Fact and Fiction: Challenging the honest majority assumption of permissionless blockchains

Runchao Han
Monash University and
CSIRO-Data61
runchao.han@monash.edu

Zhimei Sui
Monash University
zhimei.sui1@monash.edu

Jiangshan Yu
Monash University
jiangshan.yu@monash.edu

Joseph Liu
Monash University
joseph.liu@monash.edu

Shiping Chen
CSIRO-Data61
shiping.chen@data61.csiro.au

ABSTRACT

Honest majority is the key security assumption of Proof-of-Work (PoW) based blockchains. However, the recent 51% attacks render this assumption unrealistic in practice.

In this paper, we challenge this assumption against rational participants in the PoW-based blockchains in reality. In particular, we show that *the current incentive mechanism may encourage rational participants to launch 51% attacks* in two cases. In the first case, we consider a profit-driven miner of a blockchain attacking another blockchain, where the two blockchains adapt compatible mining algorithms and the victim is weaker w.r.t. the total mining power in the system. In the second case, we consider a profit-driven miner launching 51% attacks by renting mining power from cloud mining services.

We formally model such profit-driven behaviours as a series of actions through a Markov Decision Process. Our results show that, for most mainstream PoW-based blockchains, 51% attacks are feasible and profitable, so profit-driven miners are incentivised to launch 51% attacks to gain extra profit. In addition, we leverage our model to investigate the recent 51% attack on Ethereum Classic (on 07/01/2019), which is suspected to be an incident of 51% attacks. We provide insights on the attacker strategy and expected revenue, and show that the attacker's strategy is near-optimal.

1 INTRODUCTION

Proof-of-work (PoW) based consensus was first introduced by Bitcoin [1], to allow distributed nodes agreeing on the same global state of the system. In Bitcoin, all transactions are recorded as a chain of blocks. Anyone can create a block of transactions, and append it into the Bitcoin blockchain as a unique successor of the last block. To create a block, one needs to solve a crypto puzzle which is computationally hard. In particular, the puzzle solver, called a miner, needs to find a random nonce to make the hash value of the block smaller than a target value.

Different miners may create different valid blocks as successors of the same block, which creates a fork in the system. To resolve a fork, miners only accept the longest branch. However, a branch that is currently longer may be overtaken by the other branch, and all transactions in the currently longer branch will be deemed invalid. This creates an opportunity for the attacker to spend a coin in a branch, and later on creates another longer branch to erase this transaction. This is called "double spending attack". Intuitively, to launch a double-spending attack, an attacker should have enough mining power to create a branch that is longer than the current one. This requires the attacker to control a majority of mining power in the network. An attacker with a majority of mining power to double spend is known as "51% attacks".

Key assumption. PoW-based consensus relies on the "honest majority" assumption: the majority of the mining power is honest. If this assumption does not hold, then 51% attacks would be possible. Such a security guarantee depends on the total mining power in the system: with more mining power in the system, controlling the majority of mining power will be more difficult, which leads to a stronger security guarantee.

To gather more mining power to the system, Bitcoin encourages miners to join the system via an incentive mechanism. In Bitcoin, miners who create a block successfully will be rewarded for their contribution. The mining reward includes a pre-defined number of bitcoins and transaction fees of all transactions contained in the block. In this way, miners are encouraged to contribute their mining power, making the system more secure. This incentive mechanism has been employed by almost all mainstream blockchains.

Fact v.s. Fiction. Ideally, there is only one blockchain in the entire world, and all potential miners will participate in this blockchain. In this way, controlling 51% mining power can be extremely difficult. However, if there is another blockchain, the total available mining power in the world will be split into two blockchains, which can reduce the security guarantee of both blockchains.

Unfortunately, there are multiple PoW-based blockchains in the real world, so no PoW-based blockchain can enjoy the ideal security guarantee. To date (09/01/2020), there are over 2,400 cryptocurrencies[1], and several 51% attacks have been identified, resulting in the loss of more than $25.1 million (as shown in Figure 1).

Miners are rational. In 2005, three years before the birth of Bitcoin, BAR (Byzantine, Altruistic, Rational) model [3] was considered in the context of fault tolerance for cooperative services. In this model, a participant can be Byzantine (i.e., behaving arbitrarily), altruistic (i.e., honest), or rational (behaving for maximising his profit). Participants in PoW-based consensus should be considered rational, as PoW-based consensus encourages miners to join the system using mining reward.

[1] https://coinmarketcap.com/all/views/all/

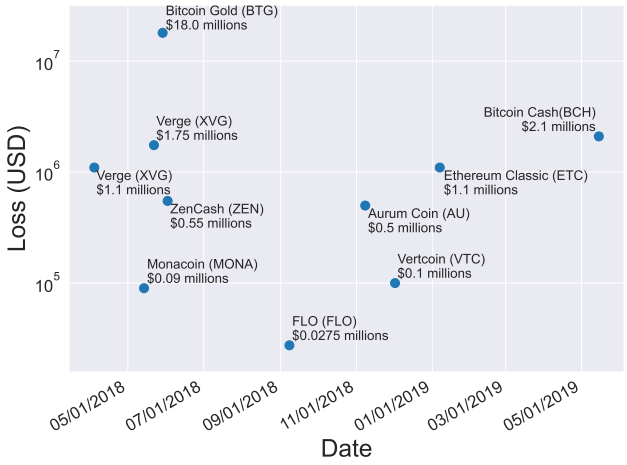


Figure 1: 51% Attacks in 2018-2020 [2].We omit the 51% attack on Litecoin Cash on June 4, 2018 as the loss is unknown.

1.1 Our contributions

We challenge the honest majority assumption against two cases. In cryptocurrency context, a 51% attack leads to double-spending, where the adversary can spend his coins more than once. Under the assumption that miners are rational, if a miner can gain more profit through 51% attacks, his rational behaviour will no longer be honestly mining, but launching 51% attacks.

We consider two scenarios leading to such 51% attacks. The first approach is using mining power on a stronger blockchain to launching 51% attacks on a weaker blockchain (which we call *mining power migration attack*). As aforementioned, total mining power in different blockchains could be very different, and such *mining power migration attack* can be feasible. The second approach is the previously known *cloud mining attack* [4], where an adversary rents mining power from the cloud mining services (such as Nicehash [5]) to launch 51% attacks.

Once launching 51% attacks profits more than honestly mining, rational miners will be incentivised to behave Byzantine and break the blockchain, and the victim blockchain will be compromised. In other words, permissionless systems assuming rational participants can sometimes incentivise participants to break itself. Thus, the assumption of rational participants in permissionless settings is unreliable in the real world.

We demonstrate that, for most mainstream PoW-based blockchains, profit-driven miners are incentivised to launch 51% attack to gain extra profit. To simulate the two attack cases, we formalise them using the extended version of the Markov Decision Process (MDP)-based model from Gervais et al. [6], and we name the model 51-MDP. 51-MDP takes parameters of blockchains and the adversary as input, and outputs the cost and reward of launching a 51% attack. To evaluate the feasibility and profitability of 51% attacks, we apply 51-MDP to real-world blockchains. The result (in Section 5) shows that it is not challenging to launch both types of 51% attacks successfully. For example, a miner with 12.5% mining power in BTC can gain approximately 6% ($18,946.5) extra profit (than honest mining) by double-spending a transaction of 3000 BCH (equivalent to $378,930) on BitcoinCash. This required mining power is in fact not difficult to achieve – at the time of writing, F2Pool.com controls 17.7% mining power in Bitcoin[2].

We investigate the 51% attack on Ethereum Classic in 2019 [7] using 51-MDP. We leverage 51-MDP to gain extra insights on the 51% attack, including explanations on the attacker's strategy and the estimated revenue of the attack. On 07/01/2019, an anonymous attacker launched a series of 51% attacks, and double-spent transactions of more than $1.1 million on a cryptocurrency exchange Gate.io [7]. Interestingly, the attacker returned ETC coins equivalent to $100,000 to Gate.io later [8]. We first analyse the attacker's strategy according to the pattern of double-spent transactions, and find that the attacker's strategy can maximise and stabilise his revenue. Then, we input double-spent transactions to our 51-MDP model, and reverse-engineered the attacker's revenue. The result shows that, the attacker is expected to earn $84773.40, which is close to $100,000. This indicates the attacker was likely to launch 51% attacks in the fine-grained way described in our paper.

1.2 Paper organisation

This paper is structured as follows. §2 provides a discussion on the related work. §3 presents our system model and formalisation on the two types of 51% attacks. The model is evaluated in §4, and the feasibility of the attacks is analysed in §5. We investigate the recent 51% attack on EthereumClassic in §6. We discuss potential remedies of our attacks in §7 and conclude the paper in §8.

In addition, Appendix A summarises other related work; Appendix B provides a study on the optimal strategy of a BTC miner to launch these two attacks on BCH; and Appendix C presents all experimental data used in this paper.

2 RELATED WORK

A growing number of attacks leveraging incentives in blockchains have recently been identified. The most related work to this paper is fickle mining [9] and bribery attacks [4]. Other related work can be found in Appendix A.

Fickle mining [9] observed that a miner may gain extra profit by performing honest mining on two blockchains (e.g. BTC and BCH), and proposed a game to model and analyse such behavior. Spiegelman et al. [10] proposed a more generic game to model and analyse a more complex case, where adaptive miners may perform different mining strategies to do honestly mining among multiple blockchains.

Bribery attacks [4] identified several ways for an attacker to bribe other miners to purchase their mining power for a short duration to launch 51% attacks. New ways [11, 12, 13] to bribe miners through higher transaction fees have also been explored.

Our work explores the impact of incentive from a different angle. In particular, we show that if miners are profit-driven, then they may turn to evil to gain extra profit. If this is true, then it indicates that rather than attracting honest mining power to increase the security of the system, the current incentive mechanism may incentivise profit-driven miners to launch attacks. We consider two scenarios

[2]https://www.blockchain.com/pools. Data fetched on 09/01/2020.

where miners may behave maliciously to gain extra profit, namely the mining power migration attack and the cloud mining attack.

Similar to the fickle mining, our mining power migration attack also considers a miner moving between blockchains. However, in contrast to the existing work where the miner only performs honest mining, we consider the miner may turn to evil and launch 51% attacks when possible. A more detailed discussion is provided in Section 5.

3 FORMALISATION

In this section, we provide a formalisation, called 51-MDP, on our considered two approaches of launching 51% attacks. 51-MDP takes our defined blockchain parameters as input, and outputs an optimal attack strategy with expected revenue of this attack.

The key ideas of the two attacks are simple (but quite effective and practical as shown in Section 5): The increasing number of proof-of-work based blockchains deployed in the world may reduce the total amount of available mining power in each blockchain. This reduces the difficulty to launch 51% attacks on a chosen blockchain. In this way, the adversary can make use of external mining power to launch 51% attacks on a PoW-based blockchain, where the external mining power comes from either other blockchains or cloud mining services.

3.1 System and threat model

We consider profit-driven miners, and potentially multiple blockchains such that their mining power is compatible with each other's mining algorithm. For simplicity, our model only considers two blockchains with the same mining algorithm. We call the blockchain with more total mining power a *stronger blockchain*, and the other one a *weaker blockchain*. As the attack (on the weaker blockchain) will be completed within a relatively short time period, our model assumes that during the attack, the difficulty parameter, total amount of honest mining power, and block mining reward do not change.

For *mining power migration attacks*, we assume that the rational miner already has a considerable portion (much less than 50%) of mining power on the *stronger blockchain*. For *cloud mining attacks*, we assume that the rentable cloud mining power from cloud mining services such as NiceHash is compatible with the victim blockchain. In addition, we assume that the adversary has sufficient money to rent cloud mining power according to the mining strategy. Later in Section 5, we will analyse the soundness of such an environment.

3.2 Notations

Table 1 provides a summary of notations. Let BC_1 and BC_2 be the stronger blockchain and the weaker blockchain in terms of the mining power, respectively. Let pr be the price of renting a unit of mining power (e.g. hash per second) for a time unit. Let D_1 and D_2 be the difficulties, R_1 and R_2 be the mining rewards of BC_1 and BC_2, respectively. We have $D_1 > D_2$ and $R_1 > R_2$. We define the fractions of the difficulties and mining rewards as: $d = \frac{D_1}{D_2}$, $r = \frac{R_1}{R_2}$.

For *mining power migration attacks*, we define mining-related parameters for two blockchains. Let $H_{h,1}, H_{h,2}$ be the honest mining power, and $H_{a,1}, H_{a,2}$ be the adversary's mining power of BC_1 and BC_2, respectively. Let $H_1 = H_{h,1} + H_{a,1}$ and $H_2 = H_{h,2} + H_{a,2}$ be the total mining powers on BC_1 and BC_2, respectively. Note that $H_1 > H_2$ according to the security model. Let $h_1 = \frac{H_a}{H_{h,1}}$ and $h_2 = \frac{H_a}{H_{h,2}}$ be the fractions of the adversary's mining power towards the honest mining power on BC_1 and BC_2. Let $\beta = \frac{H_{a,2}}{H_a}$ be the fraction of migrated mining power by the adversary.

Table 1: Notations of parameters in 51-MDP

Symbol	Definition	Comment
D_1	The difficulty of BC_1	
D_2	The difficulty of BC_2	
d	The fraction of BC_1's difficulty towards BC_2's difficulty	$d = \frac{D_1}{D_2}$
$H_{h,1}, H_{a,1}$	The honest and adversary's mining power on BC_1	
$H_{h,2}, H_{a,2}$	The honest and adversary's mining power on BC_2	
H_a, H_h	The total honest and adversary's mining power	$H_a = H_{a,1} + H_{a,2}$, $H_h = H_{h,1} + H_{h,2}$
h_1	The fraction of the adversary's mining power towards the honest mining power on BC_1	$h_1 = \frac{H_a}{H_{h,1}}$
h_2	The fraction of the adversary's mining power towards the honest mining power on BC_2	$h_2 = \frac{H_a}{H_{h,2}}$
R_1	The mining reward of a block of BC_1	
R_2	The mining reward of a block of BC_2	
r	The fraction of BC_1's mining reward of a block towards BC_2's	$r = \frac{R_1}{R_2}$
v_{tx}	The amount of the attacking transactions	
γ	The propagation parameter of the adversary	
pr	Renting price of a mining algorithm	
β	The fraction of migrated mining power by the adversary	
N_c	The number of blocks required to confirm a transaction	

For *cloud mining attacks*, since the mining power is not coming from another blockchain, we only consider the target blockchain BC_2. Let $H_{h,2}$ be the honest mining power, and H_a be the rentable mining power from the cloud mining service. Let $h_2 = \frac{H_a}{H_{h,2}}$ be the fraction of rented mining power out of rentable mining power. Let $\beta = \frac{H_{a,2}}{H_a}$ be the fraction of rented mining power towards the rentable mining power. Let pr be the price of renting mining power (in $\$/h/s$).

Let $\gamma \in [0, 1]$ be the propagation parameter of the adversary – we use it to denote the ratio of honest miners that receive the adversary's block first and mine on this block. Let N_c be the required number of blocks for the blockchain network to confirm a transaction.

3.3 The 51-MDP model

For a profit-driven miner, the willingness of launching an attack is largely influenced by its expected net revenue. To quantitatively analyse the optimal strategy of launching 51% attacks and its expected revenue, we develop a Markov Decision Process (MDP), called 51-MDP. It describes the attacks as a series of actions performed by an adversary. At any time, the adversary lies in a state, and can perform an action, which transits his state to another state by a certain probability. For each state transition, the adversary may get some reward or penalty.

Table 2: State transitions and reward matrices of 51-MDP. Notations are summarised in Table 1.

State × Action	Resulting State	Probability	Reward				Condition
			$\mathbb{R}'_{migration}$	$\mathbb{R}'_{cloudmining}$	$\mathbb{R}_{mine}$	$\mathbb{R}_{tx}$	
$(l_h, l_a, \beta, fork)$, ADOPT	(0, 0, β, ir)	1	0	0	0	0	$l_h > l_a \geq N_c$
$(l_h, l_a, \beta, fork)$, OVERRIDE	(0, 0, β, ir)	1	0	0	$l_a R_2$	v_{tx}	$l_a > l_h \geq N_c$
$(l_h, l_a, \beta, fork)$, WAIT	(l_h, l_a +1, β, p)	$\frac{\beta h_2}{\beta h_2+1}$	$\frac{-\beta h_2 R_1}{d(1+\beta h_2)}$	$\frac{-\beta h_2 D_2 pr}{1+\beta h_2}$	0	0	$l_h < N_c$
	(l_h+1, l_a, β, p)	$\frac{1}{\beta h_2+1}$	$\frac{-\beta h_2 R_1}{d(1+\beta h_2)}$	$\frac{-\beta h_2 D_2 pr}{1+\beta h_2}$	0	0	$l_h < N_c$
$(l_h, l_a, \beta, fork)$, WAIT_INC	(l_h, l_a+1, β +0.2, p)	$\frac{(\beta+0.2)h_2}{(\beta+0.2)h_2+1}$	$\frac{-(\beta+0.2)h_2 R_1}{d(1+(\beta+0.2)h_2)}$	$\frac{-(\beta+0.2)h_2 D_2 pr}{1+(\beta+0.2)h_2}$	0	0	$l_h < N_c$
	(l_h+1, l_a, β +0.2, p)	$\frac{1}{(\beta+0.2)h_2+1}$	$\frac{-(\beta+0.2)h_2 R_1}{d(1+(\beta,+0.2)h_2)}$	$\frac{-(\beta+0.2)h_2 D_2 pr}{1+(\beta+0.2)h_2}$	0	0	$l_h < N_c$
$(l_h, l_a, \beta, fork)$, WAIT_DEC	(l_h, l_a+1, β -0.2, p)	$\frac{(\beta-0.2)h_2}{(\beta-0.2)h_2+1}$	$\frac{-(\beta-0.2)h_2 R_1}{d(1+(\beta-0.2)h_2)}$	$\frac{-(\beta-0.2)h_2 D_2 pr}{1+(\beta-0.2)h_2}$	0	0	$l_h < N_c$
	(l_h+1, l_a, β -0.2, p)	$\frac{1}{(\beta-0.2)h_2+1}$	$\frac{-(\beta-0.2)h_2 R_1}{d(1+(\beta-0.2)h_2)}$	$\frac{-(\beta-0.2)h_2 D_2 pr}{1+(\beta-0.2)h_2}$	0	0	$l_h < N_c$
$(l_h, l_a, \beta, fork)$, MATCH	(l_h, l_a +1, β , ir)	$\frac{\beta h_2+\gamma}{\beta h_2+1}$	$\frac{-\beta h_2 R_1}{d(1+\beta h_2)}$	$\frac{-\beta h_2 D_2 pr}{1+\beta h_2}$	$\frac{(l_a+1)R_2\beta h_2}{\beta h_2+\gamma}$	v_{tx}	$l_h = l_a \geq N_c$
	(l_h +1, l_a, β , r)	$\frac{1-\gamma}{\beta h_2+1}$	$\frac{-\beta h_2 R_1}{d(1+\beta h_2)}$	$\frac{-\beta h_2 D_2 pr}{1+\beta h_2}$	0	0	$l_h = l_a \geq N_c$
$(l_h, l_a, \beta, fork)$, MATCH_INC	(l_h, l_a +1, β +0.2, ir)	$\frac{(\beta+0.2)h_2+\gamma}{(\beta+0.2)h_2+1}$	$\frac{-(\beta+0.2)h_2 R_1}{d(1+(\beta+0.2)h_2)}$	$\frac{-(\beta+0.2)h_2 D_2 pr}{1+(\beta+0.2)h_2}$	$\frac{(l_a+1)R_2(\beta+0.2)h_2}{(\beta+0.2)h_2+\gamma}$	v_{tx}	$l_h = l_a \geq N_c$
	(l_h + 1, l_a, β +0.2 , r)	$\frac{1-\gamma}{(\beta+0.2)h_2+1}$	$\frac{-(\beta+0.2)h_2 R_1}{d(1+(\beta+0.2)h_2)}$	$\frac{-(\beta+0.2)h_2 D_2 pr}{1+(\beta+0.2)h_2}$	0	0	$l_h = l_a \geq N_c$
$(l_h, l_a, \beta, fork)$, MATCH_DEC	(l_h, l_a +1, β -0.2, ir)	$\frac{(\beta-0.2)h_2+\gamma}{(\beta-0.2)h_2+1}$	$\frac{-(\beta-0.2)h_2 R_1}{d(1+(\beta-0.2)h_2)}$	$\frac{-(\beta-0.2)h_2 D_2 pr}{1+(\beta-0.2)h_2}$	$\frac{(l_a+1)R_2(\beta-0.2)h_2}{(\beta-0.2)h_2+\gamma}$	v_{tx}	$l_h = l_a \geq N_c$
	(l_h +1, l_a, β -0.2 , r)	$\frac{1-\gamma}{(\beta-0.2)h_2+1}$	$\frac{-(\beta-0.2)h_2 R_1}{d(1+(\beta-0.2)h_2)}$	$\frac{-(\beta-0.2)h_2 D_2 pr}{1+(\beta-0.2)h_2}$	0	0	$l_h = l_a \geq N_c$

Formally, our 51-MDP model is a four-element tuple $M := (S, A, P, R)$ where:

- S is the state space containing all possible states of an adversary.
- A is the action space containing all possible actions performed by an adversary.
- P is the stochastic transition matrix presenting the probabilities of all state transitions.
- R is the reward matrix presenting the rewards of all state transitions.

We detail each element of M below, and present an overview on the state transitions and reward matrices of 51-MDP in Table 2.

3.3.1 **The State Space** S**.** The state space S is defined as a tuple $S := (l_h, l_a, \beta, fork)$, where l_h and l_a are the length of the honest and the adversary's branches on BC_2, respectively. Eventually, one of these two branches will be accepted by the network. Let β be the ratio of mining power at BC_2 allocated by the adversary; and $fork$ be the state of the adversary's fork. For simplicity, we choose β from (0.0, 0.2, 0.4, 0.6, 0.8, 1.0). We leave the study of more fine-grained attacks as future work. The state $fork$ of the adversary's fork has three values, defined as follows:

Relevant ($fork = r$) means the adversary's fork is published but the honest blockchain is confirmed by the network. This indicates that the attack is unsuccessful at present. (Note that the adversary can keep trying and may succeed in the future.)

Irrelevant ($fork = ir$) means the adversary's fork is published and confirmed in network. This indicates a successful attack.

Private ($fork = p$) means the adversary's fork is private and only the adversary is mining on it. This indicates that an attack is in process.

3.3.2 **The Action Space** A**.** An adversary can perform the following actions:

ADOPT The adversary accepts the honest blockchain and discards his fork, which means the adversary aborts his attack.

OVERRIDE The adversary publishes his fork (which is longer than the honest one). Consequently, the honest blockchain is overridden, and the payment transaction from the adversary is successfully reverted.

MATCH The adversary publishes his fork with the same length as the honest blockchain. This action has three variants **MATCH**, **MATCH_INC**, and **MATCH_DEC**, where **_INC** and **_DEC** represent the increase and decrease of malicious mining power ratio β, respectively.

WAIT The adversary keeps mining on his fork. This action can be performed in two scenarios. One is $l_h < N_c$, indicating that the merchant is still waiting for the payment confirmation. Another one is that after the adversary publishes his blockchain by **MATCH**, $l_a \leq l_h$ still holds. In addition, there are three types in this action **WAIT**, **WAIT_INC**, **WAIT_DEC**, which represent the adversary's mining power adjustment, similar to that of **MATCH**.

3.3.3 **The State Transition Matrix** P**.** The State Transition Matrix P (Table 2) describes all state transition possibilities. It is defined as a 3-dimensional matrix $S \times A \times S : Pr(s, a \Rightarrow s')$, where the participant in the state s does the action a to transit his state to s' with the probability $Pr(s, a \Rightarrow s')$. Here $s, s' \in S$ and $a \in A$.

In MDP, an action can trigger a state s to transit to one of its possible resulting state $s'_1, s'_2, \cdots, s'_n$ with probability $Pr(s, a \Rightarrow s'_i)$, such that $\sum_{i=1}^{n} Pr(s, a \Rightarrow s'_i) = 1$.

As aforementioned, in 51% attacks, the state transition is invoked by a new block (during **WAIT**), a blockchain fork selection (by **OVERRIDE** or **MATCH**) or quitting the attack (by **ADOPT**). While the results of **OVERRIDE** and **ADOPT** are deterministic (i.e., succeeding and giving up, respectively), we discuss state transitions triggered by **WAIT**-style and **MATCH**-style actions.

Wait for attacks (WAIT[, _INC, _DEC]). The adversary is mining his fork alone. The next state update is triggered by a newly created block either by the honest network or by the adversary, with a probability directly associated to their mining power on BC_2, namely $H_{a,2} = \beta H_a$ of the adversary and $H_{h,2}$ of the honest network.

Therefore, the probability that the adversary gets the next block ($l_a \rightarrow l_a + 1$) is

$$\begin{aligned} P(l_a \rightarrow l_a + 1) &= \frac{H_{a,2}}{H_{a,2} + H_{h,2}} \\ &= \frac{\beta H_a}{\beta H_a + H_{h,2}} = \frac{\beta h_2}{\beta h_2 + 1} \end{aligned} \tag{1}$$

And vice versa for $l_h \rightarrow l_h + 1$.

$$P(l_h \rightarrow l_h + 1) = 1 - P(l_a \rightarrow l_a + 1) = \frac{1}{\beta h_2 + 1} \tag{2}$$

Attack by MATCH[, _ENC, _DEC]. The adversary tries to overtake the honest branch once $l_a \geq N_c$ and $l_a = l_h$. Besides the mining power owned by the adversary, the eclipsed mining power of $\gamma H_{h,1}$ mines on the adversary's blockchain after a **MATCH** attempt. Therefore, the possibility of $l_a \rightarrow l_a + 1$ becomes

$$\begin{aligned} P(l_a \rightarrow l_a + 1) &= \frac{H_{a,2} + H_{eclipsed}}{H_{a,2} + H_{h,2}} \\ &= \frac{\beta H_a + \gamma H_{a,2}}{\beta H_a + H_{h,2}} = \frac{\beta h_2 + \gamma}{\beta h_2 + 1} \end{aligned} \tag{3}$$

And vice versa for $l_h \rightarrow l_h + 1$.

$$P(l_h \rightarrow l_h + 1) = 1 - P(l_a \rightarrow l_a + 1) = \frac{1 - \gamma}{\beta h_2 + 1} \tag{4}$$

3.3.4 **The Reward Matrix** R**.** The Reward Matrix R is defined as $S \times A \times S : Re(s, a \Rightarrow s')$, where the participant in the state s transits to a new state s' with the reward $Re(s, a \Rightarrow s')$. Here $s, s' \in S$ and $a \in A$. The reward from a double-spending attack contains two parts: the block reward (including transaction fees) of the published longer blockchain and the double-spending transaction. To calculate the net reward, we also need to consider the cost of launching an attack. Here, we define the reward $Re(s, a \Rightarrow s')$ as a tuple ($\mathbb{R}'$, $\mathbb{R}_{mine}$, $\mathbb{R}_{tx}$) to fit into the MDP model, where $\mathbb{R}'$ represents the cost of launching a double-spending attack, $\mathbb{R}_{mine}$ represents the reward from the mined blocks including transaction fees, and $\mathbb{R}_{tx}$ represents the reward from the double-spent transaction.

With a state transition $S \times A \rightarrow S'$, the adversary will get some reward, which can be of a positive or negative value. The state transition matrix R is a 3-dimension matrix ($S \times A \times S'$), where S, A and S' are the same as P, but the value of this matrix is the reward of the corresponding state transition.

The reward of a 51% attack should be the net revenue i.e., the gross profit minus the cost. The gross profit consists of the revenue of mining $\mathbb{R}_{mine}$ and the revenue from the double-spent transaction $\mathbb{R}_{tx}$. The cost is the attack cost $\mathbb{R}'$. We define $\mathbb{R}_{mine}$, $\mathbb{R}_{tx}$ and $\mathbb{R}'$ as follows.

Mining reward $\mathbb{R}_{mine}$*.* The adversary can receive the block reward $\mathbb{R}_{mine}$ on BC_2 only when his fork is published and accepted by the honest network. Therefore, only **OVERRIDE** and the winning scenarios of **MATCH**-style actions have a positive reward, while $\mathbb{R}_{mine} = 0$ under other scenarios.

When performing **OVERRIDE**, the adversary's blockchain of length l_a is directly accepted, so $\mathbb{R}_{mine} = l_a R_2$. When performing **MATCH**-style actions, the adversary needs to get the next block so that his blockchain overrides the honest one. Therefore, $\mathbb{R}_{mine} = (l_a + 1)R_2$ holds for the winning scenarios.

Reward from the double-spent transaction $\mathbb{R}_{tx}$*.* Similar with $\mathbb{R}_{mine}$, the adversary receuves the double-spent money only when it successfully overrides the honest blockchain. Therefore, $\mathbb{R}_{tx}$ equals to the transaction amount v_{tx} for **OVERRIDE** and the winning scenarios of **MATCH**-style actions, while $\mathbb{R}_{tx} = 0$ for other scenarios.

Attack cost $\mathbb{R}'$*.* To calculate net revenue, we also need to consider the cost of launching attacks. We use $\mathbb{R}'_{migration}$ and $\mathbb{R}'_{cloudmining}$ to denote the cost of launching the *mining power migration attack* and the *cloud mining attack*, respectively. Compared to honest mining on BC_1, the cost $\mathbb{R}'_{migration}$ of mining power migration is mainly from the loss of block rewards from BC_1 due to the migrated mining power. Consequently, the cost can be computed as hashrate · time · difficulty · R_1, which is the estimated mined block multiplies the block reward of BC_1. For **ADOPT** and **OVERRIDE** actions, the time of finishing a state transition is negligible. Meanwhile, for **WAIT**-style and **MATCH**-style actions, the state transition is triggered by mining a new block, so the time it takes is depending on the difficulty of mining a block. Therefore, the $\mathbb{R}'_{migration}$ under **WAIT**-style and **MATCH**-style actions can be computed as follows:

$$\begin{aligned} \mathbb{R}'_{migration}(l_a \rightarrow l_a + 1) &= \mathbb{R}'_{migration}(l_h \rightarrow l_h + 1) \\ &= \text{hashrate} \cdot R_1 \cdot \text{time} \cdot \frac{1}{\text{difficulty}} \\ &= -\beta H_a \cdot R_1 \cdot \frac{D_2}{H_{h,2} + \beta H_a} \cdot \frac{1}{D_1} \\ &= \frac{-\beta h_2 R_1}{d(1 + \beta h_2)} \end{aligned} \tag{5}$$

When launching *cloud mining attacks*, the cost is from renting the cloud mining power. The cloud mining power is priced as $\$/(h/s)/s$, which means the price of renting a unit of mining power for a time unit. For simplicity, we adopt the simplified expression $\$/h$ in the following paper. We denote the cloud mining power price as pr. Similar with the *mining power migration attack*, only **WAIT**-style and **MATCH**-style actions take a non-negligible time period.

Therefore, either the cost of **ADOPT** or **OVERRIDE** is 0, and the cost of **WAIT**-style and **MATCH**-style actions is computed as

$$\begin{aligned} R_{BC_1}(l_a \rightarrow l_a + 1) &= R_{BC_1}(l_h \rightarrow l_h + 1) \\ &= \text{hashrate} \cdot \text{price} \cdot \text{time} \\ &= -\beta H_a \cdot Pr \cdot \frac{D_2}{H_{h,2} + \beta H_a} \\ &= \frac{-\beta h_2 D_2 Pr}{1 + \beta h_2} \end{aligned} \tag{6}$$

4 EVALUATION

In this section, we present our implemented 51-MDP model, and evaluate the impact of each parameter on the two 51% attacks using 51-MDP. The evaluation reveals the most important aspects on the profitability of 51% attacks. By combining this evaluation and public blockchain data, the attackers can find the most feasible targets to attack, and the defenders can be aware of the potential threats in advance.

4.1 Experimental methodology

We implement our 51-MDP model using Python 2.7 and the *pymdptoolbox* library [14]. All experiments run on a MacBook Pro with a 2.2 GHz Intel Core i7 Processor, a 16 GB DDR4 RAM and 256 SSD storage disk.

The 51-MDP model is infinite due to the unbounded l_a and l_h. In order to implement 51-MDP, we convert it into the finite one by giving an upper bound $limit = 10$ for l_a and l_h. We apply the *ValueIteration* algorithm [15] with a discount value of 0.9 and an epsilon value of 0.1 in our MDP-based model. We apply this discount value in order to encourage the adversary to finish the attack in a short time. Theoretically, the reward of 51% attacks will not be discounted with more steps. In practice, the longer time a 51% attack takes, the more risk it will have. For example, shifting mining power to the victim blockchain might be detected by threat intelligence services. Such risk is hard to quantify, so we use a discount factor of 0.9 to model it.

We categorise parameters in 51-MDP to five types according to their related aspects: **1)Mining status** which includes two mining difficulties (D_1 and D_2) and two ratios of adversary's mining power (h_1 and h_2); **2)Incentive** which includes mining reward (R_1 and R_2) and the adversary's transaction amount v_{tx}; **3)Adversary's network condition** which includes the propagation parameter γ of the adversary; **4)Vigilance of the merchant** which includes the number N_c of required block confirmations; and **5)Mining power price** which includes pr only.

4.2 Evaluation results

We evaluate the impact of each aspect of the net revenue of an adversary launching both types of 51% attacks. Here, net revenue is attacker's total revenue deducting its attack cost, such as the the renting cost from cloud service and the rewards attacker may get from BC_1. Since both attacks have common parameters D_2, h_2, R_2, v_{tx}, γ and N_c, we evaluate them on *mining power migration attack* only to avoid the repetition.

Mining status. Figure 2a shows the impact of mining-related parameters on the adversary's net revenue. We observe that the net revenue increases monotonically with D_2 decreasing and h_2 increasing.

Mining difficulty variation reflects the fluctuation of network mining power. When D_2 decreases, network mining power decreases, then mining on BC_2 will be easier. Also, launching a 51% attack will be in a lower cost and easier to succeed, which encourages both types of our attacks on BC_2. By these observations, attackers prefer to invest more computing power to BC_2, then h_2 increases by migrating attacker's mining power from other blockchain or renting from cloud services. Therefore, both decreasing D_2 and increasing h_2 incentivise 51% attacks on BC_2.

Incentive-related parameters. Figure 2b shows the impact of incentive-related parameters on the net revenue. We observe that increasing R_2 and v_{tx} leads the adversary to profit more.

When R_2 increases, mining BC_2 will be more profitable, and 51% attacks on BC_2 will also be more profitable. This encourages both types of 51% attacks on BC_2. The 51% attack generates v_{tx} out of thin air, so v_{tx} is the direct revenue of the 51% attack, and increasing v_{tx} directly increases the net revenue. Therefore, both increasing R_2 and v_{tx} incentivise 51% attacks on BC_2.

Adversary's network condition. Figure 2c shows the impact of γ on the relative revenue. In particular, we can see that the relative reward increases slightly with γ increasing. Interestingly, when the attacker's propagation parameter $\gamma = 0.7$, the curve slope increases.

According to our model, γ counts only when the adversary launches the **MATCH** action. When $h_2 \geq 1$, the adversary can always launch the 51% attack, regardless of the reward. Therefore, the **MATCH** action is an infrequent choice compared to **OVERRIDE**, so the influence of γ is negligible in our case.

The slope change is suspected to be when $\beta H_a + \gamma H_{h,2} \geq (1 - \gamma) H_{h,2}$. At that point, the allocated mining power from the adversary plus his eclipsed honest mining power outperforms the un-eclipsed honest power. Consequently, the adversary is confident to override the small blockchain by **MATCH** action.

Vigilance of the merchant. Figure 2d shows the impact of N_c on the net revenue. We observe the net revenue decreases monotonically with N_c increasing, and finally reaches 0.

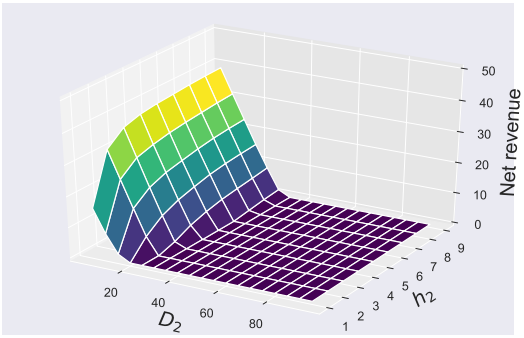
More block confirmations require the adversary to keep mining secretly for a longer time. This leads to a lower probability and greater cost of successful 51% attack through both types of attacks, and discourages 51% attacks on BC_2.

Mining power price. The impact of the mining power price pr is shown in Figure 2e. We observe that the net revenue decreases sharply with pr increasing, and finally reaches 0.

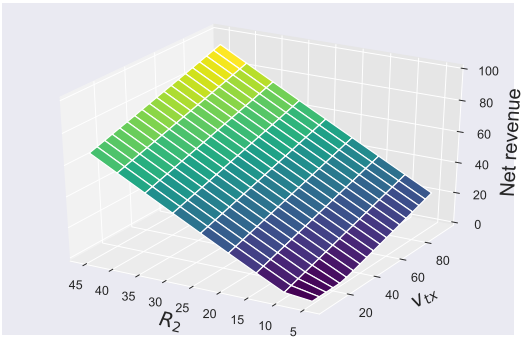
When the price of renting mining power is low, the related blockchains are vulnerable to the cloud mining attack as the attack cost is also low. Increasing pr leads to the greater cost of launching 51% attack through renting cloud mining power, which will discourage this kind of 51% attacks on BC_2.

4.3 Analysis

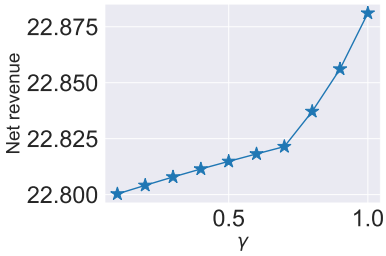
We observed several insights from our evaluation. First, an attacker's profit is mainly affected by the parameters that are out of the attacker's control. That is, to maximise attack revenue, an attacker should choose its target carefully. In particular, even though



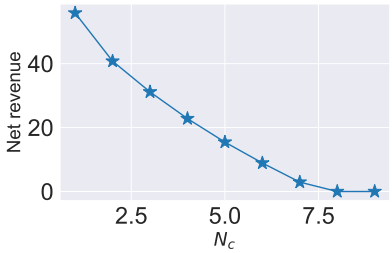
(a) Impact of h_2 and D_2.



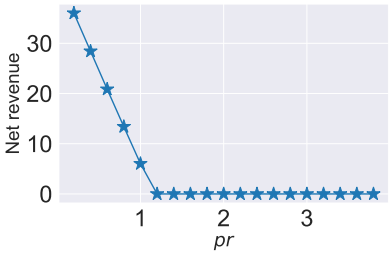
(b) Impact of v_{tx} and R_2.



(c) Impact of γ.



(d) Impact of N_c.



(e) Impact of pr.

Figure 2: Impacts of parameters on the net revenue of 51% attacks.

to some extent an attacker might have a chance to control the network propagation parameter γ, it has little impact on the revenue according to Figure 2c. All other parameters are out of the attacker's control. Thus, once choosing the targeted blockchains, the adversary has little control over the feasibility and profitability of the attack.

Second, the adversaries and the defenders should be aware of the public parameters D_2, h_2, v_{tx} and R_2. According to Figure 2a and Figure 2b, all of these parameters have a significant impact on the adversary's net revenue. Although either the adversaries or the defenders can hardly control these parameters, monitoring them in real time is possible. By monitoring these parameters, an adversary can identify a target with more expected profit, and an defender can be aware of potential attacks then react to this situation (e.g. by following the recommendations in Appendix 7.1).

Last, defenders (e.g. the cryptocurrency exchanges and the merchants) can reduce the feasibility and profitability of 51% attacks by increasing the number N_c of blocks for confirming transactions. Within these parameters, the defenders can only control N_c, and increasing N_c can greatly reduce the adversary's net revenue according to Figure 2d.

5 EVALUATION AGAINST SYSTEMS IN THE WILD

This section evaluates the security of mainstream PoW-based blockchains against 51% attacks. We evaluate the *mining power migration attack* on 3 pairs of top-ranked blockchains with the same mining algorithm, namely 1) Bitcoin (BTC) and BitcoinCash (BCH) with Sha256d, 2) Ethereum (ETH) and EthereumClassic (ETC) with Ethash, and 3) Monero(XMR) and ByteCoin (BCN) with CryptoNight. Our evaluation shows that, the *mining power migration attack* is quite easy and profitable on BTC/BCH and ETH/ETC, but it is not as effective on XMR/BCN. For the *cloud mining attack*, we evaluated the security of 15 leading PoW-based blockchains (chosen from the top 100 blockchains by their market caps [16]).

Our evaluation shows that the *cloud mining attacks* are feasible and profitable on all selected blockchains except Komodo (KMD), where the attack is feasible but not profitable.

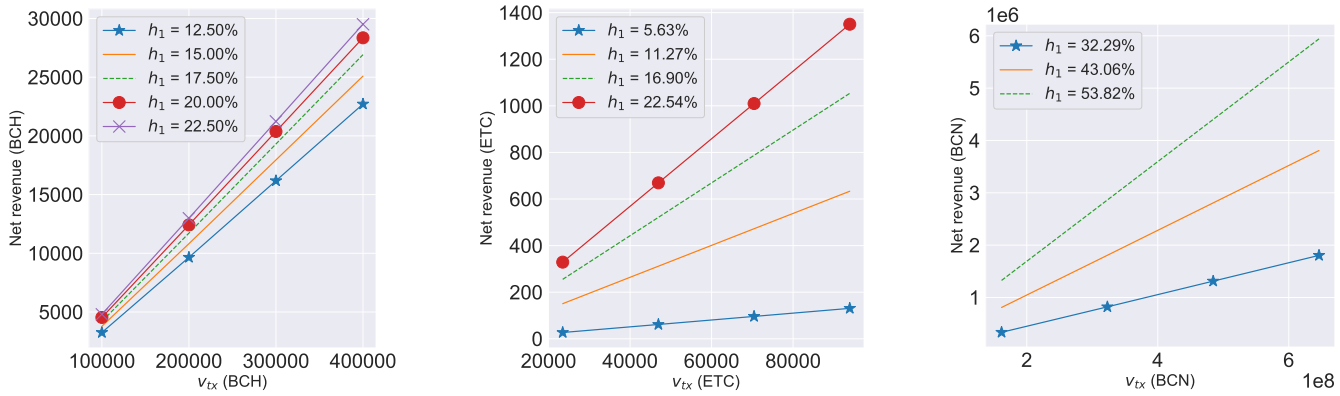
We use BTC and BCH as an example to demonstrate an optimal strategy derived from our model for a BTC miner to launch *mining power migration attacks* on BCH. The attack, together with explanations and observed insights, are presented in Appendix B.

5.1 Mining power migration attacks

We evaluate the profitability and feasibility of the mining power migration attack on 3 pairs of top-ranked cryptocurrencies with the same mining algorithm: BTC/BCH, ETH/ETC, and XMR/BCN. By permuting the adversary mining power H_a and the transaction value v_{tx}, our experiments reveal their relationship with the relative revenue.

As shown in Figure 3, it is surprisingly easy and profitable for a miner of BTC (or ETH) to launch a 51% attack on BCH (resp. ETC), while it is difficult and unprofitable for a XMR miner to attack BCN. For example, as shown in Figure 3:

- With approximately 12.5% mining power of BTC ($5000E + 15h/s$), an adversary can gain 6% (150 BCH, or $18,946.5)



(a) BTC and BCH **(b) ETH and ETC** **(c) XMR and BCN**

Figure 3: Mining power migration attacks on three different pairs of blockchains. We use $\gamma = 0.3$ for this group of experiments.

extra profit (than honest mining) by double-spending a transaction of 3000 BCH (equivalent to $378,930).

- With approximately 11.27% mining power of ETH ($16E + 12h/s$), the adversary can gain 1.33% (600 ETC, or $2,556) extra profit by double-spending a transaction of 90000 ETC (equivalent to $383,400).
- With approximately 43.05% mining power of XMR ($4E+8h/s$), the adversary can gain 0.67% (1,000,000 BCN or $619) extra profit by double-spending a transaction of 600,000,000 BCN (equivalent to $247,600).

The required mining power is not difficult to achieve. The top three mining pools in ETH are Sparkpool (30.9%), Ethermine (23.3%), f2pool2 (10.7%); and the top three mining pools in BTC are F2Pool (17.7%), Poolin (16.1%), BTC.com (11.9%) [3].

However, for XMR, a miner cannot profit much from the *mining power migration attack*. This is because the total available mining power in Monero is only about 2.8 times of the mining power in the BCN, although their market caps differ greatly. In comparison, the total available mining power in BTC is about 27.8 times of the total mining power in BCH; and the total available mining power in ETH is about 16.4 times of the total mining power in ETC.

Kwon et al. [17] also observed that BTC and BCH share the same mining algorithm, but their analysis and results differ from ours. First, our analysis is based on MDP, while their analysis is based on game theory. Second, their work still assumes the "honest majority", but we show that the adversary can launch 51% attacks on BCH and profit with less than 12.5% BTC mining power. Third, we consider the 51% attack - the Achilles' heel of PoW-based consensus, while their proposed *fickle mining* is only a mining strategy. Last, our main result is that PoW's current incentive mechanism incentivises miners to launch 51% attacks for more profit, while their main result is that PoW's current incentive mechanism weakens the security guarantee of PoW due to the *fickle mining*.

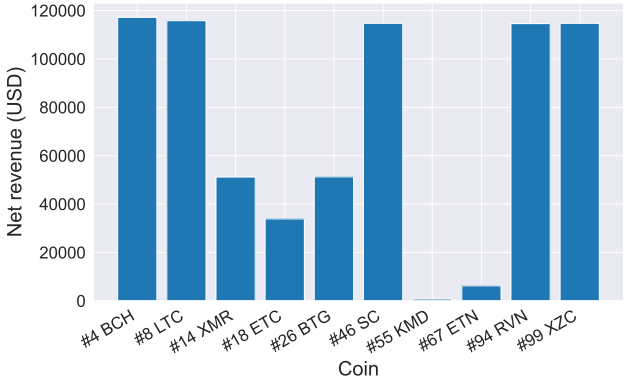


Figure 4: Cloud mining attacks on selected 10 PoW blockchains. We use v_{tx} = \$500,000 , $h_2 = 2$ and $\gamma = 0.3$. We use the value of N_c recommended by cryptocurrency communities.

5.2 Cloud mining attacks

We evaluate the security of 10 selected leading PoW-based blockchains against the *cloud mining attack*. In particular, there are 22 PoW-based blockchains in the Top 100 blockchains by their market cap[4]. Among them, DigiByte (DGB) and Verge (XVG) use multiple mining algorithms simultaneously and NiceHash does not support Bytom (BTM), ByteCoin (BCN), Electroneum(ETN), WaltonChain (WTC), and Aion (AION). In the rest 15 blockchains, NiceHash does not have enough mining power to attack five of them, including BTC with SHA256D, ETH with Ethash, ZEC with Equihash, DOGE with Scrypt and DASH with X11. Thus, we put the focus of our analysis on the rest 10 leading blockchains.

We set v_{tx} = \$500,000 (i.e. the double-spending transaction amount is \$500,000), and $h_2 = 2$ (i.e. the rentable mining power

[3]https://www.etherchain.org/charts/topMiners, and https://www.blockchain.com/pools. Data collected on 09/01/2020.
[4]Data fetched on 19 February 2019.

is twice of the honest mining power). We choose the value of N_c according to the recommended values from cryptocurrency community (listed in Figure 7).

Figure 4 summarises our evaluation results. It shows that, unfortunately, all selected blockchains are vulnerable towards the *cloud mining attack*. For example:

- the attacker needs approximately $2,000 to launch *cloud mining attack* on ETC for an hour, and the net revenue will be $33899 if successful;
- the attacker needs approximately $2,600 to launch the attack on BCH for an hour, and the net revenue will be $117198 if successful;
- the attack needs approximately $730 to launch the attack on ETN for one hour, and the net revenue will be $6222 if successful.

The only exception within these 10 blockchains is Komodo (KMD): the attacker cannot profit much by launching *cloud mining attacks* on KMD. To investigate the reason of this, we additionally evaluate the security of KMD against both *mining power migration attacks* and *cloud mining attacks*. More specifically, we evaluate the impact of the adversary's mining power (h_1 and h_2) and the transaction value (v_{tx}) on the attacker's profit (in Figure 5). The evaluation result shows that, although feasible, both attacks on KMD will not give much extra profit - the attacker can only gain 1% $\sim$ 2% more revenue compared to honest mining. In addition, while both attacks are not very profitable, the revenue of launching *cloud mining attack* is still better than the revenue of launching *mining power migration attack*. For the profitability, the reason is that KMD requires 30 blocks to confirm a transaction (i.e., $N_c = 30$) [18], which is a much higher requirement than other blockchains. As shown in §4, increasing N_c can reduce the profit of 51% attacks. Later in Appendix 7, we will also demonstrate in detail how effective it is to prevent 51% attacks by adjusting different parameters, including N_c.

6 CASE STUDY: THE 51% ATTACK ON ETHEREUM CLASSIC AT 07/01/2019

Ethereum Classic (ETC) is a PoW-based blockchain using the same mining algorithm as Ethereum (ETH). As shown in §5, both ETH and ETC are vulnerable to the cloud mining attack, and ETC is also vulnerable to the *mining power migration attack* (from ETH or other GPU-friendly PoW-based blockchains). On 07/01/2019, a 51% attack happened to ETC, where the attacker double-spent transactions of more than $1.1 million on a cryptocurrency exchange Gate.io [7]. Though the actual source is still a mystery, NiceHash, a cloud mining service, is highly suspected as the mining power source [19, 20]. Interestingly, the attacker returned ETC coins equivalent to $100,000 to Gate.io later [8].

In this section, we investigate this cloud mining attack on ETC by using 51-MDP. First, we reveal the attacker's strategy from his behaviours during the attack, and conclude that the attacker's strategy in this incident is the best practice of launching *cloud mining attacks*. Second, we evaluate the attack using 51-MDP, and compare the attacker's revenue between the two attacks. Our comparison shows that, when the rentable mining power is sufficient, *cloud mining attack* is much more profitable than *mining power migration attack*. Last, we estimate the attacker's net revenue using 51-MDP (assuming he was using NiceHash to launch *cloud mining attacks*). Our estimated net revenue is approximate to $100,000, the amount that the attacker returned to Gate.io [8]. This indicates that the attacker's revenue is near optimal, and further implies that the attacker may launch the *cloud mining attacks* in a fine-grained way.

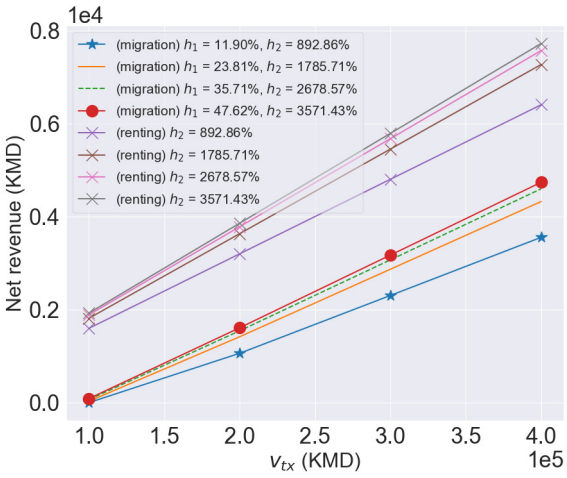


Figure 5: Profitability of mining power migration attacks and cloud mining attacks on Komodo (KMD). We choose $\gamma = 0.3$, and $N_c = 30$ - the values recommended by KMD community.

6.1 The attack details

Table 3: All double-spent transactions during the 51% attack on ETC [21]. In this attack, 12 transactions were double-spent from two accounts.

Trans. ID (in short)	From	To	Amount (ETC)	Height	Waiting time (#block)
0x1b47a700c0	0x3ccc8f7415	0xbbe1685921	600	7249357	-
0xbba16320ec	0x3ccc8f7415	0x2c9a81a120	4000	7254430	**5073**
0xb5e0748666	0x3ccc8f7415	0x882f944ece	5000	7254646	216
0xee31dffb66	0x3ccc8f7415	0x882f944ece	9000	7255055	409
0xfe2da37fd9	0x3ccc8f7415	0x2c9a81a120	9000	7255212	157
0xa901fcf953	0x3ccc8f7415	0x2c9a81a120	15700	7255487	275
0xb9a30cee4f	0x3ccc8f7415	0x882f944ece	15700	7255554	67
0x9ae83e6fc4	0x3ccc8f7415	0x882f944ece	24500	7255669	115
0xaab50615e3	0x3ccc8f7415	0x53dffbb307	5000	7256012	343
0xd592258715	0x07ebd5b216	0xc4bcfee708	26000	7261492	**5480**
0x9a0e8275fc	0x07ebd5b216	0xc4bcfee708	52800	7261610	118
0x4db8884278	0x07ebd5b216	0xc4bcfee708	52200	7261684	74
					Total: 219500 ETC

At the beginning of 2019, a 51% attack on ETC resulted in the loss of more than 1.1 million dollars. The attack lasted for 4 hours, from 0:40 am UTC, 07/01/2019 to 4:20 am UTC, 07/01/2019, approximately. During the attack, the attacker repetitively launched a coin withdrawal transaction on the Gate.io cryptocurrency exchange [22], then launched double-spending attacks [7]. Among these attempts, 12 transactions were successfully double-spent (listed in Table 3). Interestingly, several days after the attack, the attacker returned ETC equivalent to $100,000 to Gate.io [8].

The source of the mining power for this attack remains uncertain due to the anonymity of miners. However, the NiceHash cloud mining platform [5] is highly suspected: One day before the attack, an anonymous person rented all available Ethash (the mining algorithm used by ETH and ETC) mining power from NiceHash [19, 20].

6.2 Analysing the attacking strategy

According to Table 3, the attacker continuously increased the value of new transactions throughout the attack (except the last double spending of the first account). It is suspected that this behavior belongs to the strategies used by the attacker to maximise and stabilise his revenue, for the following reasons.

Stabilising the revenue. First, launching multiple small double-spending attempts can stabilise the expected revenue. The double-spending attack may fail in a limited time period, even if the adversary controls more than 50% of the computing power. Compared to a one-off attempt, the revenue will be more stable if dividing a transaction into multiple smaller transactions.

Avoiding risk management systems. Second, this strategy may be used for avoiding risk management systems of cryptocurrency exchanges. Most cryptocurrency exchanges run their own risk management system to combat the misbehaviors, like the fraudulent payments and the abnormal login attempts. A huge coin withdrawal transaction is highly possible to trigger the risk management system, while multiple small transactions would be overlooked. Meanwhile, a big transaction may lead to longer confirmation time, and a longer attack period is easier to be detected. Therefore, defeating the risk control system is naturally a part of the attacker's strategy. According to the Gate.io report [7], the risk management system ignored transactions from the attacker, as the attack was decently prepared - they registered and real-name authenticated the account on Gate.io more than 3 months before the attack. Slowly increasing the transaction value is also highly suspected as an approach for reverse-engineering the threshold of invoking the risk management system.

Using multiple wallets. In addition, we investigate the waiting time between each two attacks (quantified by using the number of blocks). The waiting time varies mostly from 67 blocks to 409 blocks. Interestingly, there are two much bigger gaps of more than 5000 blocks before the transactions 0xbba16320ec and 0xd592258715. The first gap is after the first attack, and the second gap is before the attacker changed to another account to send double-spending transactions. The first gap may be because the attacker was cautious when first launching the double-spending attack. The attacker launched a double-spending transaction of only 600 ETC coins, which is much smaller than his following transactions. After the first attack, the attacker waited for a long time to confirm the success of the first attack, then started to increase the transaction value. The second gap may be because the attacker ran out of money in his first account 0x3ccc8f7415, so changed to another account 0x07ebd5b216. The last transaction 0xd592258715 sent by 0x3ccc8f7415is is right before the second gap. It's value is 5000 ETC coins, which is much smaller than its previous transaction of 24500 ETC coins. After the transaction 0xd592258715, the attacker changed to his another account 0x07ebd5b216, which caused the time gap of 5480 blocks.

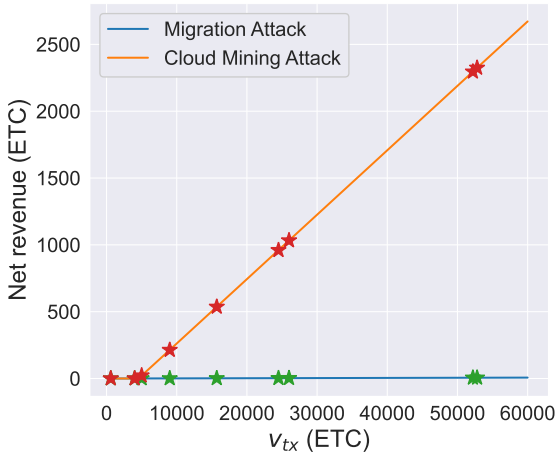


Figure 6: Simulated 51% attack on ETC. The blue line denotes the relative reward of *cloud mining attacks*. The orange line denotes the relative reward of *mining power migration attacks* for making comparisons. We also marked different transaction amounts in the attack using dots.

6.3 Evaluation using 51-MDP

To analyse this attack, we collect attack-related data during the time period of the attack (on 07/01/2019). Table 8 in Appendix C summarises the experimental data. According to Gate.io [22], during the attack's time period, the required number of blocks to confirm transactions was 12 (i.e., $N_c = 12$), which is also recommended by of ETH community and ETC community [23]. The price of ETC on that day was $5.32, and the price of BTC was $4061.47. The mining difficulty of ETC was 131.80E+12, and the ratio h_2 of attacker's mining power over the honest mining network was about 1.16, i.e., the attacker approximately controls 53.7% mining power during the attack. The reward of successfully mining a block is 4 ETC coins, and the price of renting Nicehash mining power on that day is 3.8290 BTC/TH/day. As there is no data on the attacker's connectivity w.r.t. propagating his blocks, and the impact of γ is relatively small (as previously discussed), we assume that $\gamma = 0.3$.

We permute and mark the transaction values used by the attacker. We also plot the same curve in the *mining power migration attack* to compare the profitability of two mining power sources. The result is shown in Figure 6.

The result shows that when the transaction value is over 5000 ETC, double-spending is more profitable than by honest mining. Having a transaction (or a set of transactions) of value over 5000 ETC (approximately $26,000 at the time of attack) should not be difficult for an attacker, so the incentive of launching double-spending attacks is very strong.

Compare to the *mining power migration attack*, the *cloud mining attack* is more profitable. This means that for the ETH/ETC pair, renting mining power to attack ETC is much cheaper than migrating mining power from ETH. The reason may be the GPU friendliness of the ETH/ETC mining algorithm. Both ETH and ETC use Ethash [24] as the mining algorithm. Ethash is a memory-hard function, making

it GPU-friendly while ASIC-resistant [25]. As GPU is not dedicated hardware, it can be adopted for different applications. Therefore, renting mining power for ETH/ETH is much cheaper compared to renting mining power with dedicated hardware such as ASICs.

6.4 Estimating the attacker's net revenue

According to Table 3, the attacker has stolen 219500 ETC. This is his gross revenue rather than his net revenue, and the cost of attacking is uncertain. However, as we don't know transactions of unsuccessful 51% attack attempts, the cost of attacks is hard to determine from the blockchain data.

Nevertheless, we can utilise 51-MDP to estimate the attacker's net revenue. From transactions of attackers that exist on the blockchain, we can estimate their net revenues. If we know how much mining power the attacker used, we can estimate the success rate of his attacks. With this success rate, we can reverse-engineer the net revenue of unsuccessful attempts. Our model produces the expected net revenue of a single attack, regardless whether it is successful or not. While only successful attacks were observed, failed attacks also contribute to the theoretical expected net revenue. Thus, by summing them together, we can reverse-engineer the attacker's estimated net revenue.

We find that our estimated net revenue is approximately $84773.40. This is close to $100,000 - the value that the attacker returned to Gate.io after the attack [8].

Modelling. We first derive the success rate of attacking from mining power of the attacker. Let N_c be the required number of blocks to confirm transactions, and h_2 be the ratio of attacker's mining power over the honest network. Then, the attacker controls $p = \frac{h_2}{h_2+1}$ of the total mining power. Mining can be modelled as a binomial distribution

$$B(n_a + n_h, p) \tag{7}$$

where n_a and n_h are the numbers of blocks the adversary and the honest network mined, respectively. Let $Pr(X = n_a)$ be the probability of the attacker to mine n_a blocks while honest miners mine h_h blocks, and we have

$$Pr(X = n_a) = Pr(n_a; n_a + N_c, p) \tag{8}$$

When $n_h = N_c \wedge n_a < N_c$, the attack fails. Thus, the probability of a successful 51% attack P is calculated as

$$P = 1 - \sum_{n_a=0}^{N_c-1} Pr(n_a; n_a + N_c, p) \tag{9}$$

Then we derive the estimated net revenue from observed successful attacks. Let R_s and R_f be the estimated revenue of successful and failed attack attempts, respectively. We have

$$\frac{R_s}{P} = \frac{R_s}{1-P} \Longrightarrow R_f = \frac{(1-P)R_s}{P} \tag{10}$$

and the total estimated net revenue R is

$$R = R_s + R_f = R_s + \frac{(1-P)R_s}{P} \tag{11}$$

Estimation. According to Figure 6, summing all net revenues of successful transactions, the total net revenue of the attacker derived from our model is approximately 9000 ETC coins ($R_S = 9000$). Recall that $h_2 = 1.16$, and the attacker controls $p = \frac{h_2}{h_2+1} = 53.7\%$ of ETC mining power. From Equation 9, we calculate the success rate P of attacks as

$$\begin{aligned} P &= 1 - \sum_{n_a=0}^{N_c-1} Pr(n_a; n_a + N_c, p) \\ &= 1 - \sum_{n_a=0}^{N_c-1} C^{n_a}_{n_a+N_c} p^{n_a} (1-p)^{N_c} \\ &= 1 - \sum_{n_a=0}^{11} C^{n_a}_{n_a+12} p^{n_a} (1-p)^{12} \\ &= 56.48\% \end{aligned} \tag{12}$$

From Equation 11, we calculate the estimated net revenue R as

$$\begin{aligned} R = R_s + R_f &= R_s + \frac{(1-P)R_s}{P} \\ &= 9000 + \frac{(1-0.5648)\cdot 9000}{0.5648} \\ &= 9000 + 6934.85 = 15934.85(ETCcoins) \end{aligned} \tag{13}$$

Therefore, the expected total net revenue is 9000 + 6934.85 = 15934.85 ETC coins, which is equivalent to $84773.40 at the time of attack.

This results implies that, the attacker adopted a near optimal strategy for launching *cloud mining attacks*. $84773.40 - our estimated revenue - is slightly less than $100,000 - the amount of money the attacker returned to Gate.io. The attacker is impossible to return money more than he earned from the attack, so his revenue should be more than $100,000. If so, the attacker's revenue should be optimal. To achieve the optimal revenue, the attacker should launch *cloud mining attacks* using the optimal strategy, which is fine-grained, as shown in Table 4 in Appendix B.

7 DISCUSSIONS ON ATTACK PREVENTION

This section discusses short term and long term solutions to detect and prevent both the *mining power migration attack* and *cloud mining attack*. We make use of the 51% attack incident on ETC (see §6) as an example, and demonstrate how to make use of 51-MDP to gain insights that helps to defend against such cloud mining attacks in Section 7.1.

7.1 Quick remedies

We first discuss several quick remedies for cryptocurrency exchanges to reduce the damage of 51% attacks. It consists of detecting potential attack attempts, and reacting upon detection through conventional risk management techniques.

Detecting 51% attacks. For the two 51% attacks, the attacker needs to move a considerable amount of mining power from somewhere, such as the other blockchain or a cloud mining service.

This gives us an opportunity to detect the anomaly state where a "large" portion of mining power suddenly disappears from a source. For example, a potential victim can monitor the available compatible mining power of other blockchains or cloud mining services. If there is a sudden change on the amount of total available

mining power, then this might indicate a potential 51% attack. The threshold of "large" is blockchain specific according to the risk management rules. For example, a blockchain which cares less on such attacks can set the threshold to 100% of its current total mining power. That is, once the disappearance of this amount of mining power in other sources is detected, then an alarm of a potential attack is raised. However, this will not detect an attacker who gains 90% mining power from one source, and 10% from another sources. A more cautious blockchain may set a tighter threshold, e.g. 5%, however, this may cause false positive alarms.

There are two limitations of this method. First, it may introduce false positive detections, and it is hard to identify which blockchain will be the victim upon detection. Second, it is expensive to monitor all the possible mining compatible blockchains and cloud mining services in real-time. Even though, the monitoring result may be inaccurate.

Reactions upon 51% attacks. Upon detection, a potential victim can react to manage potential risks. Several reactions can be taken to reduce the potential damage from both the two attacks studied in our paper. The first reaction is to increase the number N_c of block confirmations. As shown in Figure 2d, in our experiment setting (Table 5), with the increase of required number of confirmations, the related revenue decreases. In addition, decreasing the maximum amount of cash out in a single transaction also helps. As shown in Figure 2b, the higher the transaction value is, the more net revenue an attacker can gain. Thus, decreasing the maximum value of a transaction for cash out would discourage a rational miner to launch such 51% attacks.

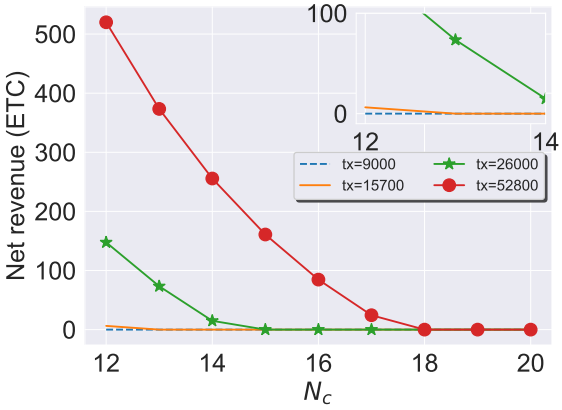


Figure 7: Impacts of v_{tx} and N_c on the ETC attack.

Our model can be used to provide recommendations on the above mentioned parameters. For example, Figure 7 shows the impact of the value v_{tx} of transactions and the number N_c of confirmations on the 51% on ETC. This analysis is produced by using our 51-MDP model, and all other parameters are set up according to the attack happened. This shows that if the value of transactions was limited to 9,000 ETC (approximately $38340.0) per transaction, then the attacker will not get any net revenue. On the other side, if the exchange wants to allow a maximum of 52,800 ETC (approximately $224928.0), then our model recommends that the N_c should be increased to 18 to eliminate the incentive of a rational miner to launch such attacks.

In addition, decreasing the maximum frequency of cash out would also limit the potential damage from an attacker, as it reduces the daily withdraw limit.

Last, if a potential attack is considered very likely, then the potential victim can halt the cash out temporarily, to increase the cost of the attack.

7.2 Long term solutions

Though easy to deploy, aforementioned quick remedies are not sufficient. First, they sacrifice the usability of blockchains. Second, all of them only minimize the effect of the potential attacks, rather than eliminating them.

Improving the PoW protocol from the protocol-level is also a promising approach to defend against our attacks. There are limited works aiming at minimizing the effects of powerful miners being malicious. For example, RepuCoin [26] aims at mitigating the 51% attacks in PoW protocols by introducing the "physics-based reputation". In RepuCoin, the weight of each miner is decided by the reputation rather than the mining power. The reputation of a miner depends on the mining power, but also takes the past contribution of miners into consideration. In this way, a 51% attacker cannot gain a high-enough reputation within a short time period, and the 51% attacks we studied become much harder to launch.

8 CONCLUSION

Honest majority is the most important assumption of PoW-based blockchains. In this paper, we challenge this core security assumption of PoW-based consensus. In particular, we show that the incentive mechanism employed by the current PoW-based consensus may encourage malicious behaviours to launch 51% attacks. We consider two scenarios leading to such 51% attacks, namely the *mining power migration attack* and the *cloud mining attack*. To investigate such 51% attacks, we develop the 51-MDP model, which can estimate the cost and the revenue of such 51% attacks. Our evaluation shows that, for most mainstream PoW-based blockchains, profit-driven miners have an incentive to launch 51% attacks to gain extra profit. We provide an analysis on the recent 51% attacks on ETC on 07/01/2019 using our model, which successfully estimates the attacker's revenue and describes the attacker's attacking strategy.

REFERENCES

[1] Satoshi Nakamoto. 2008. Bitcoin: a peer-to-peer electronic cash system.

[2] komodoplatform.com. 2019. The Anatomy Of A 51% Attack And How You Can Prevent One. (2019). https://komodoplatform.com/51-attack-how-komodo-can-help-prevent-one.

[3] Amitanand S. Aiyer, Lorenzo Alvisi, Allen Clement, Michael Dahlin, Jean-Philippe Martin, and Carl Porth. 2005. BAR fault tolerance for cooperative services. In *SOSP*.

[4] Joseph Bonneau. 2016. Why Buy When You Can Rent? - Bribery Attacks on Bitcoin-Style Consensus. In *Financial*

Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, 19–26.

[5] nicehash. 2019. Nicehash - Largest Crypto-Mining Marketplace. (2019). https://www.nicehash.com.

[6] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. 2016. On the Security and Performance of Proof of Work Blockchains. In *CCS*.

[7] gate.io. 2019. Gate.io research: confirmed the etc 51% attack and attacker's accounts - gate.io news. (2019). https://www.gate.io/article/16735.

[8] . 2019. Gate.io Got Back 100k USD Value Of ETC From The ETC 51% Attacker. (2019). https://www.gate.io/article/16740.

[9] Yujin Kwon, Hyoungshick Kim, Jinwoo Shin, and Yongdae Kim. 2019. Bitcoin vs. Bitcoin Cash: Coexistence or Downfall of Bitcoin Cash? *CoRR*, abs/1902.11064. arXiv: 1902.11064.

[10] Alexander Spiegelman, Idit Keidar, and Moshe Tennenholtz. 2018. Game of coins. *arXiv preprint arXiv:1805.08979*.

[11] Kevin Liao and Jonathan Katz. 2017. Incentivizing blockchain forks via whale transactions. In *International Conference on Financial Cryptography and Data Security*. Springer, 264–279.

[12] Fredrik Winzer, Benjamin Herd, and Sebastian Faust. 2019. Temporary censorship attacks in the presence of rational miners. Cryptology ePrint Archive, Report 2019/748. (2019).

[13] Aljosha Judmayer, Nicholas Stifter, Alexei Zamyatin, Itay Tsabary, Ittay Eyal, Peter Gazi, Sarah Meiklejohn, and Edgar Weippl. 2019. Pay-to-win: incentive attacks on proof-of-work cryptocurrencies. Cryptology ePrint Archive, Report 2019/775. (2019).

[14] Iadine Chadès, Guillaume Chapron, Marie-Josée Cros, Frédérick Garcia, and Régis Sabbadin. 2014. MDPtoolbox: a multi-platform toolbox to solve stochastic dynamic programming problems. *Ecography*, 37, 9, 916–920.

[15] Sheldon M Ross. 2014. *Introduction to stochastic dynamic programming*.

[16] coinmarketcap.com. 2019. Top 100 Cryptocurrencies by Market Capitalization. (2019). https://coinmarketcap.com.

[17] Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Y. Vasserman, and Yongdae Kim. 2017. Be selfish and avoid dilemmas: fork after withholding (FAW) attacks on bitcoin. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, 195–209.

[18] . 2018. Security: delayed proof of work (dpow). (2018). https://komodoplatform.com/security-delayed-proof-of-work-dpow/.

[19] David Z. Morris. 2019. The Ethereum Classic 51% attack is the height of crypto-irony. (2019). https://breakermag.com/the-ethereum-classic-51-attack-is-the-height-of-crypto-irony/.

[20] Wes Messamore. 2019. Nicehash to smaller cryptocurrency miners : if you can't beat 51% attackers who lease our hash power, join them. (2019). https://www.ccn.com/nicehash-to-smaller-cryptocurrency-miners-if-you-cant-beat-51-attackers-who-lease-our-hash-power-join-them.

[21] Mark Nesbitt. 2019. Deep Chain Reorganization Detected on Ethereum Classic (ETC). (2019). https://blog.coinbase.com/ethereum-classic-etc-is-currently-being-51-attacked-33be13ce32de.

[22] gate.io. 2019. Gate.io - the gate of blockchain assets exchange. (2019). https://www.gate.io.

[23] reddit. 2019. How many confirms is considered 'safe' in Ethereum? (2019). https://www.reddit.com/r/ethereum/comments/4eplsv/how_many_confirms_is_considered_safe_in_ethereum.

[24] James Ray. 2019. Dagger Hashimoto. (2019). https://github.com/ethereum/wiki/wiki/Dagger-Hashimoto.

[25] James Ray. 2018. Ethash Design Rationale. (2018). https://github.com/ethereum/wiki/wiki/Ethash-Design-Rationale.

[26] Jiangshan Yu, David Kozhaya, Jeremie Decouchant, and Paulo Verissimo. 2019. Repucoin: your reputation is your power. *IEEE Transactions on Computers*.

[27] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. 2015. The Bitcoin Backbone Protocol: Analysis and Applications. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, 281–310.

[28] Rafael Pass, Lior Seeman, and Abhi Shelat. 2017. Analysis of the Blockchain Protocol in Asynchronous Networks. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, 643–673.

[29] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. 2017. The Bitcoin Backbone Protocol with Chains of Variable Difficulty. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, 291–323.

[30] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. Ouroboros: a provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*. Springer, 357–388.

[31] Ren Zhang and Bart Preneel. 2019. Lay Down the Common Metrics: Evaluating Proof-of-Work Consensus Protocols' Security. In *Proceedings of the 40th IEEE Symposium on Security and Privacy* (S&P). IEEE.

[32] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. 2016. Optimal Selfish Mining Strategies in Bitcoin. In *Financial Cryptography and Data Security - 20th International Conference, FC 2016, Christ Church, Barbados, February 22-26, 2016, Revised Selected Papers*, 515–532.

[33] Lucianna Kiffer, Rajmohan Rajaraman, and Abhi Shelat. 2018. A Better Method to Analyze Blockchain Consistency. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, 729–744.

[34] Ittay Eyal and Emin Gün Sirer. 2014. Majority Is Not Enough: Bitcoin Mining Is Vulnerable. In *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, 436–454.

[35] Ittay Eyal. 2015. The Miner's Dilemma. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, 89–103.

[36] Miles Carlsten, Harry A. Kalodner, S. Matthew Weinberg, and Arvind Narayanan. 2016. On the Instability of Bitcoin Without the Block Reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, 154–167.

[37] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. 2016. Stubborn mining: generalizing selfish mining and combining with an eclipse attack. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, 305–320.

[38] Nick Arnosti and S Matthew Weinberg. 2018. Bitcoin: a natural oligopoly. *arXiv preprint arXiv:1811.08572*.

A OTHER RELATED WORK

A.1 Formalisations of PoW-based consensus

There are several attempts on formalising security properties of PoW-based consensus. Garay et al. [27] formalised Bitcoin's consensus under the synchronous network setting, and extracted two refined properties, namely *common prefix* and *chain quality*. Pass et al. [28] extended this formalisation to a bounded asynchronous network setting, and Garay et al. [29] further considered the dynamic difficulty adjustment of PoW. Kiayias et al. [30] further defined another property on the liveness called the *chain growth*.

A.2 Evaluation frameworks on PoW-based consensus security

Gervais et al. [6] first proposed an evaluation framework for quantitatively analyse PoW-based consensus security using MDP, with a focus on the resistance of selfish mining and double-spending. Zhang et al. [31] generalised this framework and proposed a cross-protocol evaluation framework with three new properties measuring the resistance of several attacks on PoW-based consensus, namely the *incentive compatibility* (i.e. the net revenue lower bound of honest miners under selfish mining attacks), the *subversion gain* (i.e. the profit upper bound of an adversary performing double spending), and the *censorship susceptibility* (i.e. the profit loss of honest miners under censorship retaliation attacks).

A.3 Evaluation of attacks on PoW-based consensus

Most research evaluating attacks on PoW-based consensus use the MDP-based models [32, 6, 33, 31] or the game-theoretic models [34, 35, 36, 37, 17, 9, 13, 38]. Our research adopts the MDP-based model to evaluate the two 51% attacks, which are mining power migration and cloud mining attacks, which we call 51-MDP. 51-MDP is similar with models in [32, 6] in the notations and the processes, but in addition supports the presence of external mining power. Model the external mining power is complex, because the number of parameters will be doubled. We reduce the excessive parameters in a way that simplifies the implementation and the simulation of 51-MDP without losing any correctness of the model.

B OPTIMAL STRATEGY FOR BTC/BCH

In this section, we show the optimal strategy of launching mining power migration attacks from BTC to BCH. We use the same experimental setting (with one pair of hashrate and transaction amount), as in §5. More specifically, we assume the adversary with $\alpha = 0.3$ uses the Sha256d mining power of hashrate $5000E + 15$ (i.e. $h_1 = 0.125$ and $h_2 = 3.462$), and a transaction of \$300,000 to launch *mining power migration attacks*. We use the default value 6 for N_c. We apply the *ValueIteration* algorithm [15] with a discount value of 0.9 and an epsilon value of 0.1 for 51-MDP.

Table 4 outlines the optimal strategy with notations. Each subtable describes the optimal strategy with a fixed β. For each table, the x-axis is l_h, while the y-axis is l_a. For each cell in a table, the three letters denote the optimal actions when $fork = r, ir, p$, respectively. More specifically, A, O, W, M denote **ADOPT**, **OVERRIDE**, **WAIT** and **MATCH**, respectively; **W** and **M** denote **WAIT_INC** and **MATCH_INC**, respectively; and w and m denote **WAIT_DEC** and **MATCH_DEC**, respectively. Among all cells, there are 7 different values labelled by different colours, namely: AAA in light blue; **W**AA in deep blue; **MMW** in green; OAO in yellow; **W**A**W** in red; WAW in brown; and wAw in purple (which only appears when $\beta \geq 0.2$).

We divide each matrix into four parts according to N_c (here $N_c = 6$), namely the upper left, upper right, lower left, and lower right. The upper left part represents the situation such that after the last common block in the blockchain, both honest branch and the attacker's branch do not contain enough number (N_c) of blocks as required for confirmation; whereas in the lower right part, both branches contain enough number (N_c) of blocks. In the upper right part, the honest branch contains enough number (N_c) of blocks as required for confirmation, but not the attacker's branch, whereas the lower left represents the opposite scenario.

The upper left part (when $l_a < N_c \wedge l_h < N_c$). In this scenario, the adversary's optimal action is mostly **WAIT_INC** i.e. increasing his mining power on BCH for mining more blocks. Note that in this scenario, $fork$ can only be p (i.e. the adversary's branch is still private and unpublished), and the first two letters for each cell (represent the action when $fork = r \vee fork = ir$, respectively) are unreachable states. When $l_a < N_c \wedge l_h < N_c$, the merchant does not confirm the transaction, so the adversary cannot publish his branch to double-spend. The adversary needs at least N_c blocks to revert the honest blockchain, as the merchant will accept the transaction only when $l_h \geq N_c$. Therefore, at this stage, the adversary should make $l_a \geq N_c$ as fast as possible, which can be achieved by allocating more mining power on BCH. When $l_a = 5 \wedge l_h = 0 \wedge \beta = 0.8$, the optimal action is **WAIT**. The reason is that the adversary has already gained significant advantage (5 blocks longer than the honest blockchain), and he has already secured the attack with his existing mining power with a high probability. When $\beta = 1.0$, the optimal action is **WAIT** except for $l_a = 5 \wedge l_h = 0$, where the optimal strategy is **WAIT_DEC**. In this scenario, the adversary has no more mining power for BCH, so cannot do **WAIT_INC**. When $l_a = 5 \wedge l_h = 0$, the adversary can even move some mining power on BCH back to BTC, so that he gains more reward from honestly mining BTC while securing the attack on BCH.

Table 4: Optimal strategy for a BTC miner to launch *mining power migration attacks* on BCH, where w denotes WAIT_DEC, W denotes WAIT, W denotes WAIT_INC, m denotes MATCH_DEC, M denotes MATCH, M denotes MATCH_INC, O denotes OVERRIDE, and A denotes ABORT.

(a) $\beta = 0$

	0	1	2	3	4	5	6	7	8	9
0	WAW	WAW	WAW	WAW	WAW	WAW	WAA	AAA	AAA	AAA
1	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAA	AAA	AAA
2	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAA	AAA
3	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA
4	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
5	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
6	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
7	WAW	WAW	WAW	WAW	WAW	WAW	OAO	MMW	WAW	WAW
8	WAW	WAW	WAW	WAW	WAW	WAW	OAO	OAO	MMW	WAW
9	WAW	WAW	WAW	WAW	WAW	WAW	OAO	OAO	OAO	MMW

(b) $\beta = 0.2$

	0	1	2	3	4	5	6	7	8	9
0	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA	AAA	AAA
1	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA	AAA
2	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA
3	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
4	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
5	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
6	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
7	wAw	wAw	wAw	wAw	wAw	wAw	OAO	MMW	WAW	WAW
8	wAw	wAw	wAw	wAw	wAw	wAw	OAO	OAO	MMW	WAW
9	wAw	wAw	wAw	wAw	wAw	wAw	OAO	OAO	OAO	MMW

(c) $\beta = 0.4$

	0	1	2	3	4	5	6	7	8	9
0	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA	AAA	AAA
1	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA	AAA
2	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA
3	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
4	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
5	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
6	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
7	wAw	wAw	wAw	wAw	wAw	wAw	OAO	MMW	WAW	WAW
8	wAw	wAw	wAw	wAw	wAw	wAw	OAO	OAO	MMW	WAW
9	wAw	wAw	wAw	wAw	wAw	wAw	OAO	OAO	OAO	MMW

(d) $\beta = 0.6$

	0	1	2	3	4	5	6	7	8	9
0	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA	AAA	AAA
1	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA	AAA
2	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA
3	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
4	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
5	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
6	wAw	wAw	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
7	wAw	wAw	wAw	wAw	wAw	wAw	OAO	MMW	WAW	WAW
8	wAw	wAw	wAw	wAw	wAw	wAw	OAO	OAO	MMW	WAW
9	wAw	wAw	wAw	wAw	wAw	wAw	OAO	OAO	OAO	MMW

(e) $\beta = 0.8$

	0	1	2	3	4	5	6	7	8	9
0	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA	AAA	AAA
1	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA	AAA
2	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA
3	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
4	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
5	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
6	wAw	wAw	wAw	wAw	WAW	WAW	WAW	WAW	WAW	WAW
7	wAw	wAw	wAw	wAw	wAw	wAw	OAO	MMW	WAW	WAW
8	wAw	wAw	wAw	wAw	wAw	wAw	OAO	OAO	MMW	WAW
9	wAw	wAw	wAw	wAw	wAw	wAw	OAO	OAO	OAO	MMW

(f) $\beta = 1.0$

	0	1	2	3	4	5	6	7	8	9
0	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA	AAA	AAA
1	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA	AAA
2	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	AAA
3	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
4	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
5	wAw	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW	WAW
6	wAw	wAw	wAw	wAw	wAw	WAW	WAW	WAW	WAW	WAW
7	wAw	wAw	wAw	wAw	wAw	wAw	OAO	MMW	WAW	WAW
8	wAw	wAw	wAw	wAw	wAw	wAw	OAO	OAO	MMW	WAW
9	wAw	wAw	wAw	wAw	wAw	wAw	OAO	OAO	OAO	MMW

The upper right part (when $l_a < N_c \wedge l_h \geq N_c$). In this scenario, the merchant has confirmed the transaction (as $l_h \geq N_c$), but the adversary's branch falls behind the honest blockchain. The adversary's optimal action is **Abort** with $l_h - l_a \geq 7$ (the light blue upper right corner), and mostly **WAIT_INC** (**WAIT** when $\beta = 1.0$) with $l_h - l_a < 7$. When $l_h - l_a \geq 7$, the adversary's branch significantly falls behind the honest blockchain, so he should give up to reduce the damage. When $l_h - l_a \leq 5$, the adversary's branch does not fall behind too much, so he still has a chance to catch up by increasing its minging power (i.e., **WAIT_INC**). When $\beta = 0.0 \wedge l_a - l_h = 6 \wedge l_a \neq 9$ (the dark blue area), the adversary's optimal action is **WAIT_INC** with $fork = r$ (i.e. the adversary's branch is published but the honest blockchain is confirmed), but is **ABORT** with $fork = p$ (i.e. the adversary's branch is unpublished). When the adversary publishes his branch, some miners with γ honest mining power choose to mine on this branch. In this way, the adversary obtains extra mining power from other miners, so becomes more confident on the attack.

The lower left part (when $l_a \geq N_c \wedge l_h < N_c$). In this scenario, the merchant has not confirmed the transaction (as $l_h \leq N_c$). If $l_a > N_c$, the adversary has secured the attack: he can just wait for the merchant to confirm the transaction (when the honest blockchain reaches N_c), then publish his branch to revert the blockchain. If $l_a = N_c$, the adversary only needs to mine one more block to secure the attack. When β becomes bigger, the adversary is more intended to do **WAIT_DEC** (the purple area) compared to **WAIT** (the brown area) and **WAIT_INC** (the red area). Similar with the upper right part, with bigger β, the adversary has a good chance to make the attack successful, so he can use less mining power to attack BCH while using more mining power to honestly mine BTC.

The lower right part (when $l_a \geq N_c \wedge l_h \geq N_c$). In this scenario, the merchant has confirmed the transaction (as $l_h \geq N_c$). When $l_a > l_h$, the adversary can revert the honest blockchain and double-spend his money directly by **OVERRIDE** (i.e. publishing his branch). When $l_a < l_h$, the adversary's branch slightly falls behind the honest blockchain, so he can try to catch up by **WAIT_INC** (except when $\beta = 1.0$). When $l_a = l_h$, if $fork = r$ (i.e. the adversary has published his branch), the adversary's optimal action is **MATCH_INC** (except when $\beta = 1.0$). Meanwhile, if $fork = p$ (i.e. the adversary has not published his branch), the adversary's optimal action is **WAIT_INC** (except when $\beta = 1.0$). This is because when $l_a = l_h \wedge fork = r$ (i.e. the adversary's branch is published and its length is the same as the honest blockchain), the adversary lost control on his branch: he can only do **MATCH**-style actions but cannot do **WAIT**-style actions. Thus, the adversary can maximise the probability of success only by allocating more mining power to BCH. If $l_a = l_h \wedge fork = p$ (i.e. the adversary's branch is private and its length is the same as the honest blockchain), the adversary can keep waiting and increase the mining power to secure the attack.

Table 6: Data of BTC/BCH, ETH/ETC and XMR/BCN for experiments.

(a) BTC and BCH

	BTC	BCH
Difficulty	6071846049920.0	199070336984
Price (USD)	3585.99	126.31
Algorithm	Sha256d	Sha256d
Hashrate(h/s)	39997.52E+15	1444.26E+15
Coins per Block	12.5	12.5

(b) ETH and ETC

	ETH	ETC
Difficulty	1.91E+15	122025268093982
Price (USD)	118.53	4.26
Algorithm	Ethash	Ethash
Hashrate (h/s)	142.00E+12	8.62E+12
Coins per Block	2	4

(c) XMR and BCN

	XMR	BCN
Difficulty	113361254717.0	40879087965
Price (USD)	43.64	0.000619
Algorithm	CryptoNight	CryptoNight
Hashrate (h/s)	9.29E+08	3.35E+08
Coins per Block	3.075	987.26

Table 7: Data of 15 PoW blockchains and NiceHash prices.

	Rank	Rent($/h/s)	Coin Price($)	Hashrate	N_c
Bitcoin	1	2E-18	3585.99	4E+19	6
Ethereum	3	1.36E-13	118.53	142E+14	12
BitcoinCash	4	2E-18	126.31	1.44E+18	6
Litecoin	8	3.34E-14	30.84	2.77E+14	6
Monero	14	9.13E-11	43.64	9.29E+8	10
Dash	15	3.53E-16	71.79	2.32E+15	6
EthereumClassic	18	1.36E-13	4.26	8.62E+12	12
Zcash	20	1.38E-08	54.77	3.36E+9	6
Dogecoin	23	3.34E-14	0.002132	3.76E+14	6
BitcoinGold	26	1.38E-08	11.93	3170000	6
Siacoin	46	3.74E-17	0.002389	1.88E+15	6
Komodo	55	1.38E-08	0.640292	4.48E+7	30
Electroneum	67	9.13E-11	0.006184	4.4E+9	20
Ravencoin	94	3.36E-13	0.011905	5.9E+12	6
Zcoin	99	2.79E-12	4.83	9.69E+10	6

Table 8: Details of relevant blockchains and mining power prices at the time of attack (on 07/01/2019).

ETC Price	$5.32
BTC Price	$4061.47
Difficulty	131.80E+12
h_2	1.16
Coins per Block	4
Nicehash Price	3.8290 BTC/TH/day

C EXPERIMENTAL DATA

As mentioned in the main body, we present our collected data in the tables (Table 5 – Table9). We omit the detailed explaination on each table, as they are already referenced in the according sections when they are mentioned.

We fetched the blockchain data from Coinmarketcap [16] on 19 February 2019, and prices of renting mining power from NiceHash [5] on 07 April 2019. For analysing the recent 51% attack on ETC, we fetched the blockchain data from coinmarketcap [16], and the price of renting Ethash mining power from NiceHash [5], both at the time of the attack (on 07/01/2019).

Note that in table 9, the "portion" represents the ratio of a blockchain with more net mining power over the other blockchain, where the blockchain with more net mining power is the first row of each mining algorithm, and all other rows of the same mining algorithm from the other chain. For a chain with more net mining power, the "Top Miners" represents the percentage of mining power that the top mining pools control in this chain. For the chains with less net net mining power, the "Top Miners" show the ratio of mining power of a top miner over the total mining power of the chain. For example, the top 1 mining pool in ETH controls 27.7% mining power, and this amount of mining power about 4.563 times of the total mining power in the entire ETC network.

Table 5: Values of parameters for evaluating the 51-MDP model.

	Notation	Default	Permuted
Mining Status	D_1	100	N/A
	D_2	10	np.arange(5, 100, 5)
	h_1	0.1	N/A
	h_2	2.0	np.arange(1, 10, 1)
Incentive-Related Parameters	R_1	50	N/A
	R_2	5	np.arange(5, 50, 5)
	v_{tx}	100	np.arange(5, 100, 5)
Adversary Network	γ	0.3	np.arange(0.1, 1.0, 0.1)
the Vigilance of the Merchant	N_c	4	np.arange(1, 10, 1)
Mining Power Price	pr	2	np.arange(0.2, 4, 0.2)

Table 9: Summary of blockchains sharing the same mining algorithm.

Type	Mining Algorithm	Coin	Rank	Hashrate (h/s)	Portion	Top Miners		
						#1	#2	#3
ASIC-resistant	Ethash	Ethereum (ETH)	3	1.42E+14	N/A	27.7%	22.2%	12.5%
		EthereumClassic (ETC)	18	8.62E+12	1647.4%	456.3%	365.7%	205.9%
	CryptoNight	Monero (XMR)	14	9.29E+08	N/A	37%	26%	12%
		ByteCoin (BCN)	39	3.35E+08	277.3%	102.6%	72.1%	33.3%
	Equihash	Zcash (ZEC)	20	3.36E+09	N/A	33.4%	19.2%	17.8%
		BitcoinGold (BTG)	26	3.17E+06	111111.1%	37111.1%	21333.3%	19777.8%
		Komodo (KMD)	55	4.48E+07	7518.8%	2511.3%	1443.6%	1338.3%
		Aion (AION)	84	7.22E+05	1000000.0%	334000.0%	192000.0%	178000.0%
ASIC-friendly	Sha256d	Bitcoin (BTC)	1	4.00E+19	N/A	23%	16.4%	11.6%
		BitcoinCash (BCH)	4	1.44E+18	2777.8%	638.8%	455.6%	322.2%
	Scrypt	Dogecoin (DOGE)	23	3.76E+14	N/A	18.0%	16.0%	10.0%
		Litecoin (LTC)	8	2.77E+14	135.7%	24.4%	21.7%	13.6%
	X11	Dash (DASH)	15	2.32E+15	N/A	13.0%	11.0%	11.0%
		WaltonChain (WTC)	73	1.14E+15	203.5%	26.5%	22.4%	22.4%