Revisiting Double-Spending Attacks on the Bitcoin Blockchain: New Findings

Conference Paper · June 2021

CITATIONS
0

READS
241

5 authors, including:

Jian Zheng
Sun Yat-Sen University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Huawei Huang
Sun Yat-Sen University
80 PUBLICATIONS **1,409** CITATIONS
SEE PROFILE

Canlin Li
Sun Yat-Sen University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Zibin Zheng
Sun Yat-Sen University
332 PUBLICATIONS **12,655** CITATIONS
SEE PROFILE

论文中，红色荧光表示系统模型的假设；蓝色荧光表示定义；黄色荧光表示文章的创新点；绿色荧光表示文章中某些重要定理。红色下划线表示文章中写的比较好的英文；绿色下划线表示文章中仿真结果和结论。

Revisiting Double-Spending Attacks on the Bitcoin Blockchain: New Findings

Jian Zheng*, Huawei Huang*, Canlin Li*, Zibin Zheng* Song Guo†
*School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China
†Department of Computing, The Hong Kong Polytechnic University, Hong Kong. song.guo@polyu.edu.hk
Corresponding author: Huawei Huang, huanghw28@mail.sysu.edu.cn

***Abstract*—Bitcoin is currently the cryptocurrency with the largest market share. Many previous studies have explored the security of Bitcoin from the perspective of blockchain mining. Especially on the double-spending attacks (DSA), some state-of-the-art studies have proposed various analytical models, aiming to understand the insights behind the double-spending attacks. However, we believe that advanced versions of DSA can be developed to create new threats for the Bitcoin ecosystem. To this end, this paper mainly presents a new type of double-spending attack named Adaptive DSA in the context of the Bitcoin blockchain, and discloses the associated insights. In our analytical model, the double-spending attack is converted into a Markov Decision Process. We then exploit the Stochastic Dynamic Programming (SDP) approach to obtain the optimal attack strategies towards Adaptive DSA. Through the proposed analytical model and the disclosed insights behind Adaptive DSA, we aim to alert the Bitcoin ecosystem that the threat of double-spending attacks is still at a dangerous level.**

***Index Terms*—Bitcoin Blockchain, Double-Spending Attack**

1. Introduction

Following the drastically increasing market price of Bitcoin [1] in the past few months, the cryptocurrencies continually attract ever-growing interests from all over the world. However, cryptocurrencies, e.g., Bitcoin, face the inherent threat of Double-Spending Attacks (DSA). For example, the Bitcoin Gold (BTG) [2] was stolen more than 388200 BTG coins by double-spending attacks on May 2018, and suffered from a loss of about 18.6 million dollars. On January 2019, the Ethereum Classical [3] blockchain has occurred the roll-back anomaly, which was very possibly attacked by double-spending attacks. This anomaly led to a loss of around 1.1 million dollars. Recently on December 2020, the famous cryptocurrency Aeternity [4] also was attacked by the double-spending attack, resulting in a huge loss of approximately 39 million Aeternity coins.

Since Bitcoin has become the top market-share cryptocurrency [5], [6], many organizations have invested a large amount of money as the strategic digital assets. In reality, Bitcoin miners prefer to join in a mining pool because of brutal competition of Proof-of-Work (PoW) based mining among miners. The mining-pool phenomenon leads to that most of the mining power is centralized in a small number of mining pools. Such centralization brings a high vulnerability of suffering from mining attacks. Furthermore, in Bitcoin market, the number of large-amount transactions is increasing when the number of exchanges grows. The aggregated money in exchanges draws great attention from cryptocurrency hackers. Once the double-spending attack occurs in an exchange, e.g., the Mt. Gox attack [7], the Bitcoin market will experience a drastic fluctuation. Therefore, the security of the Bitcoin blockchain has raised great attention, especially in the context of the real double-spending attacks aforementioned.

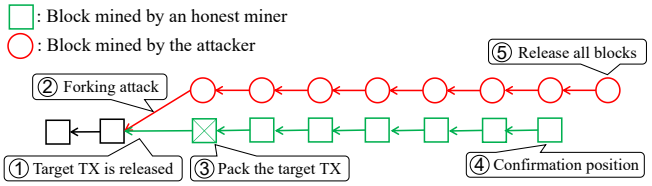

Figure 1. Illustration of a double-spending attack.

Literature Review. The double-spending attack is a classic problem in blockchains. A number of previous works have explored the double-spending attacks on Bitcoin blockchain. Nakamoto [1] raised the problem of double-spending attacks for the first time in the Bitcoin system. Pinzon *et al.* [8] proposed two new double-spending attacks: race attack and fanny attack. The race attack achieves attacking by controlling miner fees. The fanny attack chooses to control block broadcast time to achieve attacking. Liao *et al.* [9] found that a small number of attackers can increase the probability of a 51% double-spending attack by connecting rational nodes. Biais *et al.* [10] proved that it is profitable to try a 51% double-spending attack when the transaction price meets certain conditions, but such an attack requires a large amount of hashpower and money. Gervais *et al.* [11] proposed a simple double-spending attack strategy model based on Markov Decision Processes (MDP) and revealed the effectiveness of the double-spending attack strategy.

Through reviewing those representative analytical models proposed in the past few years, we have the following findings.

- Many existing attack policies [12], [13] aim at their success probabilities. However, a high success attack probability does not mean a high return of reward.

978-0-7381-3207-5/21/$31.00 © 2021 IEEE

- Several papers [1], [14] did not consider the strict limitation of the duration of attacks. Instead, they only calculated the average profit of double-spending attacks. However, in practice, the budget of the attacker is limited, making the average cost of an attack cannot be too high.

Motivation. Inspired by those facts aforementioned, in this paper we revisit the double-spending attacks towards the Bitcoin blockchain, by disclosing several insights through a tricky attack strategy. Note that, we are not aiming to launch new types of double-spending attacks. What we study in this paper is trying to alert the cryptocurrency blockchain using the following new findings: The DSA attackers can launch the attacks even though when their mining power is far less than 51% of the entire Bitcoin blockchain. Using our theoretical framework, both the users of Bitcoin blockchain and the developers of a new blockchain can estimate how well their blockchain to defend the double-spending attacks.

Contributions.

- Firstly, Based on Naive DSA, we propose a new type of double-spending attack, i.e., Adaptive DSA. We then exploit the SDP-based theoretical framework to generate the profit-maximized attack strategy under Adaptive DSA.
- Secondly, we believe that our proposed analytical model can stimulate researchers and engineers to propose new blockchain designs that are able to defend new advanced double-spending attacks.

The remaining of this paper is organized as follows. Section 2 introduces the preliminaries of the paper. Section 3 analyzes the new type of double-spending attacks. Section 4 shows the simulation results of our analytical model. Section 5 concludes this paper.

2. Preliminaries and Assumptions

In this section, we introduce the simplest double-spending attack (named Naive DSA) and the basic preliminaries of the paper, which is the same with that of other studies.

As shown in Fig. 1, if a transaction recorded in a block and receives confirmation from a specific number of new blocks (represented by $z(\in N_+)$, which is normally 6 for Bitcoin), it will be viewed as *confirmed* by the blockchain. When an attacker intends to tamper this confirmed transaction, he must launch a forking attack by creating a new branch of blocks, which is required longer than the main chain. Once such forking attack succeeds, we say that the attacker has launched a double-spending attack, through which the attacker can steal the money residing in the target transaction. The reward of mining a new block is denoted by d, which includes two aspects: the `coinbase` reward plus the transaction fees. The stolen monetary coins of a double-spending attack is represented by b.

We assume that there are only two types of miners existing in the blockchain network, i.e., the double-spending attacker and all the other honest miners. Let p and q ($p + q = 1$) represent the hashpower proportions of the attacker and the honest miners, respectively. We consider a fine-grained timeslot for our analytical framework, in which at most only a single new block can be generated within a timeslot. The duration of the attack is very short, therefore we assume that the mining difficulty keeps unchanged.

The honest miner always follows the policy of *longest chain first* by default. If two forks with the equal length exist, honest miners will choose the one containing more historical blocks generated by themselves. According to the length of the fork controlled by the attacker, the results of forking attack include `fail`, `match`, and `success` when the fork is shorter, equal to, and longer comparing with the main chain, respectively.

Table 1. SYMBOLS AND NOTATIONS

b	the amount of stolen money via double-spending the target transaction (TX)
p	the proportion of the total hashpower controlled by the attacker
q	the proportion of the total hashpower controlled by honest miners, $p + q = 1$
d	the average reward for mining a new block
$z\ (\in N_+)$	(N_+ denotes positive integers) the number of blocks between a target TX and its confirmation position
$i\ (\in N)$	the # of blocks generated by the attacker
$j\ (\in N)$	the # of blocks generated by honest miners
$cost$	the average cost of the attacker in the process of generating a new block in the blockchain system
$s_{i,j}$	the state that the attacker has generated i blocks and honest miners have generated j blocks
$d_{i,j}$	the decision the attacker makes in state $s_{i,j}$
$P_{i,j}$	the probability of state $s_{i,j}$ arrival in each attack
$R_{i,j}$	the attacker's reward when he stops at state $s_{i,j}$
$J_n(i,j)$	the revenue expectation of state $s_{i,j}$, $n = i + j$
$T(s_{i,j}, d_{i,j})$	the state transition probability distribution in state $s_{i,j}$ with decision $d_{i,j}$
$g(s_{i,j}, d_{i,j})$	the benefit function of state $s_{i,j}$ with decision $d_{i,j}$
$s'_{i,j}$	a special state indicating the attacker releases all the sub-chain and begins to mine on the longest chain

假设：
1. 交易被包含在区块中，并接受特定数量新块的确认；
2. 攻击者想更改已确认的交易需要发起分叉攻击，并生成比有效链更长的链；
3. 假设区块链网络中存在两种矿工：攻击者和诚实矿工；
4. 假设在一个时隙中最多能够生成一个区块；
5. 假设挖矿难度保持不变；
6. 诚实矿工都遵循最长链原则
7. 根据攻击者控制的分叉长度，结果可能为：fail, match, success。

3. Adaptive Double-Spending Attack

In this section, we describe the proposed new type of double-spending attack, i.e., the Adaptive Double-Spending Attack (shorten as *Adaptive DSA*). Typically, we calculate the optimal attack decisions and the theoretical boundary of the expected reward by launching the Adaptive DSA.

3.1. System Model of Adaptive DSA

The strategy of Naive DSA can make the success probability maximized, when the main chain has generated z blocks with respect to the target transaction. However, in some special situations, taking Fig. 2 for an example, most of the hashpower is controlled by the honest miners, such that the number of blocks created by the attacker will be

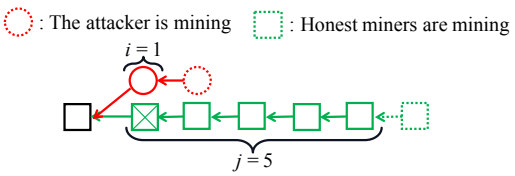

Figure 2. An unfavourable situation to the double-spending attacker.

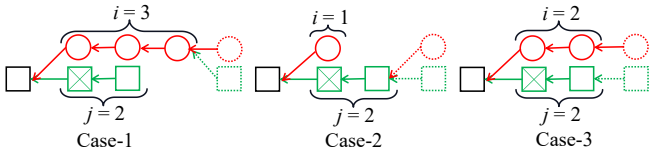

Figure 3. Three possible cases of the decision under Adaptive DSA $d_{i,j} = 0$: quit attacking.

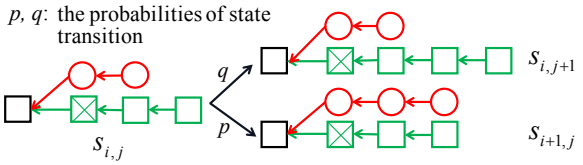

Figure 4. The decision of Adaptive DSA $d_{i,j} = 1$: keep attacking. A decision $d_{i,j}$=1 also triggers state transition: $s_{i,j} \rightarrow s_{i+1,j}$ or $s_{i,j+1}$.

在发现情况不利时，攻击者会放弃攻击，从而避免更多的损失

too few. Under such the unfavourable situation, the success probability of the attacker is very low. To prevent a large loss brought by the failed double-spending attacks, the attacker is suggested giving up the attack when the situation is unfavourable. Therefore, we are motivated to propose the Adaptive DSA. In the following, we analyze the Adaptive DSA by particularly evaluating those unfavourable situations illustrated in Fig. 2.

To calculate the reward under the Adaptive DSA, we let *cost* denote the cost that the attacker spends during the process of any new block is generated. To achieve a DSA, the attacker has to keep generating new blocks in his fork. Thus, *cost* includes not only the maintaining fee to run the mining machines, but also the *opportunistic loss* if the attacker were using his hashpower to mine blocks just as an honest miner. Once the attacker succeeds, its reward is calculated as the mining reward minus to the maintaining fee. Thus, we have $cost = p \cdot d$, $p > 0$.

To the main chain, when the number of confirmation blocks reaches z in a faster speed than the attacker's fork, the target transaction is viewed as confirmed deeply. The attacker has to quit the current attack, and prepare for the next. That is, when $j > z$, the attacker stops mining in its fork. On the other side, once the attacker generates the $(z+1)$-th block before the main chain outputs the z-th block, the double-spending attack succeeds. To maximize the profit of the attack, the attacker may also keep mining more blocks in his fork, before the main chain discovers its z-th confirmation block. Therefore, when $i \geq z$ & $i > j$, the attacker will keep attacking in his fork. Except the two cases described above, the attacker cannot make a determined decision when $0 \leq i, j \leq z-1$. The goal of this section is trying to find an optimal attack-decision matrix such that the profit of Adaptive DSA can be maximized.

3.2. Terminologies

3.2.1. Decision Point. Normally, an honest miner would not change the chain it follows, either after the main chain or a fork. Only when a new block is generated, the miner has to decide whether to change the fork it follows currently or not. Therefore, we define the instant when a new block is generated as a *decision point*, which is denoted by $s_{i,j}$, $(i \in N,\ j \in N)$. Here i and j represent the numbers of the attacker's blocks in the fork chain and the confirmation blocks in the main chain, respectively. When reaching a decision point, the attacker has to decide whether to continue attacking or quit.

3.2.2. Attack-Decision Matrix. We then let $d_{i,j} = 0/1$ denote the attack decision of the attacker while the attack is in a specific decision point $s_{i,j}$. In the attack-decision matrix $\{d_{i,j}\}$, every element $d_{i,j}$ values 0 or 1. The definition of the binary decision is as follows.

$$d_{i,j} = \begin{cases} 1, & \text{if } i = j = 0; \\ 0, & \text{if } i = z \text{ or } j = z; \\ 0 \text{ or } 1, & \text{otherwise.} \end{cases}$$

When $d_{i,j} = 0$, the attacker releases the fork, resulting in 3 possible cases shown in Fig. 3. Case-1: if $i > j$, the attacker dominates the block mining during the attack, and receives the block-mining reward immediately. Furthermore, if $i > j$ & $j \geq z$, the attacker succeeds the double-spending attack. Case-2: if $i < j$, the attacker receives nothing. Case-3: if $i = j$, the attacker will follow the fork mined by himself.

The decision $d_{i,j} = 1$ indicates that the attacker still keeps attacking in the fork chain until the next decision point shows up. As shown in Fig. 4, the current decision point $s_{i,j} (0 \leq i, j \leq z-1, i, j \in N)$ will be transferred to either state $s_{i+1,j}$ or state $s_{i,j+1}$, $0 \leq i, j \leq z-1$.

3.2.3. Occurrence-Probability Matrix. In the occurrence-probability matrix, represented by $\{P_{i,j}\}$, each element $P_{i,j}$ describes the probability of achieving a decision point $s_{i,j}$ during an attack under a given decision matrix $\{d_{i,j}\}$. In particular, $s_{0,0}$ is always the starting point of any attack. Thus, $P_{0,0} = 1$. As described in section 3.2.2, $s_{i,j}$ stems from $s_{i-1,j}$ with probability p if $d_{i-1,j} = 1$, or $s_{i,j-1}$ with probability q if $d_{i,j-1} = 1$. Note that, if $j = 0$ or $i = z$, $s_{i,j}$ can only stem from $s_{i-1,j}$ with probability p. Similarly, if $i = 0$ or $j = z$, $s_{i,j}$ can only stem from $s_{i,j-1}$ with probability q. Therefore, we can define the occurrence-probability matrix $\{P_{i,j}\}$ as follows.

$$P_{i,j} = \begin{cases} 1, \text{ if } i = j = 0; \\ d_{i-1,j} \cdot p \cdot P_{i-1,j}, \text{ if } j = 0 \text{ or } i = z; \\ d_{i,j-1} \cdot q \cdot P_{i,j-1}, \text{ if } i = 0 \text{ or } j = z; \\ d_{i-1,j} \cdot p \cdot P_{i-1,j} + d_{i,j-1} \cdot q \cdot P_{i,j-1}, \text{otherwise.} \end{cases} \tag{1}$$

3.3. Profit-Maximization Problem

Given an attack-decision matrix $\{d_{i,j}\}$, we can derive the occurrence probability matrix $\{P_{i,j}\}$ according to the stochastic state transition. Next, the expected reward can be also calculated. Let $R_{i,j}$ denote the expected reward when the attacker stops attacking at a decision point $s_{i,j}$, i.e., $d_{i,j}$ turns to 0 at $s_{i,j}$. When stopping attacking at $s_{i,j}$, the attacker will release the blocks mined in his fork, and participate in the regular mining just as an honest miner following the main chain. Thus, $R_{i,j}$ is calculated by the `coinbase` reward minus the mining cost:

$$R_{i,j} = \begin{cases} -(i+j) \cdot cost, \text{ if } i < j; \\ i \cdot d - (i+j) \cdot cost, \text{ if } z > i > j; \\ p \cdot i \cdot d - (i+j) \cdot cost, \text{ if } z > i = j; \\ p[b + d(z+1) - (i+j+1) \cdot cost] + q \cdot R_{i,j+1}, \\ \quad \text{if } i = z, j \leq z-2; \\ p[b + d(z+1) - (i+j+1) \cdot cost] + \\ \quad q[p(b + z \cdot d) - 2z \cdot cost], \text{ if } i = z, j = z-1. \end{cases} \tag{2}$$

With $f(\{d_{i,j}\})$ denoting the expected reward function under a decision matrix $\{d_{i,j}\}$, the goal is to strive for the optimal attack-decision matrix such that the expected reward is maximized. We then present the profit-maximization problem as follows.

$$\begin{aligned} &\max \ f(\{d_{i,j}\}) = \sum_{i=0}^{z} \sum_{j=0}^{z} (1 - d_{i,j}) \cdot P_{i,j} \cdot R_{i,j} \\ &s.t. \text{ Eq. (1) and Eq. (2).} \\ &\text{Variables} : d_{i,j} \in \{0,1\}, \ 0 \leq i,j \leq z, \ i,j \in N. \end{aligned} \tag{3}$$

Solving the problem (3) is infeasible when z is large, since the problem complexity increases exponentially following z. Therefore, we need to design a complexity-efficient approach to solve this problem.

3.4. Problem-Solving based on Stochastic Dynamic Programming (SDP)

In fact, the optimization problem (3) can be viewed as a two-dimensional multi-stage Stochastic Dynamic Programming (SDP) problem [15]. At the decision point $s_{i,j}$, we only need to consider the current state as well as the future expected reward, regardless what decisions have been made and how to get to that state. In other words, the problem (3) has the Markovian property.

3.4.1. SDP-based Formulation. In the context of a decision-making problem, the key elements of the SDP model are described as follows.

Stages: In a $z \times z$ decision matrix, each sub-diagonal represents a stage. We mark the sub-diagonal as 0, 1, 2, $\cdots$, $2z$-1 from the upper left to the lower right, and the whole system has a total of $2z$ stages.

States: The set of states in stage n is denoted as $S[n] = \{s_{i,j} \ | i+j = n\} \cup \{s'_{i,j} \ | i+j = n-1\}$, when $n > 0$. Especially, if $n = 0$, $S[n] = \{s_{i,j} \ | i+j = n\}$. In addition, $s'_{i,j}$ is a special state indicating that the attacker releases all the sub-chain and begins to mine on the longest chain.

Decisions: We use $d_{i,j}$ to denote the decision variable at $s_{i,j}$, $d_{i,j} \in U_{i,j}$, where $U_{i,j}$ denotes the set of all decisions.

Transition probability: The transition-probability function $T(s_{i,j}, d_{i,j})$ returns the probability that the current state $s_{i,j}$ transfers to the next state by executing the decision $d_{i,j}$. In SDP framework, we can only get the probability distribution of the next state instead of accurate states. Referring to section 3.2, the transition-probability function is written as:

$$\begin{aligned} T(s_{i,j}, d_{i,j}) = \begin{cases} p, \text{ if } s_{i,j} \to s_{i+1,j}, d_{i,j} = 1; \\ q, \text{ if } s_{i,j} \to s_{i,j+1}, d_{i,j} = 1; \\ 1, \text{ if } s_{i,j} \to s'_{i,j}, d_{i,j} = 0; \end{cases} \\ 0 \leq i,j \leq z-1, \ i,j \in N. \end{aligned} \tag{4}$$

Reward function: The reward function $g(s_{i,j}, d_{i,j})$ returns the reward when executing a decision $d_{i,j}$ at the current state $s_{i,j}$. The reward function is defined as:

$$\begin{aligned} g(s_{i,j}, d_{i,j}) = \begin{cases} 0, & \text{if } d_{i,j} = 0; \\ -cost, & \text{if } d_{i,j} = 1; \end{cases} \\ 0 \leq i,j \leq z-1, i,j \in N. \end{aligned} \tag{5}$$

3.4.2. Optimal Decision-Making. When the decision matrix $\{d_{i,j}\}$ is fixed, the profit of the attacker can be derived. Let $J_n(s_{i,j})$ denote the profit matrix, where $n = i+j$. Recall that, our objective is to maximize the expected profit by deciding the optimal decision matrix. According to Bellman optimality theorem [15], the objective function is expressed as:

$$\begin{aligned} J_n(s_{i,j}) &= \max_{d_{i,j}} \{g(s_{i,j}, d_{i,j}) + E[J_{n+1} | s_{i,j}, d_{i,j}]\} \\ &= \max_{d_{i,j}} \{g(s_{i,j}, d_{i,j}) + T(s_{i,j}, d_{i,j}) \cdot J_{n+1}(S[n+1])\} \\ &= \max\{J_{n+1}(s'_{i,j}), \ -cost + p \cdot J_{n+1}(s_{i+1,j}) \\ &\quad + q \cdot J_{n+1}(s_{i,j+1})\}, \ 0 \leq i,j \leq z-1, \ i,j \in N. \end{aligned} \tag{6}$$

where $s'_{i,j}$ in fact indicates the special state occurring in the stage $S[n+1]$. In addition, we should define the boundary condition of $J_n(s_{i,j})$. For the right boundary (i.e., $0 \leq i \leq z-1, \ j = z$), the attacker is failed and gains nothing. Thus, $J_{i,j} = 0$ $(0 \leq i \leq z-1, \ j = z)$. For the lower boundary (i.e., $0 \leq j \leq z-1, \ i = z$), the target transaction has not been confirmed and the attacker should keep mining until $j = z$. The attack is successful if $i = z+$

Algorithm 1: SDP for Adaptive DSA

Input : $b, p, d, z, cost$.
Output: decision matrix $\{d_{i,j}\}$, expected reward.
Initialize boundary conditions according to Eq. (7), Eq. (8), and set $d_{0,0} = 1$.
for $i = z - 1$ *to* 0 **do**
  for $j = z - 1$ *to* 0 **do**
    Search solutions according to Eq. (6).
Return $\{d_{i,j}\}$, $J_0(s_{0,0})$.

$1, 0 \leq j < z$. Therefore, the expected reward at $s_{z+1,j}$ $(0 \leq j < z)$ can be calculated easily. The system may transit to state $s_{z,z}$, resulting in the special state `match`. State $s_{z,z}$ may transit to $s_{z+1,z}$ with a probability p. Thus, the expected reward at $s_{z,z}$ is $p(b + z \cdot d)$.

To stop the attack, the attacker will release all the sub-chain and the reward is determined by i and j. If $i > j$, the attacker gains $i \cdot d$. If $i < j$, the attacker gains nothing. If $i = j$, the attacker gains the expected reward $i \cdot p \cdot d$. As described, we construct the boundary condition of Adaptive DSA as Eq. (7) and Eq. (8).

$$J_n(s_{i,j}) = \begin{cases} 0, \text{ if } 0 \leq i \leq z-1, j = z; \\ p \cdot [b + (z+1) \cdot d] + q \cdot J_{z,j+1} - cost, \\ \quad \text{if } i = z, 0 \leq j \leq z-1; \\ p \cdot [b + z \cdot d], \text{ if } i = z, j = z. \end{cases} \tag{7}$$

$$J_{n+1}(s'_{i,j}) = \begin{cases} 0, & \text{if } 0 \leq i < j \leq z-1; \\ i \cdot d, & \text{if } 0 \leq j < i \leq z-1; \\ p \cdot i \cdot d, & \text{if } 0 \leq i = j \leq z-1. \end{cases} \tag{8}$$

According to Eq. (6), Eq. (7) and Eq. (8), we can solve the problem (3) using the backward recursion [15]. The pseudo code for the problem-solving under Adaptive DSA is shown in Algorithm 1.

4. Performance Evaluation

In this section, we first implement a discrete-time framework to simulate a Bitcoin blockchain. We then conduct numerical simulations to disclose the insights of the proposed analytical model for the new type of double-spending attacks.

4.1. Metrics

The expected reward of the attacker: The expected reward of the attacker during the entire double-spending attack is used as an index to evaluate the effectiveness of a specific double-spending attack strategy. The unit of expected reward is measured in the number of BTC coins. Recall that, Naive DSA is in fact the Adaptive DSA when all attack-decision points are valued 1. Thus, The expected reward of Naive DSA is equal to that of Adaptive DSA whose decision points are all set to 1. For Adaptive DSA, the expected reward is equal to $J_0(s_{0,0})$.

The minimum money in the target transaction (TX): The minimum amount of stolen money contained in the target transaction.

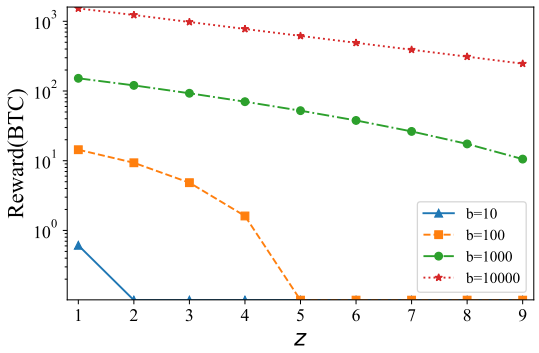

Figure 5. The attacker's expected reward v.s. varying z, while $p = 0.3$, and $d = 6.25$.

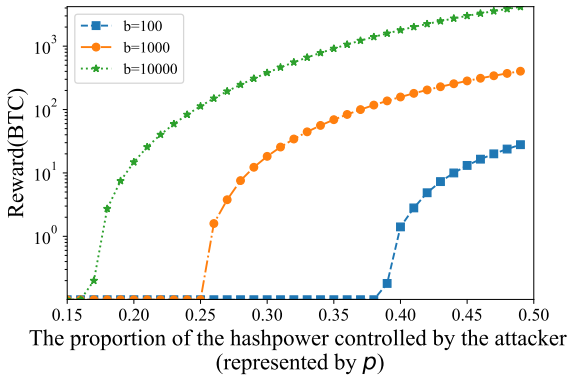

Figure 6. The attacker's expected reward v.s. varying p, while $z = 7$, and $d = 6.25$

4.2. Evaluation Results

4.2.1. Impact of The Number of Confirmation Blocks. Fig. 5 shows the impact of z on the expected reward of the attacker. We can observe that the expected reward of the attacker decreases exponentially with the increasing z. Once the expected reward is greater than zero, the attacker is likely to launch the attack and gain the attack reward. For a transaction valued 100 BTC coins (i.e., $b = 100$), the merchant should wait at least 4 blocks for the secure confirmation according to Naive DSA. However, if a transaction valued 1000 BTC coins (i.e., $b = 1000$), the merchant must wait more than 10 confirmation blocks to defend the Adaptive DSA. This result shows that the honest BTC users may overestimate the safety for a given transaction if they only defend against the Naive DSA strategy. Instead, the security level of BTC blockchain should be upgraded to

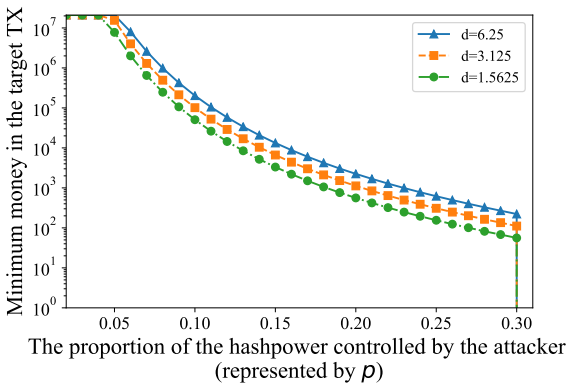

Figure 7. The minimum money contained in the target transaction (i.e., the *attack-threshold* of Adaptive DSA) v.s. varying p, while $z = 7$.

defend against the Adaptive DSA, which can alert the payee a safe number of confirmation blocks.

4.2.2. Impact of Hashpower Controlled by The Attacker. When the attacker's hashpower is fixed and other parameters remain unchanged, the minimum money contained in the target transaction that can make the expected reward positive is named the *attack threshold* of the double-spending attack under a given hashpower p. In this group of simulations, we evaluate the relationship between such the *attack threshold* and the proportion of attacker's hashpower p, under Adaptive DSA strategy.

Fig. 6 shows that the attacker will not launch a double-spending attack on any arbitrary transaction. Only when the expected reward is high can the attacker have the motivation to launch a double-spending attack. When the attacker has a certain fixed hashpower, the transaction with a higher transaction amount is more vulnerable to suffering from a double-spending attack.

Then, Fig. 7 presents the minimum amount of stolen money required to launch a profitable double-spending attack versus different values of p. We observe that when fixing $z = 7$, a smaller d makes the *attack threshold* lower. As the attacker's hashpower p increases, the *attack threshold* decreases quickly.

5. Conclusion

Based on the Naive DSA, we proposed a new advanced version of DSA model, i.e., the Adaptive DSA. Based on the Stochastic Dynamic Programming framework, we developed rigorous analysis for such Adaptive DSA. To pursue the maximized profit of the attacker, we then devised the SDP-based algorithm to calculate the decision matrix for the attack strategy. Through numerical simulations, we studied the correlations between different parameters and the expected reward of the attacker. Several insights behind those double-spending attack strategies have been revealed. We think our study in this paper can alert the Bitcoin users that the security level of Bitcoin blockchain is not that high as expected. The proposed analytical model can stimulate both researchers and engineers to evaluate the threat of the new type of double-spending attacks, and then develop the corresponding defence strategies.

本文提出的攻击者最大化收益的攻击决策分析模型有利于我们设计相应的对策
最好不要出现大额交易。或者对于大额交易设置一些限制，避免被攻击或者降
低被攻击的概率

6. Acknowledgement

This work described in this paper was supported by the Key-Area Research and Development Program of Guangdong Province (No.2019B020214006), the National Natural Science Foundation of China (No.62032025, No.61902445), and the Guangdong Basic and Applied Basic Research Foundation (No.2019A1515011798).

References

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

[2] Bianews. Liao xiang's response to btg's shuanghua attack: upgrading the mining algorithm is being implemented to completely eliminate 51% attacks technically. [Online]. Available: https://www.sohu.com/a/232781696_115060

[3] P. strange Jun. Etc was attacked by 51% and stolen about $1.1 million. [Online]. Available: https://cloud.tencent.com/developer/news/383561

[4] TechFlow. Aeternity is attacked by 51%. how can ae defend against crisis in advance? [Online]. Available: https://www.sohu.com/na/437657586_120697730

[5] J. Wu, J. Liu, W. Chen, H. Huang, Z. Zheng, and Y. Zhang, "Detecting mixing services via mining bitcoin transaction network with hybrid motifs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[6] H. Huang, W. Kong, S. Zhou, Z. Zheng, and S. Guo, "A survey of state-of-the-art on blockchains: Theories, modelings, and tools," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–42, 2021.

[7] W. Chen, J. Wu, Z. Zheng, C. Chen, and Y. Zhou, "Market manipulation of bitcoin: Evidence from mining the mt. gox transaction network," in *IEEE Conference on Computer Communications (INFOCOM)*, 2019, pp. 964–972.

[8] C. Pinzón and C. Rocha, "Double-spend attack models with time advantange for bitcoin," *Electronic Notes in Theoretical Computer Science*, vol. 329, pp. 79–103, 2016.

[9] K. Liao and J. Katz, "Incentivizing double-spend collusion in bitcoin," in *Financial Cryptography Bitcoin Workshop*, 2017.

[10] B. Biais, C. Bisiere, M. Bouvard, and C. Casamatta, "The blockchain folk theorem," *The Review of Financial Studies*, vol. 32, no. 5, pp. 1662–1715, 2019.

[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 *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 3–16.

[12] G. Ramezan, C. Leung, and Z. J. Wang, "A strong adaptive, strategic double-spending attack on blockchains," in *IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018.

[13] C. Grunspan and R. Pérez-Marco, "On profitability of nakamoto double spend," *arXiv preprint arXiv:1912.06412*, 2019.

[14] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016, pp. 305–320.

[15] S. M. Ross, *Introduction to stochastic dynamic programming*. Academic press, 2014.