Corking by Forking: Vulnerability Analysis of Blockchain

Shengling Wang*, Chenyu Wang*, Qin Hu†(Corresponding Author)
*College of Information Science and Technology, Beijing Normal University, Beijing, China.
Email: {wangshengling@bnu.edu.cn, wcyhenry@mail.bnu.edu.cn}
†Department of Computer Science, The George Washington University, Washington D.C., USA.
Email: qinhu@gwmail.gwu.edu

***Abstract*—The great market success of Blockchain makes it an extremely valuable target for attackers. A well-known attack in Blockchain is the *forking attack*, where divergent blockchains are produced for inserting some new features to facilitate security breaches. The state-of-the-art works mostly focus on how to detect attacks in real-time transactions, which is in hindsight and cannot deter the forking attack from the root. To take precautions, we employ the large deviation theory to study the vulnerability of blockchain networks incurred by intentional forks from a micro point of view, boosting forward-looking and strategic planning mechanisms for resisting the forking attack. Our study is *fine-grained*, because it offers not only the vulnerability probability of a blockchain network but also its decay speed, through which we find setting the parameter related to the robust level has more power than enhancing the computer power in speeding up the failure of attacks. This finding is valuable since it renders an opportunity to improve the robustness of a blockchain network in a cost-efficient way. Our analysis is *complementary*, since it studies both the impacts of the computational power as well as the number of confirmations on the vulnerability of a blockchain network, providing a theoretical basis to design reasonable schemes for invigorating a blockchain network from technical as well as managerial levels. Extensive experiments carried out on a large-scale cloud platform running the Ethereum protocol show the experimental and analytical results match well, verifying the effectiveness of our analysis.**

I. Introduction

Last few years witness a revolutionary trustable and sharable innovation colloquially known as Blockchain[1]. Essentially, a blockchain is a distributed consensus ledger, which records all digital transactions and events, owned and monitored by all participants without any central control. Besides the merits of *decentralization*, *transparency*, *self-regulating* and *traceability*, Blockchain enables instant value transfer, which radically improves our payments experience, driving us from Internet of information to *Internet of value*. These merits facilitate Blockchain to become the underlying fabric of mainstream crypto-currency systems such as Bitcoin [1] and Ethereum [2]. It is predicted that the market size of Blockchain will reach USD 7,683.7 Million by 2022 at a compound annual growth rate of 79.6% [3].

The great market success of Blockchain makes it an extremely valuable target for attackers. A well-known attack in Blockchain is the *forking attack*, where divergent blockchains are produced for inserting some new features to facilitate security breaches. We call such a forged blockchain the *intentional* fork, which is different from the *accidental* fork generated due to inconsistent views on states of blockchains. A typical example of the forking attack is *double spending*, where the same set of crypto-currencies is spent more than once. It is the features of data and blockchain networks that offer the chance to the forking attack since the former ease the publishing of fake transactions or events in a blockchain network while the latter make it possible to approve an intentional fork.

On one hand, data has a trait of *a free commons* [4], i.e., cheap to copy and transfer, making it a light work to announce invalid digital transactions or events to a blockchain network. On the other hand, the latency in propagation and processing of transactions or events over a blockchain network may yield to different views on blockchains, which defies consensus, breaching the uniqueness of value exchange, and hence opens a gate to reach a consensus on an intentional fork. Leveraging the above traits, an adversarial attacker with significant computational power can append invalid transaction or event records to the blockchain for carrying out forking attacks.

The trait of the free commons is the nature of data, leaving no room for us to resist the forking attack from the perspective of data. However, the creation and prorogation of intentional forks are influenced by multi-dimensional factors, such as the computational power and the transmission performance of blockchain networks, which can be adjusted and optimized. Hence, refraining from intentional forks is a feasible countermeasure to guard against the forking attack (e.g., double-spending). Such a job is meaningful since intentional forks cork the healthy development of blockchain networks, leading to their vulnerability.

The state-of-the-art works [5]–[8] on the typical forking attack, namely double-spending, mostly focus on how to detect attacks in real-time transactions, which is in hindsight and cannot deter the forking attack from the root. To take precautions, an in-depth analysis of forking attacks needs to be taken. In detail, if malicious nodes have dominant computational power (e.g., more than 50% of total computational capacity), they can control the whole blockchain network, succeeding in attacks with overwhelming probabilities. However, in reality, the computational power of malicious nodes hardly competes with

[1] In this paper, we use *Blockchain* to denote the technology while *blockchain* to indicate a chain of blocks.

that of all honest nodes. Unexpectedly, the forking attack may still succeed even malicious nodes are at such an unfavorable position. The reason behind this fact is that the fluctuation of the computational power difference between malicious nodes and honest ones temporally breaks the dominance of the latter. Hence, to restrain intentional forks, it is imperative to analyze the short-term effects of the strength antagonism between malicious nodes and honest ones on the vulnerability of a blockchain network.

In this paper, we take advantage of the large deviation theory [9] to study the vulnerability of blockchain networks incurred by intentional forks from a micro point of view, where transient states of the computational power provided by malicious nodes and honest ones are taken into consideration. Our study is *fine-grained*, because it offers not only the vulnerability probability of a blockchain network but also its decay speed, which can be utilized to answer two fundamental questions: *when should we take countermeasures against the forking attack and how fast should we do?* Our analysis is *complementary*, since it studies both the impacts of the computational power as well as the number of confirmations on the vulnerability of a blockchain network, providing a straightforward explanation on the question, namely *to what extent we could boycott forking attacks with how much computational power and how many blocks needed to confirm a transaction at least?* The above answers can boost forward-looking and strategic planning mechanisms for resisting the forking attack.

Conclusively, our contributions can be summarized as follows:

- The strength antagonism between malicious nodes and honest ones is studied from the microcosmic perspective, based on which the probability distribution of a blockchain network being vulnerable is deduced.
- The decay speed of the probability that a blockchain network becomes vulnerable is derived, through which we find setting the parameter related to the robust level has more power than enhancing the computer power in speeding up the failure of attacks. This finding is valuable since it renders an opportunity to improve the robustness of a blockchain network in a cost-efficient way.
- A concept of the effective robust level is introduced to indicate the minimum difference in the number of the blocks completed by malicious nodes and honest ones to confirm a transaction within a unit time for fighting the forking attack, according to which, we can design optimal defensive mechanisms from a management level.
- A concept of the effective computational power is defined to denote the minimum blocks completed by honest nodes during a time interval to guarantee the robustness of a blockchain network, whose solution provides a theoretical basis to design reasonable schemes for invigorating the blockchain network from a technical level.
- Extensive experiments carried out on a large-scale cloud platform running the Ethereum protocol show the experimental and analytical results match well, verifying the effectiveness of our analysis.

The rest of the paper is organized as follows. The related work is presented in Section II. Sections III and IV carry out the vulnerability analysis of Blockchain under the special and general cases, respectively. Our experimental evaluation results are reported in Section V. We conclude the paper in Section VI.

II. Related Work

In general, there are mainly three kinds of attacks in Blockchain, namely *selfish mining*, *double spending* and *eclipse attacks*.

Selfish mining is a malicious behavior where a hostile miner withholds some mined blocks to increase its relative revenue at the expense of other nodes. In [10], Eyal *et al.* concluded that selfish mining is feasible for any colluding group of miners in Bitcoin mining process and proposed a novel modification on the Bitcoin protocol to protect it from this attack. Based on this conclusion, Sapirshtein *et al.* [11] investigated the profit threshold to find out the minimum faction of resource when a selfish mining attack is successful, which plays as a bound for measuring the security of a blockchain system against this attack. While in [12], Tosh *et al.* studied the selfish mining in the blockchain-based cloud scenario with the consideration of different reward mechanisms. To prevent the blockchain from selfish attack, Zhang *et al.* [13] proposed a novel defense mechanism with a policy of forking-resolving that rejects blocks published with time delay and encourages those incorporating links to competing blocks of their predecessors.

Double spending, as we mentioned above, means the same set of coins is consumed in multiple transactions so that the absolute revenue of the attacker increases. O. Karame *et al.* [5] analyzed the problem of double-spending payment in the fast Bitcoin scenario and illustrated the ineffectiveness of countermeasures recommended by the Bitcoin developers in some special cases, based on which a modification on the Bitcoin implementation was proposed to facilitate identifying double-spending attacks for fast payment. Based on their prior work, they continued to design a lightweight detection method for double-spending attacks and further investigated the accountability for such misbehavior in [6]. And Gervais *et al.* [7] discovered that the current measures of scalability in Bitcoin could be exploited by an adversary to delay the blocks delivery for some nodes so as to enable them to execute double-spending transactions. For efficient detection of double-spending attacks in fast Bitcoin payment system, Liu *et al.* [8] proposed an artificial-immune-based model, which employs several immune-based nodes with a detection module in a Bitcoin network. To solve the double-spending problem for zero-confirmation transactions, a novel prevention mechanism was devised in [14] through exploiting the flexibility of the scripting language and a vulnerability of the elliptic curve digital signature algorithm.

Eclipse attacks refer to the isolation of victims where they can not access to the information in the decentralized network but only are allowed to communicate with the attacker, which is first proposed by Heilman *et al.* [15] in the Bitcoin's peer-to-peer network. Through successfully launching an eclipse attack, the attacker can further take advantage of the victims to implement more attacks related to the mining and consensus system, such as selfish mining and double spending mentioned above. Besides, eclipse attacks on Ethereums peer-to-peer network were studied in [16], which demonstrated the feasibility of launching eclipse attacks with only two hosts by exploring the vulnerabilities of Kademlia peer-to-peer protocol in Ethereum. While in [17], Nayak *et al.* proposed a kind of stubborn mining with long-shot gambles by combining the eclipse attack and selfish mining, and uncovered that the eclipsed victim could even benefit from being attacked sometimes due to the desired strategies of attackers.

III. Vulnerability Analysis for the Special Case

In this section, we take advantage of the large deviation theory to analyze the vulnerability of a blockchain network, where there are two kinds of nodes, namely honest nodes and malicious ones. The computationally expensive proof-of-work (PoW) is employed as the consensus protocol since it accounts for more than 90% of the total market capitalization of existing digital currencies [18]. PoW involves searching for a random number (nonce), whose average work is exponential in the number of zero bits required, making it a hard work for one single node to complete a block alone. Hence, nodes, in fact, carry out Bernoulli trails at a vast pace, which suggests that the block completion can be assumed as a Poisson process [19], [20].

Since the *main chain* is the consensus longest blockchain (ledger), the PoW of an honest node has a chance to be approved only if the honest node appends its blocks to the main chain. Differently, to carry out the forking attack (e.g., double spending), a malicious node should execute the PoW for the chain containing its fake transactions, namely the *fake chain*. If the fake chain is longer than the main one, the forking attack may happen, making the blockchain network vulnerable. To analyze such vulnerability, we conduct the study from simple to complex cases. In the following, we firstly analyze a special case that there is only one honest node and one malicious node. Though this case is simple, it is meaningful for our further analysis for a general case.

Assume that block completions occur with the rates of $\lambda > 0$ and $\Lambda > 0$ at the honest node and the malicious one respectively. As we mentioned above, the joint computational power of all honest nodes is usually larger than that of all malicious nodes. To coincide with this situation in the two-node blockchain network, we set $\lambda > \Lambda$. To study the vulnerability of a blockchain network from a micro point of view, we divide the whole time period $\mathbb{T}$ into multiple equal-sized intervals with each's length being d and analyze the transient states of the computational power provided by the malicious node and the honest one. Let C_t and A_t respectively be the total numbers of blocks completed by the malicious node and the honest one during the time period $[0, \mathbb{T})$, i.e., $C_t = c_1 ... + c_i ... + c_t$ and $A_t = a_1 ... + a_i ... + a_t$, where $t = \mathbb{T}/d$, c_i and a_i $(i = 1, 2, \cdots, t)$ are respectively the numbers of blocks completed by the malicious node and the honest one during the i^{th} interval. In addition, we define W_t as the difference in numbers of blocks completed by the malicious node and the honest one at the end of the t^{th} interval. That is,

$$\begin{aligned} W_t &= C_t - A_t \\ &= (c_1 + ... + c_t) - (a_1 + ... + a_t), \end{aligned} \tag{1}$$

where W_t reflects the computational power difference (i.e., the strength difference) between the malicious node and the honest one and $W_0 = 0$. The cumulant generating function of W_t can be described as

$$\psi(\theta) = \lim_{t \to \infty} \frac{1}{t} \log \mathbf{E}[\mathrm{e}^{\theta W_t}], \tag{2}$$

which is a function defined on $\theta \in \mathbb{R}$.

When the strength difference between the malicious node and the honest one is the largest, the blockchain network is the most vulnerable. Hence, to analyze the vulnerability of the blockchain network, we need to study the metric Q, which is the maximum value of W_t, i.e., $Q = \sup_{t \geq 0} W_t$. To be specific, Q is the largest difference in number of blocks completed by the malicious node and the honest one when $\mathbb{T} \to \infty$, reflecting their largest strength difference.

In fact, the vulnerability of a blockchain network depends on not only the largest strength difference between two antagonistic sides but also the security mechanism designed for the network. One simple but important security mechanism is to set the number of consecutive blocks that need to be appended to the longest blockchain to confirm a transaction. Hence, under such a security mechanism, it is important to study the probability of $Q > \Theta$, where Θ is defined as follow:

Definition III.1 (Robust level)**.** *The robust level* $\Theta > 0$ *is the required difference in number of the blocks completed by a malicious node and an honest one to confirm a transaction.*

Hence, the robust level Θ indicates the difficulty for a malicious node to carry out the forking attack. The bigger the robust level Θ is, the harder a malicious node attacks a blockchain network. Based on this definition, Θ denotes the least total number of blocks that the malicious node needs to generate more than the honest one. If $Q > \Theta$, the blockchain is vulnerable and vice versa. Hence, the probability $\mathbf{P}(Q > \Theta)$ reflects the vulnerability degree of the blockchain network, which is an important index for us to decide when we should take countermeasures against the forking attack.

Let $\Theta = lb$ for any $b > 0$. According to Cramer's theorem [9], when $l \to \infty$, $\mathbf{P}(Q > \Theta)$ could be expressed as $\mathbf{P}(Q > lb) \sim \exp(-lI(b))$. That is,

$$\lim_{l \to \infty} \frac{1}{l} \log \mathbf{P}(Q > lb) = -I(b), \tag{3}$$

where

$$I(b) = \inf_{t \geq 0} t\psi^*(\frac{b}{t}). \tag{4}$$

In (4), $\psi^*(\cdot)$ can be calculated as

$$\psi^*(x) = \sup_{\theta \in \mathbb{R}} \{\theta x - \psi(\theta)\}, \tag{5}$$

which is the convex conjugate or the Legendre transformation of $\psi(\cdot)$. According to (1) and (2), we have

$$\begin{aligned}\psi(\theta) &= \lim_{t\to\infty} \frac{1}{t} \log \mathbf{E}[\mathrm{e}^{\theta(C_t - A_t)}] \\ &= \lim_{t\to\infty} \frac{1}{t} \log \mathbf{E}[\mathrm{e}^{\theta C_t}]\mathbf{E}[\mathrm{e}^{-\theta A_t}] \\ &= \lim_{t\to\infty} \frac{1}{t} \log \mathbf{E}[\mathrm{e}^{\theta C_t}] + \lim_{t\to\infty} \frac{1}{t} \log \mathbf{E}[\mathrm{e}^{-\theta A_t}] \\ &= \psi_C(\theta) + \psi_A(-\theta),\end{aligned}$$

where $\psi_C(\cdot)$ and $\psi_A(\cdot)$ are the cumulant generating functions of C_t and A_t, respectively.

To further derive $\psi^*(x)$, we have the following theorem.

Theorem III.1. *$\forall x \in \mathbb{R}$, $\psi^*(x) = \inf_{y\in\mathbb{R}}\{\psi_C^*(y) + \psi_A^*(y - x)\}$.*

Proof: On one hand, we have

$$\begin{aligned}\psi^*(x) &= \sup_{\theta\in\mathbb{R}}\{\theta x - \psi(\theta)\} \\ &= \sup_{\theta\in\mathbb{R}}\{\theta x - \psi_C(\theta) - \psi_A(-\theta)\} \\ &= \sup_{\theta\in\mathbb{R}}\{\theta(y + x - y) - \psi_C(\theta) - \psi_A(-\theta)\} \\ &= \sup_{\theta\in\mathbb{R}}\{\theta y - \psi_C(\theta) + (-\theta)(y - x) - \psi_A(-\theta)\} \\ &\le \sup_{\theta\in\mathbb{R}}\{\theta y - \psi_C(\theta)\} + \sup_{\theta\in\mathbb{R}}\{(-\theta)(y - x) - \psi_A(-\theta)\} \\ &= \sup_{\theta\in\mathbb{R}}\{\theta y - \psi_C(\theta)\} + \sup_{\theta\in\mathbb{R}}\{\theta(y - x) - \psi_A(\theta)\} \\ &= \psi_C^*(y) + \psi_A^*(y - x).\end{aligned} \tag{6}$$

Since the above equation holds for any $y \in \mathbb{R}$, the following inequality can be obtained

$$\psi^*(x) \le \inf_y \{\psi_C^*(y) + \psi_A^*(y - x)\}. \tag{7}$$

On the other hand, due to $\psi^*(x) = \sup_{\theta\in\mathbb{R}}\{\theta y - \psi_C(\theta) + (-\theta)(y - x) - \psi_A(-\theta)\}$ as we proved in (6), there exists $y_0 \in \mathbb{R}$ which makes

$$\begin{aligned}\psi^*(x) &= \sup_{\theta\in\mathbb{R}}\{\theta y_0 - \psi_C(\theta)\} + \sup_{\theta\in\mathbb{R}}\{(-\theta)(y_0 - x) - \psi_A(-\theta)\} \\ &= \psi_C^*(y_0) + \psi_A^*(y_0 - x) \\ &\ge \inf_y\{\psi_C^*(y) + \psi_A^*(y - x)\}.\end{aligned} \tag{8}$$

According to (7) and (8), we can derive the theorem naturally. ■

Note that for any Poisson process X with $\mathbf{E}X = \mathbf{Var}(X) = \bar{\lambda}$, its cumulant generating function [21] can be calculated as $\log \mathbf{E}[\mathrm{e}^{\theta X}] = \bar{\lambda}(\mathrm{e}^\theta - 1)$.

Thus, $\psi_C(\theta)$ and $\psi_A(\theta)$ can be expressed as,

$$\psi_C(\theta) = \Lambda(\mathrm{e}^\theta - 1), \tag{9}$$

$$\psi_A(\theta) = \lambda(\mathrm{e}^\theta - 1). \tag{10}$$

According to (5), (9) and (10), we have

$$\psi_C^*(x) = \begin{cases} x\log(x/\Lambda) + \Lambda - x, & x \ge 0, \\ \infty, & o.w., \end{cases}$$

$$\psi_A^*(x) = \begin{cases} x\log(x/\lambda) + \lambda - x, & x \ge 0, \\ \infty, & o.w.. \end{cases}$$

To derive $\psi^*(x)$, suppose that $\pi(y) = \psi_C^*(y) + \psi_A^*(y - x)$ when x is given. When $y > x \ge 0$, $\frac{d^2\pi}{dy^2} > 0$. Combined this condition with $\frac{d\pi}{dy} = 0$, the minimum value of $\pi(y)$ (i.e., $\psi^*(x)$) can be calculated as

$$\begin{aligned}\pi(y^*) = \lambda + \Lambda - \sqrt{4\lambda\Lambda + x^2} \\ + \frac{1}{2}\left(\sqrt{4\lambda\Lambda + x^2} - x\right)\log\left(\frac{\sqrt{4\lambda\Lambda + x^2} - x}{2\lambda}\right) \\ + \frac{1}{2}\left(\sqrt{4\lambda\Lambda + x^2} + x\right)\log\left(\frac{\sqrt{4\lambda\Lambda + x^2} + x}{2\Lambda}\right),\end{aligned} \tag{11}$$

where $y^* = \frac{1}{2}(x + \sqrt{x^2 + 4\Lambda\lambda})$.

Through substituting x in (11) with $\frac{b}{t}$, we can obtain the infimum of $t\psi^*(\frac{b}{t})$ when $t = \frac{b}{\lambda - \Lambda}$. Thus,

$$I(b) = b\log\frac{\lambda}{\Lambda}. \tag{12}$$

$I(b)$ is the rate function of $\mathbf{P}(Q > \Theta)$ descending exponentially. Hence, $I(b)$ is an index indicating the decay speed of the probability that a blockchain network becomes vulnerable. If $I(b)$ is small, we should speed up our counterattacks and vice versa.

Fig. 1 illustrates how $I(b)$ changes with b as well as the difference in block completion rates of the honest node and the malicious one, i.e., $\lambda - \Lambda$, and the block completion rate of malicious node is set to a fixed value. It can be found that the increase of either b or $\lambda - \Lambda$ leads to that of $I(b)$. This implies that no matter enhancing the robust level related parameter or strengthening the computational power of the honest node is able to speed up the failure of the forking attack. Moreover, this figure also shows that the decay rate $I(b)$ increases linearly with b while goes up logarithmically with the difference of block completion rates (reflecting the computational power difference) $\lambda - \Lambda$, which demonstrates the former is more powerful than the latter in term of speeding up the failure of attacks. Hence, an exciting conclusion can be drawn that a blockchain network can get better protection in a cheap way, i.e., setting the robust level related parameter, rather than enhancing computational power, which is obviously costly. This enlightens us to focus on the feasible setting of the robust level due to its importance. To that aim, we present the following concept.

Definition III.2 (Effective robust level)**.** *The effective robust level is the minimum difference in number of the blocks completed by a malicious node and an honest one to confirm a transaction, for making the vulnerability probability of a*

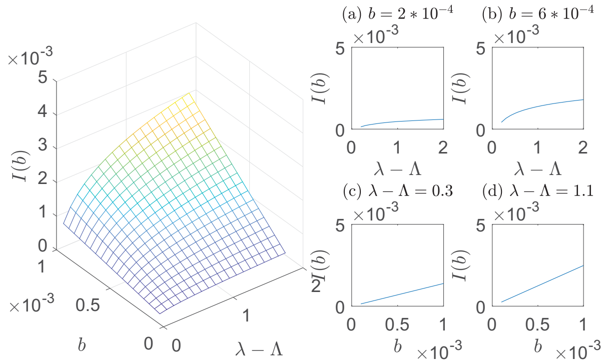

(a) $b = 2 * 10^{-4}$ (b) $b = 6 * 10^{-4}$ (c) $\lambda - \Lambda = 0.3$ (d) $\lambda - \Lambda = 1.1$

Fig. 1. Decay rate of $\mathbf{P}(Q > \Theta)$ in the special case.

blockchain network smaller than a given threshold $\epsilon \in (0, 1]$, which is called the vulnerability tolerance degree.

According to this definition, the effective robust level Θ^* of a blockchain network is

$$\Theta^*(\epsilon) = \min\{\Theta : \mathbf{P}(Q > \Theta) \leq \epsilon\},$$

which can be calculated through the following theorem.

Theorem III.2. *Given $\lambda > \Lambda > 0$ and $\epsilon \in (0, 1]$, the effective robust level of a blockchain network can be solved by*

$$\Theta^* = -\frac{\log \epsilon}{\log \frac{\lambda}{\Lambda}}.$$

Proof: In light of (3), we can find that when $l \to \infty$, $\mathbf{P}(Q > \Theta)$ can be approximated by $\mathrm{e}^{-I(b)l}$. Solving $\mathbf{P}(Q > \Theta) \leq \epsilon$, we can get $I(b) \geq -\frac{\log \epsilon}{l}$. With the expression of $I(b)$ shown in (12), we can derive the value of Θ^*. ■

Although large b has high efficiency in counterattacking intentional forks, it leads to a long-delay transaction. If we need to strengthen the robustness of a blockchain network serving for real-time transactions, it is imperative to improve the computational power of an honest node. In this case, we should answer a fundamental question: how much computational power an honest node needs to provide for fighting against forking attacks? To address the problem, we introduce the following concept.

Definition III.3 (Effective computational power)**.** *The effective computational power of an honest node is its minimum number of completed blocks during a time interval, which can make the vulnerability probability of the blockchain network smaller than the vulnerability tolerance degree $\epsilon \in (0, 1]$.*

According to the definition, the effective computational power a^* is

$$a^*(\epsilon) = \min\{\lambda : \mathbf{P}(Q > \Theta) \leq \epsilon\},$$

which can be calculated by the following theorem.

Theorem III.3. *Given $\Lambda > 0$ and $\epsilon \in (0, 1]$, the effective computational power of an honest node can be solved by*

$$a^* = \Lambda \mathrm{e}^{-\frac{\log \epsilon}{\Theta}}.$$

The proof of Theorem III.3 is similar to that of Theorem III.2. Hence, we omit its proof to avoid redundancy.

IV. Vulnerability Analysis for the General Case

In this section, we analyze the vulnerability of Blockchain in a general case where there are $M > 1$ honest nodes and $n > 1$ malicious ones. Because the consensus among the honest nodes is ultimately achieved by considering the longest blockchain to be the correct [19], the block chained in the true longest blockchain (i.e., the main chain) is the valid one. Hence, we have the following definition,

Definition IV.1 (Valid honest nodes)**.** *An honest node is valid only if it works on the main chain.*

To conduct our analysis, let time 0 be the beginning observation time, from when the main chain (denoted as $chain_0$) spreads over the whole blockchain network and the honest nodes who receive the main chain turn into the valid ones and try to append blocks to it. Denote T_1 be the earliest time when a valid honest node chained a block to the main chain successfully since time 0, implying that a new main chain (i.e., $chain_1$) is produced at T_1. Hence, the number of valid honest nodes increases with the propagation of the main chain since time 0 and regresses to 1 until T_1 because only one honest node has the information of new main chain at that time[2]. For analyzing conveniently, once the main chain is updated, time will be reset to 0, indicating the next observation time begins. Similarly, let T_i be the time when the i^{th} new main chain, namely $chain_i$, is produced and hence the number of valid honest nodes increases as the main chain spreads over the blockchain network during $(0, T_i)$ and reduces to 1 at T_i $(i = 1, 2, ...)$. Fig. 2 shows the time division in the general case.

To estimate the number of valid honest nodes at any time, we adopt a kind of compartmental model, i.e., the susceptible-infected (SI) model [22], to depict the evolution of honest nodes, who have two states: the susceptible ($\mathbb{S}$) and the infected ($\mathbb{I}$). An honest node in state $\mathbb{S}$ (i.e., the susceptible honest node) is an invalid node who works on a non-main chain while that in state $\mathbb{I}$ (i.e., the infected honest node) is a valid one. An honest node in state $\mathbb{S}$ can turn into that in state $\mathbb{I}$ if any valid honest node sends it the main chain. Let the probability of an invalid honest node receiving the main chain from any valid honest node be β and the number of valid honest nodes at time $s \in [0, T_i)$ $(i = 1, 2, ...)$ be $\chi(s)$. Then, $\chi(s)$ evolves according to the following dynamics:

$$\begin{cases} \frac{d\chi(s)}{ds} = \beta(M - \chi(s))\chi(s), \\ \chi(0) = 1. \end{cases} \tag{13}$$

On the other hand, malicious nodes collude to carry out forking attacks, working together on the chain containing their

[2] In reality, the probability that two honest nodes append the block to the main chain at the same time is very low. Therefore, we think only one honest node completes a block at T_1 for simplicity.

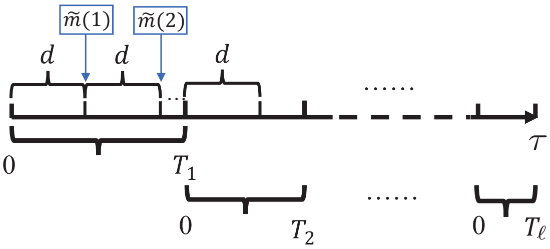


Fig. 2. Time division in the general case.

fake transactions (i.e., the fake chain). Hence, they also need to reach a consensus and append blocks to the longest fake chain. As a result, we have the following definition.

Definition IV.2 (Valid malicious nodes)**.** *A malicious node is valid only if it works on the longest fake chain.*

Since the fake chain is delivered dedicatedly among malicious nodes, its transmission has a clear purpose which does not like the delivery of the longest blockchain with randomness. Hence, the number of valid malicious nodes at any time can be roughly equivalent to the total number of malicious nodes, i.e., n, if the malicious nodes are geographically close, and otherwise, it can be calculated through the SI model similar to the solution for the number of the valid honest nodes. In this paper, we assume the number of valid malicious nodes at any time to be n for simplicity.

To conduct our analysis, as shown in Fig. 2, we further divide our analysis time $[0, \mathbb{T})$ into multiple equal-sized time intervals with each being $d < T_i$ $(i = 1, 2, ...)$ and $\frac{\mathbb{T}}{d} = \tau \in Z^+$, where $\mathbb{T} = \sum_{i=1}^{\ell} T_i$ and ℓ is a sufficiently large number. Within each d, the average numbers of blocks completed by an individual honest node and malicious one are respectively λ and Λ. Because nodes work independently, based on the above analysis, the average numbers of blocks completed by all valid malicious nodes and honest ones within an interval are respectively $n\Lambda$ and $m\lambda$, where m is the average number of valid honest nodes within each interval. m can be estimated by $\frac{\sum_{j=1}^{\ell}\sum_{k=1}^{\lfloor T_j/d \rfloor}\widetilde{m}(k)}{\sum_{j=1}^{\ell}\lfloor T_j/d \rfloor}$, where $\widetilde{m}(k)$ is the average number of valid honest nodes at the interval k, which can be calculated according to the SI model we introduced above. $m\lambda > n\Lambda$ since the joint computational power of all honest nodes is usually larger than that of all malicious nodes in reality[3].

Similar to the analysis of the special case, we need to study the transient states of the computational powers provided by all malicious nodes and honest ones. Hence, let $\mathcal{C}_\tau$ and $\mathcal{A}_\tau$ respectively be the total numbers of blocks completed by all valid malicious nodes and honest nodes during time period $[0, \mathbb{T})$, i.e., $\mathcal{C}_\tau = \widetilde{c}_1 ... + \widetilde{c}_k ... + \widetilde{c}_\tau$ and $\mathcal{A}_\tau = \widetilde{a}_1 ... + \widetilde{a}_k ... + \widetilde{a}_\tau$, where $\widetilde{c}_k$ and $\widetilde{a}_k$ $(k = 1, 2, \cdots, \tau)$ are respectively the numbers of blocks completed by all valid malicious nodes and honest nodes during the interval k. $\widetilde{c}_k = n\widehat{c}_k$ and $\widetilde{a}_k = \widetilde{m}(k)\widehat{a}_k$, where $\widehat{c}_k$ and $\widehat{a}_k$ are respectively the average numbers of blocks completed by an individual malicious node and honest one at the interval k. Thus, the difference $(\mathcal{W}_\tau)$ in numbers of blocks completed by all valid malicious nodes and all valid honest ones at the end of the τ^{th} interval can be calculated as

$$\mathcal{W}_\tau = \mathcal{C}_\tau - \mathcal{A}_\tau,$$

which reflects the joint computational power difference (i.e., the joint strength difference) between all valid malicious node and all valid honest ones and $\mathcal{W}_0 = 0$. The cumulative generating function of $\mathcal{W}_\tau$ is,

$$\widetilde{\psi}(\theta) = \lim_{\tau\to\infty}\frac{1}{\tau}\log \mathbf{E}[\mathrm{e}^{\theta\mathcal{W}_\tau}].$$

To analyze the vulnerability of the blockchain network with multiple honest nodes and malicious ones, we study the variable $\widetilde{Q}$ which is the maximum value of $\mathcal{W}_\tau$, i.e.,

$$\widetilde{Q} = \sup_{\tau\geq 0}\mathcal{W}_\tau.$$

$\widetilde{Q}$ reflects the joint largest computational power difference between all valid malicious nodes and all valid honest ones. Given a robust level Θ, if $\widetilde{Q} > \Theta$, the blockchain network suffers vulnerability issue and vice versa. Hence, the probability $\mathbf{P}(\widetilde{Q} > \Theta)$ reflects the vulnerability degree of the blockchain network with multiple malicious nodes and honest ones, which provides us an important reference on when to strike back against the forking attack. Let $\Theta = lb$ and $l \to \infty$, according to Cramer's theorem, for $\mathbf{P}(\widetilde{Q} > \Theta)$, we have

$$\lim_{l\to\infty}\frac{1}{l}\log \mathbf{P}(\widetilde{Q} > lb) = -\mathcal{I}(b),$$

where

$$\mathcal{I}(b) = \inf_{\tau\geq 0}\tau\widetilde{\psi}^*(\frac{b}{\tau}). \tag{14}$$

In (14), $\widetilde{\psi}^*(\cdot)$ is the Legendre transformation of $\widetilde{\psi}(\cdot)$, which can be calculated as

$$\begin{aligned}\widetilde{\psi}^*(x) =& \sup_{\theta\in\mathbb{R}}\{\theta x - \widetilde{\psi}(\theta)\}\\ =& \sup_{\theta\in\mathbb{R}}\{\theta x - n\Lambda(\mathrm{e}^{\theta} - 1) - m\lambda(\mathrm{e}^{-\theta} - 1)\}.\end{aligned} \tag{15}$$

Due to $\frac{b}{\tau} > 0$, we only consider the case of $x > 0$ in (15). Let $g(\theta) = \theta x - n\Lambda(\mathrm{e}^{\theta} - 1) - m\lambda(\mathrm{e}^{-\theta} - 1)$. Because $g''(\theta) = -n\Lambda\mathrm{e}^{\theta} - m\lambda\mathrm{e}^{-\theta} < 0$, θ^* that makes $g'(\theta) = 0$ can maximize $g(\theta)$. Thus, θ^* can be solved as

$$\theta^* = \log\frac{x + \sqrt{x^2 + 4mn\lambda\Lambda}}{2n\Lambda}.$$

Hence, we have

$$\begin{aligned}\widetilde{\psi}^*(x) =& x\log\frac{x + \sqrt{x^2 + 4mn\lambda\Lambda}}{2n\Lambda} - \frac{x + \sqrt{x^2 + 4mn\lambda\Lambda}}{2}\\ &- \frac{2mn\Lambda\lambda}{x + \sqrt{x^2 + 4mn\Lambda\lambda}} + m\lambda + n\Lambda.\end{aligned} \tag{16}$$

[3] In the general case, the block completion rates of an individual malicious node and an honest one do not need to satisfy $\Lambda < \lambda$. We do not use other notations to indicate the block completion rates of an individual malicious node and an honest one in the general case since they have the same meanings as in the special case.

Let $x = \frac{b}{\tau}$ in (16), we can obtain the infimum of $\tau\widetilde{\psi}^*(\frac{b}{\tau})$ at $\tau = \frac{b}{m\lambda - n\Lambda}$. Thus, we have

$$\mathcal{I}(b) = b \log \frac{m\lambda}{n\Lambda}.$$

$\mathcal{I}(b)$ is the rate function of $\mathbf{P}(\widetilde{Q} > \Theta)$ which falls off exponentially with Θ. Hence, $\mathcal{I}(b)$ indicates the decay speed of the probability that a blockchain network suffers vulnerability issue. If $\mathcal{I}(b)$ is small, we should accelerate our counterattacks and vice versa.

Fig. 3 illustrates the impacts of b as well as the difference in joint block completion rates of all valid honest nodes and all valid malicious ones, i.e., $m\lambda - n\Lambda$, on the decay rate function $\mathcal{I}(b)$, and the joint block completion rate of valid malicious nodes is set to a fixed value. The trend similar to that in the special case can be observed. To be specific, b has more power than the joint block completion rate difference (reflecting the joint strength difference) in term of speeding up the failure of forking attack. This means we can employ a simple and cheap way, i.e., increasing b, to better guard the security of a blockchain network consisted of multiple honest nodes and malicious ones.

To find an appropriate robust level, we also consider the concept of the effective robust level Θ^* in this general case, which denotes the minimum difference in number of the blocks completed by all valid malicious nodes and honest ones to confirm a transaction within a unit time, for making the vulnerability probability of a blockchain network smaller than the vulnerability tolerance degree $\epsilon \in (0, 1]$. To be specific, for any $\epsilon \in (0, 1]$, Θ^* is

$$\Theta^*(\epsilon) = \min\{\Theta : \mathbf{P}(\widetilde{Q} > \Theta) \leq \epsilon\}.$$

Θ^* can be calculated by the following theorem, whose proof is similar to that of Theorem III.2 and thus we omit it to avoid redundancy.

Theorem IV.1. *Given $m\lambda > n\Lambda > 0$, $\epsilon \in (0, 1]$ and $l \gg 1$, the effective robust level of the blockchain network can be solved by*

$$\Theta^* = -\frac{\log \epsilon}{\log \frac{m\lambda}{n\Lambda}}.$$

If we want to boost the security of a blockchain network through strengthening the computational power of honest nodes so as to avoid the high transaction delays incurred by a large b, the effective computational power also needs to be paid attention in the general case. Specifically, the effective computational power $\widetilde{a}^*$ of all valid honest nodes is the minimum number of blocks completed by them during any time interval to make the vulnerability probability of the blockchain network smaller than the vulnerability tolerance degree $\epsilon \in (0, 1]$. In other words, $\widetilde{a}^*$ can be written as

$$\widetilde{a}^*(\epsilon) = \min\{m\lambda : \mathbf{P}(\widetilde{Q} > \Theta) \leq \epsilon\}.$$

Then we can calculate the effective computational power $\widetilde{a}^*$ of all valid honest nodes in the general case as follow.

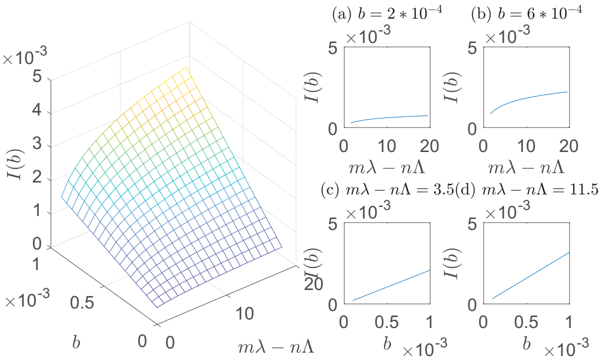


Fig. 3. Decay rate of $\mathbf{P}(\widetilde{Q} > \Theta)$ in the general case.

Theorem IV.2. *Given $n > 0$, $\Lambda > 0$ and $\epsilon \in (0, 1]$, the effective computational power of all honest nodes is*

$$\widetilde{a}^* = n\Lambda e^{-\frac{\log \epsilon}{\Theta}}.$$

The proof of this theorem is omitted due to the same reason in Theorem III.3.

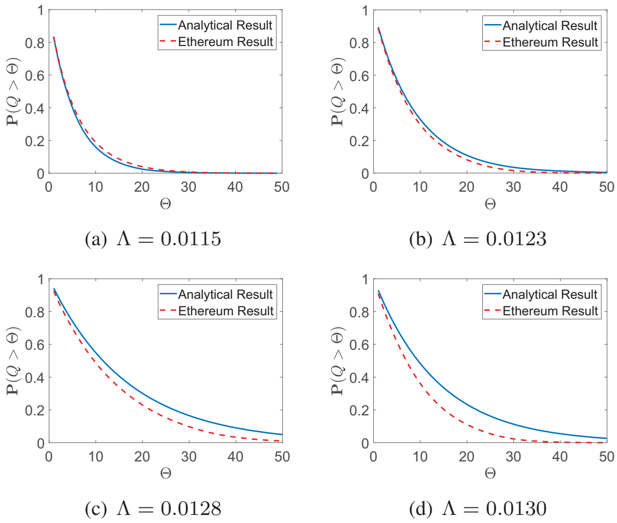
V. Experimental Evaluation

In this section, we evaluate our vulnerability analysis of Blockchain on a large-scale cloud platform configured with the Ethereum protocol. We have created multiple virtual machines (VMs) on our cloud platform with each running Ubuntu 16.04, having Intel Xeon E7-4830 v2 2.20 GHz CPU and 2 GB RAM. Each VM is mounted a v1.8.12 Ethereum client [23] to act as a miner. The difficulty of PoW in the *geth* implementation is adopted to represent the block canonical difficulty in the Ethereum project yellow paper [2], which is set to different values in our experiments to verify the effectiveness of our proposed vulnerability analysis.

A. Evaluation of Vulnerability Analysis for the Special Case

We firstly evaluate our proposed vulnerability analysis in the special case where there are only one honest node and one malicious node. To calculate $\mathbf{P}(Q > \Theta)$ for a given $b \in (0, 5 \times 10^{-3}]$, we set $l = 10000$, which is large enough satisfying the requirement of (3). Then, we can calculate $C_t - A_t$, i.e., the difference in total numbers of blocks completed by the malicious node and the honest one at the end of the t^{th} interval. By this means, we can get the value of W_t, as well as $Q = \sup_{t \geq 0} W_t$. Each experiment is repeated for 1,000,000 times to obtain the probability distribution of $Q > \Theta$, which reflects the vulnerability degree of the blockchain network.

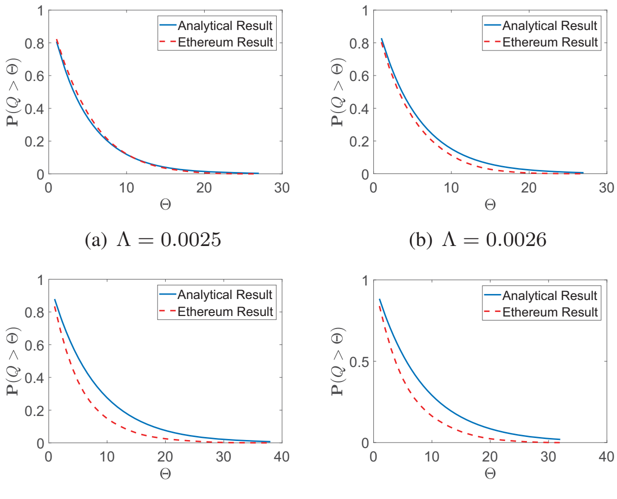
Figs. 4 and 5 report the experimental and analytical results on $\mathbf{P}(Q > \Theta)$ in the special case. It is worth noting that since Q denotes the maximum difference in numbers of blocks completed by the malicious node and the honest one, we employ $\Theta \in \mathbb{Z}^+$ as the independent variable in our experiments. In particular, we firstly set the difficulty of PoW as 5,000,000, and the block completion rate of the honest node as $\lambda = 0.0138$ while that of the malicious node Λ are $0.0115, 0.0123, 0.0128$ and 0.0130, respectively.



(a) $\Lambda = 0.0115$ (b) $\Lambda = 0.0123$

(c) $\Lambda = 0.0128$ (d) $\Lambda = 0.0130$

Fig. 4. Special case with $\lambda = 0.0138$ when the difficulty of PoW is 5,000,000.



(a) $\Lambda = 0.0025$ (b) $\Lambda = 0.0026$

(c) $\Lambda = 0.0027$ (d) $\Lambda = 0.0028$

Fig. 5. Special case with $\lambda = 0.0032$ when the difficulty of PoW is 10,000,000.

The experimental and analytical results under this parameter setting are shown in Fig. 4. Then we modify the difficulty of PoW to 10,000,000, and the block completion rate of the honest node to $\lambda = 0.0032$, while that of the malicious node $\Lambda = 0.0025, 0.0026, 0.0027$, and 0.0028, respectively. Fig. 5 illustrates the experimental and analytical results under this parameter setting.

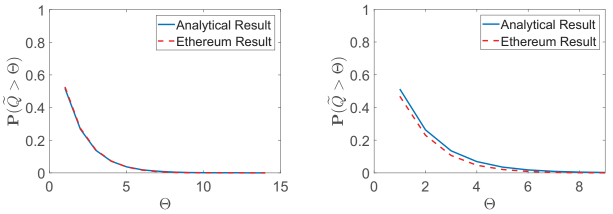
As shown in Figs. 4 and 5, our experimental and analytical results match well under different block completion rates of the malicious node and different difficulties of PoW. In addition, $\mathbf{P}(Q > \Theta)$ slumps exponentially with the increase of Θ. Because l is set to a fixed value, we can get the conclusion that when b goes up, it becomes harder for the malicious node to make the blockchain network vulnerable.

B. Evaluation of Vulnerability Analysis for the General Case

Next, we evaluate our proposed vulnerability analysis for the general case with multiple honest nodes and malicious ones. Similarly, given $b \in (0, 5 \times 10^{-3}]$ and the numbers of valid malicious nodes and honest ones n and m, for calculating $\mathbf{P}(\widetilde{Q} > \Theta)$, we set $l = 10000$ to calculate $\mathcal{C}_\tau - \mathcal{A}_\tau$ and further get $\widetilde{Q}$. Each experiment is repeated for 1,000,000 times to obtain a more accurate probability distribution.

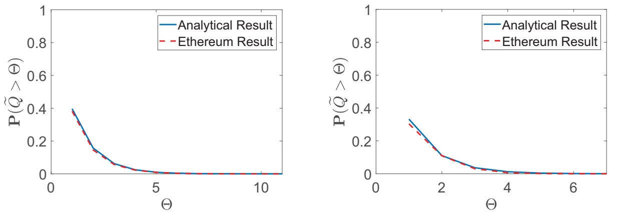
In Figs. 6, 7 and 8, the experimental and analytical results on $\mathbf{P}(\widetilde{Q} > \Theta)$ are plotted in the general case with multiple honest nodes and malicious ones, i.e., 6 honest nodes vs. 3 malicious nodes, 9 honest nodes vs. 3 malicious nodes and 9 honest nodes vs. 6 malicious nodes. In Fig. 6, the total block completion rates of 6 honest nodes are $m\lambda = 0.0790$ and 0.0355 while those of 3 malicious nodes are $n\Lambda = 0.0422$ and 0.0182, where difficulties of PoW are set as 5,000,000 and 10,000,000, respectively (PoW difficulties are following the same settings in Figs. 7 and 8); in Fig. 7, total block completion rates of 9 honest nodes are changed to $m\lambda = 0.1065$ and 0.0548 while those of 3 malicious nodes are still $n\Lambda = 0.0422$ and 0.0182; and in Fig. 8, the total block completion rates of 9 honest nodes keep being $m\lambda = 0.1065$ and 0.0548 while those of 6 malicious nodes are $n\Lambda = 0.0790$ and 0.0355.

Similar to the special case, our experimental and analytical results match well. In addition, according to these figures, we can observe that $\mathbf{P}(\widetilde{Q} > \Theta)$ sharply decreases with the increase of Θ in both experimental and analytical results. Comparing Fig. 6 with 6 honest nodes and Fig. 7 with 9 honest nodes given the same number of malicious nodes, we can find that the value of $\mathbf{P}(\widetilde{Q} > \Theta)$ is generally decreased when the number of honest nodes increases. Taking the results of $\Theta = 1$ in Fig. 6(a) and Fig. 7(a) as an example, $\mathbf{P}(\widetilde{Q} > 1)$ diminishes from 51.7% to 39.6%, which indicates that more honest nodes can better resist the attack from a certain number of malicious nodes. Similarly, through comparing Figs. 7(a) with 8(a), one can find that the values of $\mathbf{P}(\widetilde{Q} > \Theta)$ under different b increase obviously with the increment of malicious nodes, when the number of honest nodes is kept constant. For instance, given the target difference between the numbers of blocks mined by the malicious nodes and the honest ones $\widetilde{Q} > 1$, the value of $\mathbf{P}(\widetilde{Q} > 1)$ moves up from 39.6% to 74.2% when the number of malicious nodes increases from 3 to 6. It implies that more malicious nodes can enhance the successful attack rate given the fixed number of honest nodes.



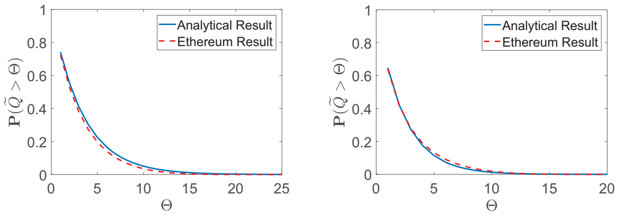
(a) Difficulty of PoW$= 5,000,000$ (b) Difficulty of PoW$= 10,000,000$

Fig. 6. General case with 6 honest nodes and 3 malicious nodes.



(a) Difficulty of PoW$= 5,000,000$ (b) Difficulty of PoW$= 10,000,000$

Fig. 7. General case with 9 honest nodes and 3 malicious nodes.



(a) Difficulty of PoW$= 5,000,000$ (b) Difficulty of PoW$= 10,000,000$

Fig. 8. General case with 9 honest nodes and 6 malicious nodes.

VI. Conclusion

In this paper, we take advantage of the large deviation theory to study the vulnerability of blockchain networks. To be specific, the probability and its decay speed of a blockchain network's being vulnerable are deduced. In addition, we propose the concepts of the effective robust level and the effective computational power. The solutions of the effective robust level and the effective computational power can provide a theoretical basis to design reasonable schemes for invigorating a blockchain network from managerial as well as technical levels. Through our analysis, we find: 1) the probability of the largest strength difference between malicious and honest node(s) being greater than a threshold decays exponentially with the increment of the robust level related parameter; 2) setting the robust level related parameter has more power than enhancing the computational power in speeding up the failure of attacks. This finding is valuable since it renders an opportunity to improve the robustness of a blockchain network in a cost-efficient way. Extensive experiments carried out on a large-scale cloud platform running the Ethereum protocol show the experimental and analytical results match well, verifying the effectiveness of our analysis.

Acknowledgement

This work has been supported by the National Natural Science Foundation of China (No. 61772080).

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, 2014.

[3] *Marketandmarkets*. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/blockchain-technology-market-90100890.html

[4] S. Spiekermann, A. Acquisti, R. Böhme, and K.-L. Hui, "The challenges of personal data markets and privacy," *Electronic Markets*, vol. 25, no. 2, pp. 161–167, 2015.

[5] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 906–917.

[6] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Čapkun, "Misbehavior in bitcoin: A study of double-spending and accountability," *ACM Transactions on Information and System Security (TISSEC)*, vol. 18(1), no. 2, pp. 1–32, 2015.

[7] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 692–705.

[8] Z. Liu, H. Zhao, W. Chen, X. Cao, H. Peng, J. Yang, T. Yang, and P. Lin, "Double-spending detection for fast bitcoin payment based on artificial immune," in *National Conference of Theoretical Computer Science*. Springer, 2017, pp. 133–143.

[9] A. J. Ganesh, N. O'Connell, and D. J. Wischik, *Big queues*. Springer, 2004.

[10] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Communications of the ACM*, vol. 61, no. 7, pp. 95–102, 2018.

[11] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Financial Cryptography and Data Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 515–532.

[12] D. K. Tosh, S. Shetty, X. Liang, C. A. Kamhoua, K. A. Kwiat, and L. Njilla, "Security implications of blockchain cloud with analysis of block withholding attack," in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Press, 2017, pp. 458–467.

[13] R. Zhang and B. Preneel, "Publish or perish: A backward-compatible defense against selfish mining in bitcoin," in *Cryptographers? Track at the RSA Conference*. Springer, 2017, pp. 277–292.

[14] C. Prez-Sol, S. Delgado-Segura, G. Navarro-Arribas, and J. Herrera-Joancomart, "Double-spending prevention for bitcoin zero-confirmation transactions," Cryptology ePrint Archive, Report 2017/394, 2017, https://eprint.iacr.org/2017/394.

[15] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network." in *Proceedings of the 24th USENIX Conference on Security Symposium*, 2015, pp. 129–144.

[16] Y. Marcus, E. Heilman, and S. Goldberg, "Low-resource eclipse attacks on ethereum's peer-to-peer network," Cryptology ePrint Archive, Report 2018/236, 2018, https://eprint.iacr.org/2018/236.

[17] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *Proceedings of the 2016 IEEE European Symposium on Security and Privacy*. IEEE, 2016, pp. 305–320.

[18] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 3–16.

[19] N. Papadis, S. Borst, A. Walid, M. Grissa, and L. Tassiulas, "Stochastic models and wide-area network measurements for blockchain design and analysis," in *Proceedings of the 37th Annual IEEE International Conference on Computer Communications (INFOCOM 2018)*. IEEE, 2018, pp. 1–9.

[20] J. Göbel, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay," *Performance Evaluation*, vol. 104, pp. 23–41, 2016.

[21] S. M. Ross, *Introduction to probability models*. Academic press, 2014.

[22] T. Zhou, J.-G. Liu, W.-J. Bai, G. Chen, and B.-H. Wang, "Behaviors of susceptible-infected epidemics on scale-free networks with identical infectivity," *Physical Review E*, vol. 74, no. 5, pp. 56–109, 2006.

[23] *Geth*. [Online]. Available: https://github.com/ethereum/go-ethereum/