Double-Spending With a Sybil Attack in the Bitcoin Decentralized Network

Shijie Zhang and Jong-Hyouk Lee, *Senior Member, IEEE*

***Abstract*—A double-spend attack is one of the major security issues in most blockchain systems, but it is difficult to successfully launch unless an adversary has massive computing power. In this paper, we introduce a new attack model that combines a double-spend attack with a Sybil attack in the Bitcoin network. We present analysis results that a double-spending attacker can make a block propagation delay by conducting Sybil attacks and increase the probability of winning the mining race, thus successfully launching the double-spend attack. We develop the probability of success of this new attack in mathematics forms and analyze this attack model from the perspective of economics. We present the attacker's break-even point in various situations and demonstrate the effect of the proposed attack.**

***Index Terms*—Bitcoin, double-spending, Sybil attack.**

I. Introduction

BITCOIN is the first decentralized cryptocurrency created by Satoshi Nakamoto in 2008, which relies on a tamper-proof public ledger [1]. The core of Bitcoin is the blockchain technology. Blockchain is served as the decentralized public ledger containing techniques such as cryptography, consensus algorithm, peer-to-peer (P2P), etc., [2], [3]. Because of the security features such as the privacy and anonymity of the blockchain, many researchers have devoted themselves to studying it in recent years. However, the security of the blockchain systems themselves has become an important issue.

Double-spending is such an attack that poses a huge threat to the blockchain systems. It was a major security issue when developing cryptocurrencies operating over the Internet. In a double-spend attack, an attacker can manage to use the same coins or tokens multiple times [1], [4]–[6]. Bitcoin is the first cryptocurrency that successfully addressed the double-spending issue by adopting Proof-of-Work (PoW) that nodes called miners compete to get an ability to create a block to be added in the main chain [1].

In general, the implementation of a double-spend attack requires the attacker to hold lots of computing power [1]. In this paper, we focus on how attackers with less computing power can increase the probability of a double-spending attack by combining a Sybil attack [7] in the Bitcoin network. Other studies have showed that other types of attacks can help increase the probability of a successful double-spend attack. For instance, Bissias *et al.* indicated that an Eclipse attack can increase the effect of double-spend attack [8], [9]. Eyal's findings suggest that colluding miners can also successfully launch double-spend attacks with selfish mining [10]. Gervais *et al.* subsumed the combination of network-level attacks and double-spend attacks [11]. However, unlike previous studies, our attack model combines double-spending with a Sybil attack to increase a propagation delay of correct block information over the blockchain network. Our new contributions presented in this paper are as follows:

1) In this paper, we design a new combined attack model. We increase the block propagation delay by introducing a Sybil attack to influence the communication protocol (i.e., gossip protocol) between Bitcoin nodes.
2) We study the probability of a successful double-spend attack in the context of a Sybil attack. Our studies obtain a good result that one attacker with only a 32% share of the computing power in the Bitcoin network can successfully double-spend. Note that the threshold of a 25% share of the computing power proposed by Eyal [10] is the one that the attacker can profit from selfish mining, rather than the threshold of a successful double-spend attack.
3) We also contribute an economic evaluation of this attack model to find the attacker's break-even point [5], [8]. By summarizing the changes in break-even points, we can clearly find that the price an attacker pays for the attack is tiny in some cases and the proposed attack can bring huge profits to the attacker.

The rest of this paper is organized as follows. We present relevant attacks and related work in Section II. The way of information dissemination in the Bitcoin network is described in Section III. The proposed attack is presented in Section IV. The probability of a successful attack is developed in Section V. In Section VI, we present the economic evaluation of the proposed attack. Section VII concludes this paper.

Manuscript received February 27, 2019; revised May 1, 2019; accepted May 23, 2019. Date of publication June 7, 2019; date of current version October 3, 2019. This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning, South Korea (NRF-2017R1A1A1A05001405). Paper no. TII-19-0638. *(Corresponding author: Jong-Hyouk Lee.)*

The authors are with the Department of Software, Protocol Engineering Laboratory, Sangmyung University, Cheonan 31066, South Korea (e-mail: zhangshijie@pel.smuc.ac.kr; jonghyouk@pel.smuc.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2019.2921566

II. Relevant Attacks and Related Work

A. Double-Spend Attack

A double-spend attack is a potential threat to cryptocurrencies. Karame *et al.* briefly introduced the process of a double-spend attack [6]. For instance, in Bitcoin, an attacker creates a transaction TX_A that is released from an unspent transaction output (UTXO) to a merchant. TX_A will be logged in the main chain. Honest miners then start mining. Before TX_A is added into the new block in the main chain, the attacker secretly mines his own branch that follows the currently latest block. The attacker then creates another transaction TX_B which is in conflict with TX_A from the same UTXO. On the contrary, TX_B included in the attacker's branch moves the funds to the attacker's second address. As soon as the merchant receives enough block confirmations for TX_A, the merchant sends the goods to this attacker. The attacker keeps on mining and once he makes his chain longer than the main chain, the attacker broadcasts his own branch to the network. Due to the longest chain rule [1], other miners finally agree on this attacker's branch. TX_A is thus replaced by TX_B. Accordingly, the attacker gets the goods and funds at the same time.

B. Sybil Attack

A Sybil attack was named after the subject of the book Sybil and suggested in 2002 by Brian Zill at Microsoft research [7]. Douceur first introduced a specific Sybil attack process [7]. One hostile peer can conduct a Sybil attack by creating lots of fake identities to defraud the system to break its trust and redundancy mechanism. In 2003, Karlof and Wagner found that the Sybil attack was a potential threat to the wireless sensor network which is a distributed network [12]. Since then, the Sybil attack was proved by many researchers as a huge threat for P2P network systems. Bissias *et al.* found that Sybil attacks can also harm Bitcoin systems, and they proposed a mixed solution named Xim [13] to prevent this attack. Eyal *et al.* found that a Sybil attack can also influence Bitcoin mining [10]. In [14], BitFury Group concluded that Sybil attacks are likely to succeed in both Proof-of-Stake and Proof-of-Work (PoW) blockchain systems.

C. Long Delay Attack

Some studies have shown the impact of block propagation delay on the robustness of the Bitcoin blockchain system. Decker *et al.* demonstrated that the information propagation delay will cause the blockchain forks [15]. In [16], Kiayias *et al.* found the relationship between the transaction processing speed and the security of blockchain systems and indicated that the attackers with less than 50% share of the computing power in the whole network can cause damage to the systems in case the block propagation time is large. In [17], Garay *et al.* proposed the Bitcoin backbone protocol and analyzed this protocol in the bounded-delay model. Pass *et al.* [18] proved that the block propagation delay that makes the Nakamoto protocol meet *consistency* and *chain quality* has an upper bound. Moreover, they also proposed the thought of the long delay attack.

D. Limitations of Prior Work

In [18], Pass *et al.* mentioned that *consistency* and *chain quality* of the blockchain can be destroyed by increasing the delay of block propagation. However, they did not elaborate on how to achieve large block propagation delays and how much a double-spend attack can benefit from a long delay attack.

Prior research [1], [4], and [5] have studied the probability of a successful double-spend attack under the assumption that the Bitcoin network is fully synchronous. In fact, the Bitcoin network is asynchronous or partial synchronous, and the propagation of blocks has a latency. In that case, the blockchain forks caused by the block propagation latency may occur [15]. Consequently, we take the blockchain forks into account when calculating the probability of success of the proposed attack.

A number of papers discussed how to calculate the probability of a successful double-spend attack. The probability model was first mentioned in the Bitcoin white paper written by Nakamoto [1]. He presented a preliminary idea about the probability model, which was proved not to be very accurate later. Nakamoto's mistake was that he considered the time interval of mining each block to be fixed, hence the attacker's random process that creates a number of blocks within a certain period of time is a Poisson distribution [1]. Other researchers such as Rosenfeld did not regard the time of mining one block as the average time. Conversely, he inferred that the total time spent for mining blocks is a gamma distribution and the attacker's random process was proved to be a negative binomial distribution [5]. Grunspan similarly proposed a specific proof process to show that the random process of attacker mining is consistent with the negative binomial distribution [4]. However, the probability model of the prior work is not directly suitable for our proposed attack. Note that the probability model used in this paper is modified based on Rosnefeld's and Crunspan's probability models.

III. Information Dissemination in the Bitcoin Network

Bitcoin adopts the gossip protocol to disseminate the transaction and block information [19]. The gossip protocol was proposed by Demers *et al.* in 1987 [20] and was first used in distributed systems for information synchronization. The gossip protocol spreads information to the entire network in a rumored manner like its name, with the time complexity $O(\log N)$ (N is the total number of nodes) [20], [21]. Because of the good fault tolerance of the gossip protocol, the crash or restart of some Bitcoin nodes does not affect the dissemination of the transaction and block information [22]. Accordingly, within a certain range of information propagation delay, Bitcoin guarantees the eventual consistency that all the nodes will eventually converge to the same chain.

The communication method between Bitcoin nodes is based on *push–pull* in the gossip protocol [20], [21]. Each node that receives the new block information is in an "infected" state. They first check the validity of the new block, and then send the *inv* messages to the connected neighbor nodes to inform them that the new block is ready [15]. The following *getdata* messages will be sent by the neighbor nodes [15]. The neighbor nodes

request the *inv* messages senders to propagate the new block information. After the *inv* messages senders receive the *getdata* messages, they send the block information to their neighbor nodes. Such a complete block request and send communication cycle contains three information exchanges. In our proposed attack model, a Sybil attack is used to influence the rule of communication between Bitcoin nodes. The specific changes on the rule of communication will be presented in the next section.

IV. Proposed Double-Spending With a Sybil Attack

In this section, we describe the proposed attack: double-spending with a Sybil attack. We explain how this attack works from the perspective of the attack target and attack process. Moreover, we present our hypothetical attack model.

A. Attack Target

There is no doubt that the ultimate goal of the attacker is to succeed in the double-spend attack. The attacker hopes to increase a propagation delay of correct block information over the Bitcoin network by performing a Sybil attack. In our hypothetical attack environment, the Bitcoin network is not fully synchronous. Once the new block is mined and broadcast to the whole network, some honest nodes will receive the new block information and add the latest block to the longest chain (i.e., main chain). Because of the existence of the block propagation delay, the remaining honest nodes may not receive the new block information, and they will wait for a while. We assume that the upper limit time of the block propagation round is Δ and waiting for a block information takes d. If d exceeds Δ, these nodes will start a new round of consensus (i.e., mining competition), in this case, the blockchain forks will occur. The attacker's goal is to maximize the block propagation delay so that it exceeds Δ in each round of the block propagation. In this way, not only can the growth of the main chain be slowed down but the blockchain forks can also cause the waste of honest nodes' computing power [15]. Therefore, the probability that the length of the attacker's chain exceeds the main chain will increase a lot compared to the case without a Sybil attack, which provides many advantages for the attacker to achieve his ultimate target.

B. Attack Process

First, the attacker fills the Bitcoin network with multiple fake nodes having fake identities with zero computing power. We call these fake nodes Sybil nodes. With these Sybil nodes with fake identities, the attacker launches a double-spend attack. The attacker initiates a transaction TX_0 with the value of v to a merchant. Then, TX_0 is disseminated to other nodes. After TX_0 is verified, the honest nodes put TX_0 in their respective memory pools and compete for the right of creating the next block. Before the valid block including TX_0 is added to the main chain, the attacker starts secretly creating a private chain forked from the latest block, and one block of his chain contains another transaction TX_1 conflicting with TX_0 that moves some money to himself.

Algorithm 1: Sybil Attack.

```
1: 𝒮 ← Sybil nodes
2: ℋ ← Honest nodes
3: if 𝒮 receive a new block then
4:     𝒮 send inv messages to ℋ
5:     ℋ receive inv messages
6:     if ℋ respond to 𝒮 with getdata messages then
7:         𝒮 do not send the new block to ℋ
8:         while d ≤ Δ do
9:             ℋ try to send getdata messages to 𝒮
               repeatedly
10:            if ℋ get the new block then
11:                break
12:            end if
13:            d increases
14:        end while
15:        ℋ start the next block mining
16:    end if
17: end if
```

Afterward, the private chain carries out a mining race with the main chain. The attacker exploits his Sybil nodes to launch a Sybil attack to delay the growth rate of the main chain. Note that these Sybil nodes just influence the pace of the blockchain consensus instead of participating in block mining. Algorithm 1 with the time complexity $O(\Delta)$ describes the process of Sybil attack during a round of block propagation. The rule of the block propagation in the attack follows the gossip protocol described in Section III. Once Sybil nodes receive a newly mined block, each fake identity of these Sybil nodes sends an *inv* message to every connected honest node. The honest nodes receiving the *inv* messages send the *getdata* messages to these Sybil nodes. The trick here is that these Sybil nodes do not send the new block information to these honest nodes, thus the part of the block propagation is "frozen". Meanwhile the honest nodes that have not received the block information can continuously try to send the *getdata* messages to the nodes who send them *inv* messages. However, since each Sybil node owns a number of fake identities, most *inv* messages of the honest nodes come from these fake identities, thus these honest nodes are still likely not to receive the new block information in Δ. As long as d exceeds Δ, these honest nodes have to stop waiting and start the next round of block mining competition. On the flip side, the nodes receiving the new block will check the validity of the new block and then extend the longest chain by one. Accordingly, at this block height, there is a divergence between the honest nodes who hold different blockchain ledgers, i.e., blockchain forks, which can waste the computing power of the honest nodes. Although there will be some forks in the blockchain under the influence of Sybil nodes, the attacker still aims to catch up the longest chain. Since the growth rate of the longest chain is delayed, the attacker can accumulate the advantage of delay to make his private chain catch up the longest chain in fewer rounds than the case without a Sybil attack, and finally TX_0 is erased, making TX_1 valid.

Fig. 1(a)–(c) describe three different states of the blockchain during an attack. We only depict the main chain and the private

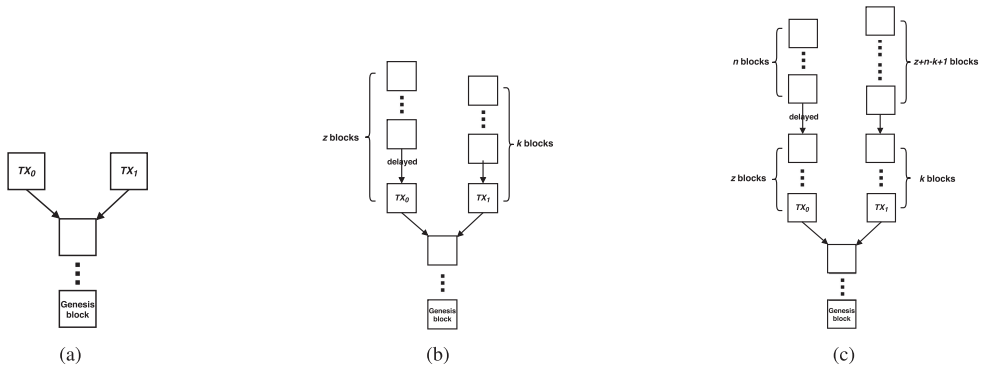

Fig. 1. Three states of the blockchain that may occur from the start to the success of the attack. (a) Initial state. (b) Second state. (c) Third state.

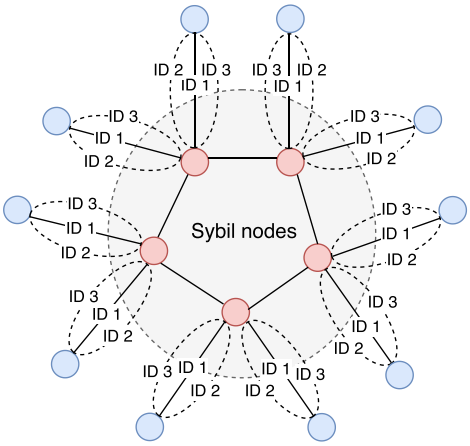

Fig. 2. Sybil attack model.

chain instead of adding other forks caused by the Sybil attack. The first state is that the attacker starts mining on his own branch after he sends TX_0 to the merchant. The second state is that the merchant has waited for z block confirmations of TX_0, while the attacker has already mined k blocks. The third state occurs when k is less than z. If k is greater than z, there is no third state and the attacker directly succeeds in the proposed attack at the second state. At the third state, we suppose that the honest miners have mined n blocks after TX_0 was confirmed. The Sybil attack continues slowing down the propagation of blocks in the main chain until the attacker forges a private chain that is one longer than the length of the main chain. At that time, the attacker will broadcast his own branch throughout the network, and soon the private chain will get the consensus of most miners since it has a longer length.

C. Attack Model

In terms of the Sybil attack, we assume that the attacker deploys a number of Sybil nodes $\mathcal{S}$ as shown in Fig. 2. Each $\mathcal{S}$ (represented in red) is connected to two honest nodes (represented in blue). In order to maximize the delay of block propagation to make d exceed Δ, each $\mathcal{S}$ exploits a large number of fake identities to defraud the connected honest nodes to ensure multiple *getdata* messages of these honest nodes are sent to $\mathcal{S}$, thus these honest nodes cannot timely obtain the block information from other honest nodes in Δ. As shown in Fig. 2, each $\mathcal{S}$ communicates with the connected honest nodes via his three different identities. Note that the actual number of fake identities per Sybil node is much larger, and it does not affect the calculation in the following sections.

On the flip side, since $\mathcal{S}$ have zero computing power, only the attacker's own node participates in the creation of the private chain. We assume the attacker's computing power constitutes q of the total computing power in the network. Therefore, the honest miners' fraction of the total computing power is $p = 1 - q$. We assume that $q < p$ and each honest node has identical computing power. We also assume that the computing power of the attacker node and the honest nodes is positively correlated with their mining speed. During the proposed attack, the attacker cannot change his computing power or break the rule of PoW to speed up his mining speed. Instead, the attacker just leverages Sybil nodes to influence the block propagation process.

The environmental parameters of this attack model are presented in Table I.

V. Probability of Successful Attack

In this section, we study the probability of a successful attack for the proposed attack model. We use some of the parameters shown in Table I to develop mathematical expressions inspired from the probability models of Nakamoto's work [1], Grunspan's work [4], and Rosenfeld's work [5].

First we consider the following two cases that an attacker succeeds in double-spending. One is that the attacker, with mining speed α, mines blocks greater than or equal z blocks mined by honest miners within S'_z. In this case, the attacker has caught up the main chain. Another case is that an attacker has mined k blocks within S'_z and he also has a chance to catch up the gap $z - k$ with probability P_{z-k}. By summarizing these two cases, we can get the probability of the success of the proposed attack P [4]

$$P = Pr[X_z \geq z] + \sum_{k=0}^{z-1} Pr[X_z = k] P_{z-k}$$

$$= 1 - \sum_{k=0}^{z-1} Pr[X_z = k](1 - P_{z-k}). \quad (1)$$

TABLE I
PARAMETER DEFINITIONS AND VALUES OF THE ATTACK MODEL

Parameter	Description	Value or Range
$\mathcal{S}$	Sybil nodes	
$\mathcal{H}_S$	Honest nodes connected by $\mathcal{S}$	
$\mathcal{H}_R$	Remaining honest nodes	
N	Total number of nodes in the Bitcoin network	0-20000
μ	Fraction of the number of $\mathcal{S}$ to N	0-1
q	Fraction of the attacker's computing power	0-1
p	Fraction of the honest miners' computing power	0-1
Δ	Upper limit time of each block propagation	0-600 s
d	Time to wait for a block information	0-600 s
τ_0	Average time to create one block set by Bitcoin	600 s
z	Number of consensus rounds consisting of block mining and propagation	>0
X_z	Number of blocks mined by the attacker in z rounds	>0
T_z	Time to mine a block in zth round	>0
S_z	Sum of time to mine blocks in z rounds	>0
${T_z}'$	Time to add one block to the main chain in zth round	>0
${S_z}'$	Time it takes the merchant to wait for z block confirmations	>0
α'	Mining speed of the honest miners	0-1/600
α	Mining speed of the attacker	0-1/600
β	Growth rate of the main chain	>0

What we should pay attention to here is the target that the attacker wants to pursue. No matter how many chains are forked in the network, according to the longest chain principle in PoW, the main chain is the longest chain. In other words, the main chain is the chain with the most cumulative computing power. Now we need to figure out the probability $Pr[X_z = k]$ that the attacker creates k blocks within S'_z. The attacker's mining process is different from the case where a Sybil attack is not used. Under the influence of the block propagation delay, the time T'_z spent for adding one block to the main chain is sure to increase. Similarly, β is certain to decrease. We previously assumed that Sybil nodes try to delay the block propagation to make d exceed Δ in each consensus round. Hence, the actual time spent for one block added to the main chain in each consensus round is: $T_z + \Delta$. So we can get the expected value of T'_z

$$E[T'_z] = E[T_z] + \Delta = \frac{1}{\alpha'} + \Delta, \tag{2}$$

where α' is related to the honest miners' computing power, while α is related to the attacker's computing power. As mentioned before, the proportion of one miner's computing power to the total computing power is equal to his fraction of the total mining speed in the whole network [5]. Then, we have

$$p = \frac{\alpha'}{\alpha + \alpha'}\ ,\ q = \frac{\alpha}{\alpha + \alpha'}. \tag{3}$$

Recall that all of the nodes except Sybil nodes will participate in block mining. Therefore, the total mining speed in the network equals the sum of the mining speed of the attacker and honest miners [4]. Then, we have

$$\alpha + \alpha' = \frac{1}{\tau_0}. \tag{4}$$

By combining (3) and (4), we have

$$\alpha' = \frac{p}{\tau_0}\ ,\ \alpha = \frac{q}{\tau_0}. \tag{5}$$

However, because of the existence of blockchain forks, the computing power of some honest nodes will be wasted. In fact, $E[T_z]$ is not equal to $1/\alpha'$, and the hash rate used for the calculation of PoW in the main chain will be slower. Now we need to find the computing power that effectively contributes to the main chain. According to the deployment of Sybil nodes shown in Fig. 2, each $\mathcal{S}$ is connected to two honest nodes. There are two cases we need to consider:

1) The new block is mined by $\mathcal{H}_R$. Ideally, there may be up to $2\mu N$ honest nodes that will not receive the block information in a consensus round (sometimes $\mathcal{H}_S$ may inevitably receive the block information sent by $\mathcal{H}_R$, thus the actual number is less than $2\mu N$), so they will start the next block mining and mine on the forked chain. If $\mathcal{H}_R$ have more computing power, about $2\mu N$ honest nodes' computing power will be wasted.
2) The new block is mined by $\mathcal{H}_S$. In this case, $\mathcal{H}_R$ may not receive the block information in a consensus round, and they will mine on the forked chain. Nevertheless, if $\mathcal{H}_R$ have more computing power, the forked chain will become the main chain, nearly $2\mu N$ honest nodes' computing power will be wasted.

Combining the above two situations, we can infer that the number of honest nodes whose computing power is wasted is $2\mu N$ at most if the computing power of $\mathcal{H}_R$ is greater than or equal to $\mathcal{H}_S$. The following calculation of the probability P is carried out under the assumption that $2\mu N$ honest nodes' computing power is wasted and the computing power of $\mathcal{H}_R$ is greater than or equal to $\mathcal{H}_S$. Next, we can get the ratio δ of the nodes whose computing power is wasted to all honest nodes according to this conclusion. It follows that:

$$\delta = \frac{2\mu N}{N - \mu N} = \frac{2\mu}{1 - \mu}. \tag{6}$$

Because each honest node has identical computing power, we can get the wasted part p_{waste} and the effective part p^* in p:

$$\begin{aligned} p^* &= p - p_{\text{waste}} \\ &= p - \frac{2\mu}{1-\mu}p \\ &= p\left(\frac{1-3\mu}{1-\mu}\right). \end{aligned} \tag{7}$$

Next, we modify (2) and then use (5) and (7) to get the expressions of $E[T_z]$ and $E[T'_z]$. It follows that:

$$E[T_z] = \frac{\tau_0}{p^*} = \frac{\tau_0(1-\mu)}{p(1-3\mu)}. \tag{8}$$

$$E[T'_z] = \Delta + \frac{\tau_0}{p^*} = \Delta + \frac{\tau_0(1-\mu)}{p(1-3\mu)}. \tag{9}$$

Obviously, one miner's average mining time and his average mining speed are the reciprocal of each other. Thus we can use the expression of $E[T'_z]$ in (9) to denote the growth rate β. It

follows that:

$$\beta = \frac{1}{E[T_z']} = \frac{p(1-3\mu)}{\Delta p(1-3\mu) + \tau_0(1-\mu)}. \quad (10)$$

In Section II, we indicated that the distribution function $Pr[X_z = k]$ is not a Poisson distribution. The time spent for mining one block is memoryless and the total mining time has a gamma distribution [4], so the time variable t in the coefficient of Poisson distribution is a random variable with a gamma distribution rather than an expected value. According to the probability formula proposed by Grunspan [4], some variables in the original probability formula need to be modified. The honest miners' mining speed α' in the original formula is replaced by the growth rate β of the main chain. Therefore, the modified distribution function can be denoted as

$$Pr[X_z = k] = \int_0^{+\infty} \frac{(\alpha t)^k}{k!} e^{-\alpha t} \frac{\beta^z}{(z-1)!} t^{z-1} e^{-\beta t} dt. \quad (11)$$

Next, we use some properties of the gamma distribution and gamma function [23], [24] to simplify (11). First suppose

$$\lambda_1 = \frac{\alpha}{\alpha+\beta}\ ,\ \lambda_2 = \frac{\beta}{\alpha+\beta}$$

we then simplify (11) as

$$Pr[X_z = k] = \lambda_1^k \lambda_2^z \binom{k+z-1}{k}. \quad (12)$$

As shown, the attacker's mining process in our proposed attack model is also a negative binomial distribution, i.e., the same as the attacker's mining process shown in [4], [5].

We already know the probability $Pr[X_z = k]$, the last step of calculating the probability of successful double-spending with the Sybil attack is to determine the probability (i.e., P_{z-k}) that the attacker catches up the main chain in the case when he has mined k blocks. Nakamoto had studied this problem before. He thought that this problem was analogous to a Gambler's Ruin problem and could be solved by using the property of Markov chain [1], but here we need to modify Nakamoto's formula. In our hypothetical case, the attacker's mining speed is no longer compared to the honest miners' mining speed, but rather the growth rate of the main chain. So the mining speed α' of the honest miners is also replaced by β. It follows that:

$$P_{z-k} = \left(\frac{\alpha}{\beta}\right)^{z-k}. \quad (13)$$

Equations (5), (10), (12), and (13) can be used to do a simplification on (1). Therefore, the final formula is

$$P = 1 - \sum_{k=0}^{z-1} (\lambda_2^z \lambda_1^k - \lambda_1^z \lambda_2^k) \binom{k+z-1}{k}, \quad (14)$$

where

$$\lambda_1 = \frac{q\left(\Delta p(1-3\mu) + \tau_0(1-\mu)\right)}{q\left(\Delta p(1-3\mu) + \tau_0(1-\mu)\right) + \tau_0 p(1-3\mu)},$$

$$\lambda_2 = \frac{\tau_0 p(1-3\mu)}{q\left(\Delta p(1-3\mu) + \tau_0(1-\mu)\right) + \tau_0 p(1-3\mu)}.$$

We set $\Delta = 100$ s to calculate (14) to get P for the proposed attack model. We compare P with the probability of a successful double-spending in Nakamoto's attack model (i.e., double-spend attack without the help of any attack). Note that the formula of the probability calculation of Nakamoto's attack model is different from that proposed in [1]. Instead, we also use (14) and modify the values of λ_1 and λ_2 since there are no chain forks and block propagation delay in Nakamoto's attack model. Fig. 3(a) presents our comparison results. From Fig. 3(a), we can clearly see the results with different ratio μ when $q = 0.25$. In general, with the number of blocks z increases, the probability P decreases. However, we can see the effect of the Sybil attack. The probability P is significantly improved compared to the case without a Sybil attack (i.e., Nakamoto's model). Even in the case where z is equal to 10, P exceeds 20% when $\mu = 16\%$ and is close to 50% when $\mu = 20\%$. Moreover, it is obvious to find that the more the number of Sybil nodes, the greater the probability of success of our proposed attack. Note that the situation where $\mu = 20\%$ is the theoretical best effect of our proposed attack. When $\mu = 20\%$, $\mathcal{H}_S$ have the same computing power as $\mathcal{H}_R$. Hence, there may be two forked chain with the same length. The target that the attacker wants to catch up is always the longest chain. When μ increases, the target pursued by the attacker becomes the forked chain created by $\mathcal{H}_S$. In this case, the effect of Sybil attack will be reduced, which explains why the calculation of P is performed under the assumption $\mu \leq 20\%$.

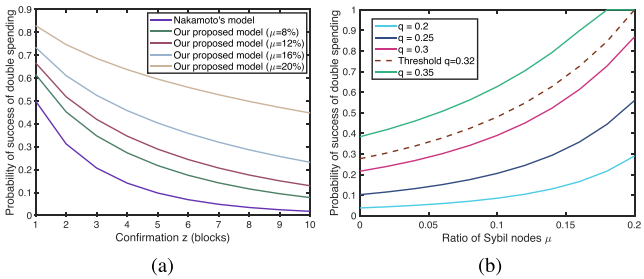

Fig. 3. P changes as z and μ increase. (a) Compared with Nakamotos model. (b) Threshold.

Fig. 3(b) depicts the change of P with the increase of μ under the attacker's different computing power when $z = 6$ (the number of security blocks set up by Bitcoin [5]). Fig. 3(b) also presents the minimum value of the attacker's computing power q that can thoroughly destroy the Bitcoin blockchain. From Fig. 3(b), we can discover that the attacker with $q \geq 0.32$ has a 100% probability of a successful attack when he deploys at most $20\%N$ Sybil nodes, in which $q = 0.32$ is the threshold of destroying the Bitcoin blockchain. In other words, holding a 32% share of the computing power is sufficient to successfully rewrite the blockchain history, which is far lower than $q = 50\%$ proposed by Nakamoto [1] and $q = 49.1\%$ proposed by Decker [15]. Consequently, combining the above analysis, we conclude that the Sybil attack can greatly improve the effect of the double-spend attack.

VI. Economic Evaluation

In this section, we do a break-even analysis on the attack model through calculating the profit and loss of the attacker.

Through this economic analysis, we also provide a way to evaluate the attack model from the perspective of both sides of the mining game: For the attackers, how can they maximize their profits in the proposed attack? For the honest miners, how can they reduce the desire of attackers to attack?

We simplify this complex economic problem by assuming the following:

1) The value of a goods the attacker wants to get is v. We assume that this value is the same for both the merchant and the attacker.
2) As we know, if the double-spend attack succeeds, the attacker will get the value of a goods without paying v. Moreover, if this attack fails, the attacker will still get v whereas he must pay v to the merchant to buy this commodity.
3) Besides, the attacker will also pay the cost of mining blocks on his own branch. We assume that the cost of mining one block by the attacker is equal to the reward for each block [5]. The cost of mining blocks is the product of rewards each block and the number of blocks. Thus, in case of winning the race, the attacker will publish his own blocks and earn the whole block rewards himself, thus offsetting the cost of mining. On the contrary, if the attacker fails, all the blocks he mined will be rejected and he will lose the whole rewards of these blocks.

Because it is quite difficult to estimate the time spent and the number of blocks the attacker has mined when this attack is successfully completed in the case where $k < z$, we determine the scope of the economic evaluation, which is the attacker's break-even point within the time S'_z the merchant waits for z confirmations. Now we start the calculation of the attacker's cost. We denote by O the expected value of the number of blocks mined by the attacker within the time the merchant waits for z confirmations. Let r denote each block reward. If the attacker fails, which happens with probability $1 - P$, he will lose the total value $v + Or$ within S'_z. So the cost of his attack within S'_z is

$$C = (1 - P)(v + Or). \tag{15}$$

Then, we know that whether the attacker succeeds or fails, his bonus is always v, it is quite easy to get the formula of attacker's profits

$$\pi = v - (1 - P)(v + Or). \tag{16}$$

We can use some of the known variables mentioned before to represent O. O is equal to the expected value of the number of blocks mined per unit time multiples the expected value of S'_z (the number of blocks mined per unit time and S'_z are independent random variables). The expected value of the number of blocks mined per unit time is actually the attacker's average mining speed α. So the remaining problem is the calculation of the expected value of S'_z. We already know the expected value of T'_z from (9), so $E[S'_z]$ equals the number of blocks mined by the honest miners within S'_z multiple $E[T'_z]$. It follows that:

$$E[S'_z] = z \times E[T'_z] = z\left(\Delta + \frac{\tau_0(1-\mu)}{p(1-3\mu)}\right). \tag{17}$$

$$O = \alpha \times E[S'_z] = \frac{qz}{\tau_0}\left(\Delta + \frac{\tau_0(1-\mu)}{p(1-3\mu)}\right). \tag{18}$$

Next, (16) can be simplified by using (18) as follows:

$$\pi = v - (1 - P)\left(v + \frac{qzr}{\tau_0}\left(\Delta + \frac{\tau_0(1-\mu)}{p(1-3\mu)}\right)\right) \tag{19}$$

where r is set to 12.5 BTC, τ_0 is set to 600 s in Bitcoin. The attacker's break-even point is a situation where his cost equals his revenue, so we set $\pi = 0$. Then, we can get the expression of the attacker's break-even point

$$v = \frac{qzr(1-P)\left(\Delta + \frac{\tau_0(1-\mu)}{p(1-3\mu)}\right)}{P\tau_0}. \tag{20}$$

By calculating (20), we can get the attacker's different break-even points with the increase of μ and z under $q = 0.25$. The range of variables is set that the number of blocks z goes from 2 to 10 with a step length of 2 and the ratio of Sybil nodes μ goes from 6 to 20% with a step length of 0.02. The different results of break-even points are displayed in Table II.

From Table II, we can easily discover that under any ratio of Sybil nodes μ, the break-even point dramatically increases with the growth of z. Fig. 4(a) and (b) indicates this change more intuitively under the case $q = 0.25$ and $q = 0.3$. Obviously the longer the merchant spends waiting for confirmations, the lower the probability of successful attack. Besides, Fig. 4 also illustrates that under any z, the break-even point decreases steadily

TABLE II
ATTACKER'S BREAK-EVEN POINT UNDER DIFFERENT μ AND z

μ \ z	2	4	6	8	10
6%	14.27	64.94	178.46	400.33	805.31
8%	13.47	59.15	157.01	340.08	660.12
10%	12.59	53.22	136.21	284.42	531.98
12%	11.61	47.13	116.08	233.35	420.01
14%	10.50	40.84	96.64	186.75	323.11
16%	9.24	34.32	77.85	144.44	240.05
18%	7.77	27.47	59.64	106.14	169.36
20%	6.02	20.16	41.82	71.35	109.34

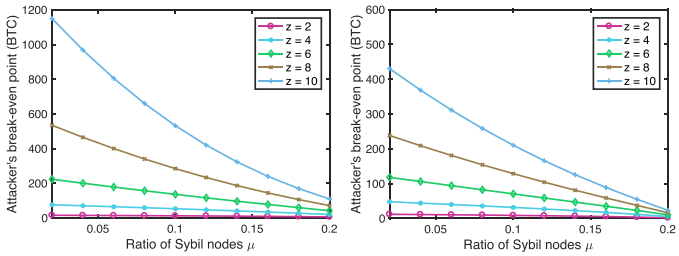

Fig. 4. Variations of the attacker's break-even point with μ increases. (a) $q = 0.25$. (b) $q = 0.3$.

with the growth of μ and q. In general, it is easier for the attacker to make a profit by deploying a number of Sybil nodes than in the case where a Sybil attack is not used.

In order to profit from the proposed attack, the attacker needs to make sure that his revenue is greater than his break-even point. So when his break-even point is low, it will be quite easy for him to benefit from attacking. In the discussion of calculating the probability of successful attack, we have already learnt that the probability P will increase as μ grows. It means that if the attacker succeeds in the proposed attack with a higher probability P, he will have a greater chance of offsetting his mining cost. Accordingly, in case P is high, the attacker's break-even point will be very low, which is good for him. In summary, with the influence of the Sybil attack, not only does the probability of success of double-spend attack increase but the attacker can also have a great chance to get a great profit. According to the results shown in Fig. 4, we can know that the honest miners can either prolong the number of block confirmations for waiting or increase the value of goods, forcing the attacker to abandon attacking due to the high cost and low profit of the proposed attack.

VII. Conclusion

In this paper, we proposed a combined attack model which used a Sybil attack to improve the probability of success of double-spending in the Bitcoin network. One attacker can create lots of Sybil nodes with multiple fake identities to delay the propagation of valid blocks to help him dominate in the mining race with honest miners. Meanwhile, under the influence of Sybil nodes, the blockchain forks will occur, thus causing the waste of the computing power of some honest nodes. Our findings demonstrated that the probability of success of the proposed attack was much greater than that of Nakamoto's attack model in any case and a 32% share of the computing power in the Bitcoin network was enough for the attacker to rewrite the blockchain history. In addition, we also analyzed the proposed attack from the perspective of economics. The results illustrated that the proposed attack can also make the attacker easily profit. Our suggestion is that the number of confirmed blocks and the value of goods must be improved to prevent the attacker's malicious behavior.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018.

[3] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, IEEE, 2017, pp. 557–564.

[4] C. Grunspan *et al.*, "Double spend races," *J. Enterprising Culture (JEC)*, vol. 21, no. 08, pp. 1–32, 2018.

[5] M. Rosenfeld, "Analysis of hashrate-based double spending," *Preprint, 2014, arXiv:1402.2009*.

[6] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proc. ACM Conf. Comput. Commun. Secur.*, ACM, 2012, pp. 906–917.

[7] J. R. Douceur, "The sybil attack," in *Proc. Int. Workshop Peer-to-Peer Syst.*, Springer, 2002, pp. 251–260.

[8] G. Bissias, B. N. Levine, A. P. Ozisik, and G. Andresen, "An analysis of attacks on blockchain consensus," 2016, *arXiv:1610.07985*.

[9] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *Proc. USENIX Secur. Symp.*, 2015, pp. 129–144.

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

[11] 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 *Proc. ACM Special Interest Group Secur., Audit Control (SIGSAC) Conf. Comput. Commun. Secur.*, ACM, 2016, pp. 3–16.

[12] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proc. 1st IEEE Int. Workshop Sensor Netw. Protocols Appl.*, IEEE, 2003, pp. 113–127.

[13] G. Bissias, A. P. Ozisik, B. N. Levine, and M. Liberatore, "Sybil-resistant mixing for bitcoin," in *Proc. 13th Workshop Privacy Electron. Soc.*, ACM, 2014, pp. 149–158.

[14] *Proof of stake versus proof of work white paper*, 2015. [Online]. Available: https://bitfury.com/content/downloads/pos-vs-pow-1.0.2.pdf

[15] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proc. IEEE P2P*, IEEE, 2013, pp. 1–10.

[16] A. Kiayias and G. Panagiotakos, "Speed-security tradeoffs in blockchain protocols," *IACR Cryptology ePrint Archive*, vol. 2015, p. 1019, 2015. [Online]. Available: https://eprint.iacr.org/2015/1019.pdf

[17] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, Springer, 2015, pp. 281–310.

[18] R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, Springer, 2017, pp. 643–673.

[19] P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using P2P network traffic," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, Springer, 2014, pp. 469–485.

[20] A. Demers *et al.*, "Epidemic algorithms for replicated database maintenance," *ACM Special Interest Group Operating Syst. (SIGOPS) Operating Syst. Rev.*, vol. 22, no. 1, pp. 8–32, 1988.

[21] M. Jelasity, "Gossip," in *Self-Organising Software*. Berlin, Heidelberg, Germany: Springer, 2011, pp. 139–162.

[22] L. Alvisi *et al.*, "How robust are gossip-based communication protocols?" *ACM Special Interest Group Operating Syst. (SIGOPS) Operating Syst. Rev.*, vol. 41, no. 5, pp. 14–18, 2007.

[23] R. D. Gupta and D. Kundu, "Exponentiated exponential family: An alternative to gamma and weibull distributions," *Biometrical J., J. Math. Methods Biosciences*, vol. 43, no. 1, pp. 117–130, 2001.

[24] E. Artin, *The Gamma Function*. New York, NY, USA: Courier Dover Publications, 2015.

Shijie Zhang received the B.S. degree in information management and system from Nanjing Forestry University, Nanjing, China, in 2017. Since 2017, he is working toward the master's degree with the Department of Software at Sangmyung University, Cheonan, Republic of Korea.

He is a Research Assistant with the Protocol Engineering Lab, Sangmyung University. His research interests include system security and management, network security, and blockchains.

Jong-Hyouk Lee (M'07–SM'12) received the Ph.D. degree in computer engineering from Sungkyunkwan University, Suwon, Republic of Korea, in 2010.

He is now leading the Protocol Engineering Lab., Sangmyung University, Cheonan, Republic of Korea. He is an author of the Internet Standards: IETF RFC 8127 and IETF RFC 8191. His research interests include protocol engineering and performance analysis.

Prof. Lee was the recipient of the IEEE Best Land Transportation Paper Award in 2015, Haedong Young Scholar Award in 2017, and IEEE Systems Journal Best Paper Award in 2018.