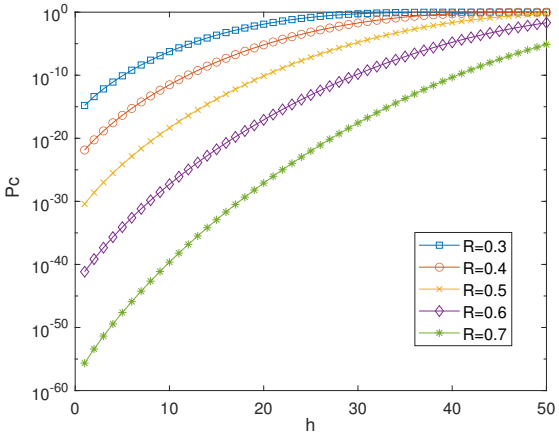

Fig. 4: Probability P_c of secure transaction in a shard

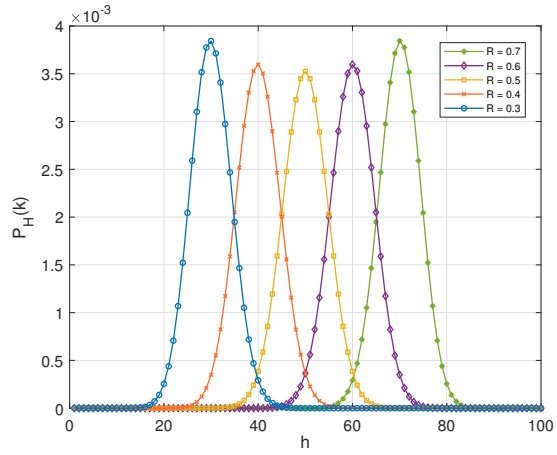

Fig. 5: Probability of malicious nodes number h in a shard contains k nodes (P_H)

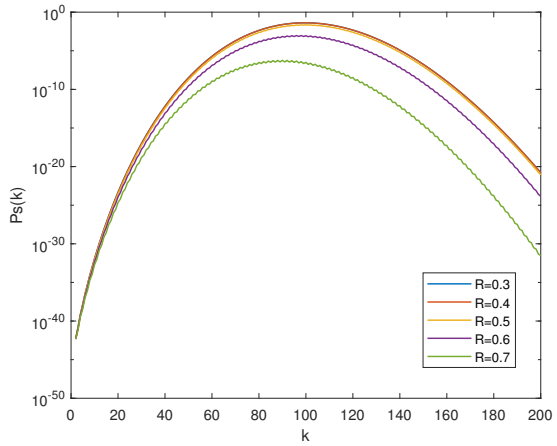

Fig. 6: Probability of secure transaction in a shard with k nodes (P_S)

nodes to all nodes in the whole blockchain network R. As R changes from 0.3 to 0.7, the corresponding number of malicious nodes to the peak value of $P_H(k)$ will rise up, which is indicated in Fig. 5. The result in Fig. 6 implies the dominated influence in $P_S(k)$ given by the rate of malicious nodes R. As R is less than 0.5, the peak value of $P_S(k)$ could be over 10^{-2}, which is considerable for a shard transaction when all values from the same curve are accumulated. However, if the number of malicious nodes M is over 50% of N, The successful transaction rate $P_S(k)$ will be less than 10^{-5}, which could be negligible from the perspective of transaction security.

B. Cross-shard transaction

The simulation result of $P(m)$ in Fig. 7 is similar to $P(k)$ of non-cross-shard transaction in Fig. 3. But the peak value of probability curve is changed because validating nodes number m in cross-shard transaction differs from shard's nodes number k in non-cross-shard. In Fig. 8, P_c in cross-shard transaction is similar to the probability of non-cross-shard transaction in Fig. 4. The probability P_c still converges to 100% as R is less than 0.5. But the P_c is never higher than 10^{-4} as R is over 0.5, even if h changes.

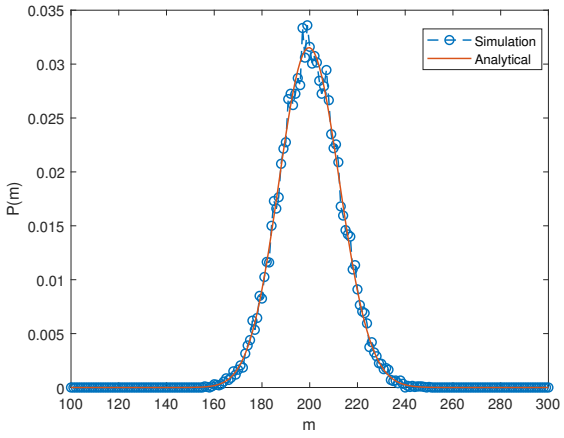

Fig. 7: Probability of validating nodes number m in cross-shard transaction

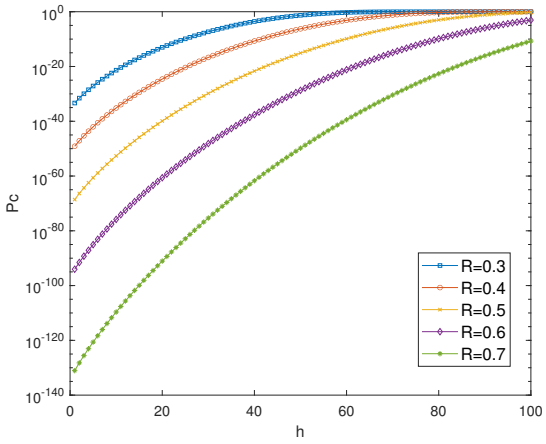

Fig. 8: Probability P_c of secure cross-shard transaction

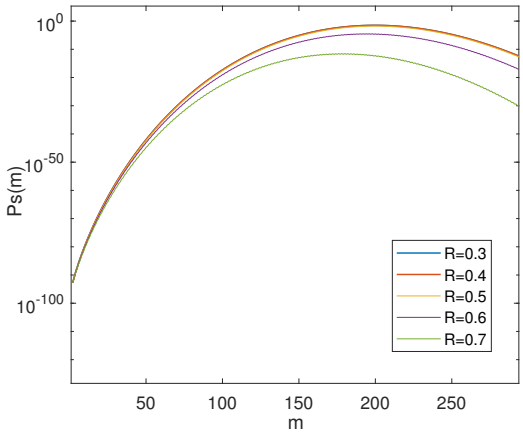

Fig. 9: Probability of secure cross-shard transaction (P_S)

As R is 0.3 and 0.4, the value of $P_S(m)$ is close in Fig. 9, which means R has less influence in the successful transaction rate of cross-shard transaction. But once malicious nodes occupy the majority of network ($R > 0.5$), the secure transaction can hardly happen.

The model assumes that the size of validating nodes set m in the cross-shard transaction is larger than the size of shards k in the non-cross-shard transaction, which causes the corresponding vertical peak values of all curves are right-shifted in the model of cross-shard transaction. It indicates that if attackers want to compromise the security of cross-shard transactions, they need to cost more computing power than non-cross transactions. In other words, the cross-shard transaction is normally more secure than the non-cross-shard transaction in shards.

IV. Conclusion

Most of the current sharding designs in blockchain systems use a random distribution method to assign validating nodes from the whole blockchain network to complete the consensus. The analysis of the sharding model indicates the security level can be affected by the rate of malicious nodes in both cross-shard transactions and non-cross-shard transactions. In the future, more advanced validating nodes distribution methods and consensus algorithms should be explored and applied in a new blockchain system to reduce the malicious node's influence on the sharding system's security performance. The sharding could be one of the practical ways to improve the scalability of blockchain networks.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," tech. rep., Manubot, 2019.

[2] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[3] M. Swan, *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.

[4] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 931–948, 2018.

[5] B. Cao, L. Zhang, M. Peng, and M. Imran, "Wireless blockchain: Principles, technologies and applications," 2020.

[6] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer pbft consensus for blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1146–1160, 2021.

[7] D. Yu, W. Li, H. Xu, and L. Zhang, "Low reliable and low latency communications for mission critical distributed industrial internet of things," *IEEE Communications Letters*, vol. 25, no. 1, pp. 313–317, 2021.

[8] L. Zhang, H. Xu, O. Onireti, M. A. Imran, and B. Cao, "How much communication resource is needed to run a wireless blockchain network?," *arXiv preprint arXiv:2101.10852*, 2021.

[9] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 17–30, 2016.

[10] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 583–598, IEEE, 2018.

[11] E. Syta, P. Jovanovic, E. K. Kogias, N. Gailly, L. Gasser, I. Khoffi, M. J. Fischer, and B. Ford, "Scalable bias-resistant distributed randomness," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 444–460, Ieee, 2017.

[12] D. Boneh, M. Drijvers, and G. Neven, "Compact multi-signatures for smaller blockchains," pp. 435–464, 2018.

[13] E. Syta, I. Tamas, D. Visher, D. I. Wolinsky, P. Jovanovic, L. Gasser, N. Gailly, I. Khoffi, and B. Ford, "Keeping authorities" honest or bust" with decentralized witness cosigning," pp. 526–545, 2016.

[14] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *25th usenix security symposium (usenix security 16)*, pp. 279–296, 2016.

[15] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv preprint arXiv:1710.09437*, 2017.

[16] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, *et al.*, "Spanner: Google's globally distributed database," *ACM Transactions on Computer Systems (TOCS)*, vol. 31, no. 3, pp. 1–22, 2013.

[17] Y. Dodis and A. Yampolskiy, "A verifiable random function with short proofs and keys," in *International Workshop on Public Key Cryptography*, pp. 416–431, Springer, 2005.

[18] M. H. Manshaei, M. Jadliwala, A. Maiti, and M. Fooladgar, "A game-theoretic analysis of shard-based permissionless blockchains," *IEEE Access*, vol. 6, pp. 78100–78112, 2018.

[19] J. Yun, Y. Goh, and J.-M. Chung, "Trust-based shard distribution scheme for fault-tolerant shard blockchain networks," *IEEE Access*, vol. 7, pp. 135164–135175, 2019.