Fact and Fiction: Challenging the Honest Majority Assumption of Permissionless Blockchains

Runchao Han[1,2], Zhimei Sui[1], Jiangshan Yu[1⋆], Joseph Liu[1], and Shiping Chen[2]

[1] Monash University
[2] CSIRO-Data61

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 miners in the PoW-based blockchains in reality. In particular, we show that *the current incentive mechanism may encourage rational miners to launch 51% attacks* in two cases. In the first case, we consider a miner of a stronger blockchain launches 51% attacks on a weaker blockchain, where the two blockchains share the same mining algorithm. In the second case, we consider a miner rents mining power from cloud mining services to launch 51% attacks. As 51% attacks lead to double-spending, the miner can profit from these two attacks. If such double-spending is more profitable than mining, miners are more intended to launch 51% attacks rather than mine honestly.
We formally model such 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 – first introduced by Bitcoin [32] allows distributed participants (aka. nodes) to agree on the same set of transactions. In Bitcoin, all transactions are organised as a *blockchain*, i.e., 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 computationally hard Proof-of-Work (PoW) puzzle. In PoW, the puzzle solver (aka. miner) needs to find a nonce to make the hash value of the block smaller than a target value.

The blockchain may have *forks*: miners may create different valid blocks following the same block. In Bitcoin, miners always choose the longest fork of its

⋆ Corresponding author.

blockchain in order to agree on a single fork. However, a fork that is currently longer may be reverted by another fork, and all transactions in the currently longer fork will be deemed invalid. This gives the attacker an opportunity to spend a coin in a fork, then creates another longer fork to erase this transaction. This is called *double-spending attack*. To launch a double-spending attack in PoW-based consensus, an attacker should have enough mining power to create a fork growing faster than the current one. This requires the attacker to control a majority of mining power in the network. Double-spending attacks using the majority of mining power is known as *51% attacks*.

Honest majority. To avoid 51% attacks, PoW-based consensus should assume the *honest majority*: the majority of mining power in the system follows the protocol. Otherwise, the adversary with the majority of mining power can launch 51% attacks. Such 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.

Fact and Fiction. Ideally, there is only one blockchain in the world, and all miners will participate in this blockchain. This makes controlling 51% mining power extremely difficult. However, there exists numerous PoW-based blockchains [13]. As the total available mining power is shared amoung different blockchains, no blockchain enjoys the ideal security guarantee.

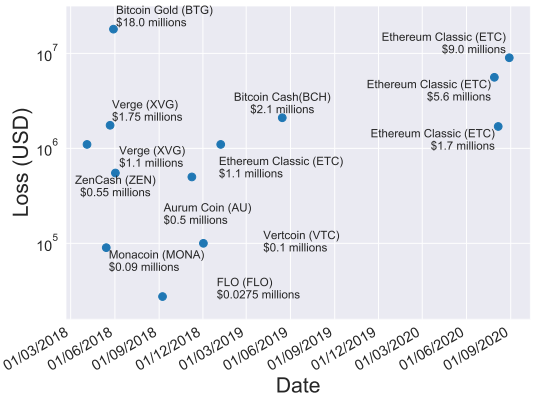


Fig. 1: 51% Attacks in 2018-2020 [25]. We omit the 51% attack on Litecoin Cash on June 4, 2018 as the loss is unknown.

The existence of multiple blockchains gives the opportunity to 51% attacks and makes the *honest majority* breakable. As shown in Figure 1, there have been several 51% attacks, causing the loss of more than $41 million. Most notably, within a month from 29/07/2020 to 29/08/2020, there were three huge 51% attacks on Ethereum Classic (ETC) [3–5], where the largest one reverted more than 8,000 blocks and caused the loss of $9.0 million.

Incentive and rationality. For better security guarantee, a PoW-based blockchain should attract miners to contribute mining power. To attract miners, PoW-based blockchains usually employ an incentive mechanism, where a miner creating a block will be rewarded for his contribution. Such incentive mechanism makes PoW-based blockchains to assume miners are *rational* [7], i.e., making decisions for profit. The frequent huge 51% attacks on ETC indicate that, miners' rational choice may not only be mining honestly, but can also be launching 51% attacks. This leaves us a question that, does the incentive mechanism really encourages miners to mine honestly and secures the blockchain?

1.1 Our contributions

While prior research [8, 11, 15, 16, 21, 22, 24, 26, 27, 33, 41, 45] analyse PoW-based blockchains as a stand-alone system, we analyse PoW-based blockchains in the precense of externally available mining resources. We formally analysing two variants of 51% attacks using externally available mining power, and show that 51% attacks are feasible and more profitable than honestly mining for most blockchains. Our analysis leads to two results: 1) the honest majority assumption does not hold for these blockchains, and 2) instead of encouraging miners to mine honestly, the incentive mechanism encourages miners to launch 51% attacks and break "honest majority". Specifically, we make the following contributions.

Two 51% attacks. We consider two variants of 51% attacks that make use of externally available mining power. One is *mining power migration attack*, where the adversary migrates mining power from a stronger blockchain to attack a weaker blockchain. The other is the previously known *cloud mining attack* [10], where the adversary rents mining power from cloud mining services (e.g., Nicehash [35]) to attack a blockchain.

Formalisation. Whether these two attacks are feasible or profitable are unknown. Straightforward estimations are coarse-grained so may lead to biased estimations and consequently wrong conclusions. To identify PoW-based consensus' (overlooked) weaknesses and provide insights and directions towards securing them, we formalise the two 51% attacks using 51-MDP– a MDP-based model extended from Gervais et al. [21]. 51-MDP takes parameters of blockchains and the adversary as input, and outputs the cost and reward of launching a 51% attack. Of independent interest, 51-MDP can be leveraged to formally study all attacks on PoW-based blockchains while considering external environment.

Evaluaion. We apply 51-MDP to evaluate two attacks on existing PoW-based blockchains. The results show that for most PoW-based blockchains, launching both 51% attacks is feasible and more profitable than hoenstly mining. For example, a miner with 12.5% mining power in Bitcoin can profit 6% ($18,946.5) more than honestly mining Bitcoin by double-spending a transaction of 3,000 BCH ($378,930) on BitcoinCash. The required mining power is not difficult to obtain – at the time of writing, F2Pool controls 17.7% mining power in Bitcoin [14].

Case study. We apply 51-MDP to study the 51% attack on ETC happened at 07/01/2019. On 07/01/2019, an anonymous attacker launched a series of 51%

attacks and double-spent more than $1.1 million on a cryptocurrency exchange Gate.io [20]. The attack is suspected to be a *cloud mining attack* using mining power from Nicehash [30]. We first analyse the pattern of double-spent transactions, and reveal the attacker's strategy for maximising and stabilising revenue. We then apply 51-MDP to reverse-engineer the attacker's revenue. The results show that, the attacker is expected to earn $84773.40, which is close to $100,000 – the attacker returned to Gate.io later [2]. This indicates the attacker was likely to launch 51% attacks in the fine-grained way described in our paper.

Countermeasures. We discuss potential countermeasures of these two 51% attacks derived from our observed insights. The recent recommended update aligns with our suggestions. Due to the page limit, we defer the discussion to Appendix D.

1.2 Paper organisation

Section 2 presents our 51-MDP model and its evaluation. Section 3 analyses the feasibility and profitability of the two 51% attacks. Section 4 studies the 51% attack on ETC in 2019. Section 5 analyses related work, and Section 6 concludes the paper. Appendix A summarises less related work. Appendix B provides supplementary details of 51-MDP and evaluation, and Appendix C analyses the attacker's strategy when attacking ETC. Appendix D discusses potential remedies of our attacks. Appendix E provides a study on the optimal strategy of a BTC miner to launch these two attacks on BCH. Appendix F presents all experimental data used in this paper.

2 Formalisation

We consider two 51% attacks that make use of externally available mining power: *mining power migration attacks* and *cloud mining attacks*. *Mining power migration attacks* use mining power from other blockchains, while *cloud mining attacks* use mining power from cloud mining services. We formally study these two 51% attacks by proposing a Markov Decision Process (MDP)-based model called 51-MDP. 51-MDP takes our defined blockchain parameters as input, and outputs an optimal attack strategy with expected revenue of this attack.

2.1 System model and notations

We assume miners are rational and blockchains may share the same mining algorithm. For simplicity, our model only considers two blockchains BC_1 and BC_2 with the same mining algorithm. 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. Let $d = \frac{D_1}{D_2}$ and $r = \frac{R_1}{R_2}$. As 51% attacks (on BC_2) are usually completed within a short time period, we assumes D_1, D_2, R_1 and R_2 remain stable during the attack.

In a *mining power migration attack*, the adversary migrates its mining power on BC_1 to launch 51% attacks on BC_2. Let $H_{a,1}, H_{a,2}$ be the adversary's mining

power, and $H_{h,1}, H_{h,2}$ be the honest mining power on BC_1 and BC_2, respectively. Let $H_a = H_{a,1}+H_{a,2}$, $H_h = H_{h,1}+H_{h,2}$, $H_1 = H_{h,1}+H_{a,1}$ and $H_2 = H_{h,2}+H_{a,2}$. Let $\beta = \frac{H_{a,2}}{H_a}$ be the fraction of mining power that the adversary allocates to BC_2. Let $h_1 = \frac{H_a}{H_{h,1}}$ and $h_2 = \frac{H_a}{H_{h,2}}$ be the ratio between the adversary's mining power and the honest mining power on BC_1 and BC_2, respectively.

In a *cloud mining attack*, the adversary rents mining power to launch 51% attacks on BC_2. We assume that the rentable mining power is compatible with the victim blockchain BC_2. We assume that the adversary has sufficient money for renting mining power. To keep notations consistent, we denote the rentable mining power as H_a. Thus, $h_2 = \frac{H_a}{H_{h,2}}$ is the fraction of rented mining power out of rentable mining power, and $\beta = \frac{H_{a,2}}{H_a}$ be the fraction of rented mining power out of the rentable mining power. Let pr be the price of renting a unit of mining power (e.g. hash per second) for a time unit.

Let $\gamma \in [0, 1]$ be the adversary's propagation parameter: when there are two simultaneous blocks mined by the adversary and an honest miner, γ of honest miners receive the adversary's block earlier than the honest block. Let N_c be the required number of blocks for the blockchain network to confirm a transaction.

2.2 The 51-MDP model

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

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

The 51-MDP model – summarised in Table 1 – 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. Formally, our 51-MDP model is a four-element tuple (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; and R is the reward matrix presenting the rewards of all state transitions.

State space S consists of four dimensions $(l_h, l_a, \beta, fork)$. Parameters l_h and l_a are the length of the honest and the adversary's forks on BC_2, respectively. Eventually, nodes will agree on only one of these two forks. Let $\beta \in [0, 1]$ be the ratio of mining power allocated on BC_2 out of the adversary's total mining power, and $\delta \in [0, 1]$ be the step of adjusting β. We denote the the state of the adversary's fork as $fork$, which has three possible values.

- **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.

Action space A includes actions that the adversary can perform given a state. The adversary's possible actions include:

- **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.
- **WAIT.** The adversary keeps mining on his fork. The adversary can perform **WAIT** in two scenarios. One is when $l_h < N_c$, i.e., the merchant is still waiting for the payment confirmation. The other is when **MATCH** has failed, i.e., $N_c < l_a \leq l_h$ but the adversary does not give up his fork.

When performing **MATCH** and **WAIT**, the adversary can adjust mining power allocated to BC_2. We denote two variants of **MATCH** as **MATCH_INC** and **MATCH_DEC**, where the adversary adds and reduces δh_2 mining power allocated to BC_2, i.e., $\beta \mapsto \{\beta + \delta, \beta - \delta\}$, respectively. Similarly, we denote two variants of **WAIT** as **WAIT_INC** and **WAIT_DEC**.

State Transition Matrix P is defined as a 3-dimensional matrix $S \times A \times S$: $Pr(s, a \mapsto s')$, where S is the state space, and A is the action space. Each point (s, a, s') means that, the participant at state $s \in S$ performs the action $a \in A$ to transit his state to $s' \in S$ with probability $Pr(s, a \mapsto s')$. An action a

transits a state s to one of multiple possible states $s'_1, s'_2, \cdots, s'_n$ with probability $Pr(s, a \mapsto s'_i)$, where $\sum_{i=1}^{n} Pr(s, a \mapsto s'_i) = 1$.

When $a =$ **WAIT[_INC, _DEC]**, the adversary is mining his fork alone, until either the honest miners or the adversary mine a new block. The probability of $l_a \mapsto l_a + 1$ (i.e., the adversary mines the next block) and $l_h \mapsto l_h + 1$ (i.e., the honest miners mine the next block) are

$$P(l_a \mapsto l_a + 1) = \frac{H_{a,2}}{H_{a,2} + H_{h,2}} = \frac{\beta H_a}{\beta H_a + H_{h,2}} = \frac{\beta h_2}{\beta h_2 + 1} \tag{1}$$

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

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

$$P(l_a \mapsto l_a + 1) = \frac{\beta H_a + \gamma H_{a,2}}{\beta H_a + H_{h,2}} = \frac{\beta h_2 + \gamma}{\beta h_2 + 1} \tag{3}$$

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

Reward Matrix R is defined as $S \times A \times S : Re(s, a \mapsto s')$, where the adversary performs action $a \in A$ which transits the system from state $s \in S$ to a new state $s' \in S$ while getting reward $Re(s, a \mapsto s')$. The reward is twofold: the reward of mining $\mathbb{R}_{mine}$ and the reward from the double-spent transaction $\mathbb{R}_{tx}$. The adversary also costs some money on the mining power, and we denote the cost as $\mathbb{R}^-$. Thus, $Re(s, a \mapsto s') = \mathbb{R}_{mine} + \mathbb{R}_{tx} - \mathbb{R}^-$.

$\mathbb{R}_{mine}$. The adversary receives 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[_INC, _DEC]** have a positive $\mathbb{R}_{mine}$, while $\mathbb{R}_{mine} = 0$ in 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[_INC, _DEC]**, the adversary needs to win the next block so that his blockchain overrides the honest one, leading to $\mathbb{R}_{mine} = (l_a + 1)R_2$.

$\mathbb{R}_{tx}$. Similar to $\mathbb{R}_{mine}$, the adversary receives the double-spent money only when his fork is published and accepted by the honest network. Therefore, $\mathbb{R}_{tx} = v_{tx}$ for **OVERRIDE** and the winning scenarios of **MATCH**-style actions, while $\mathbb{R}_{tx} = 0$ for other scenarios.

$\mathbb{R}^-$. We analyse the cost of *mining power migration attacks* $\mathbb{R}^-_{migration}$ and *cloud mining attacks* $\mathbb{R}^-_{cloud}$, separately. $\mathbb{R}^-_{migration}$ is the loss of block rewards from BC_1 due to the migrated mining power. Consequently, the cost can be computed as the mining reward of the migrated mining power on BC_1 during

the time of state transition. For **ADOPT** and **OVERRIDE** actions, state transitions take negligible time. For **WAIT**-style and **MATCH**-style actions, a state transition is triggered by a new block. Therefore, $\mathbb{R}^{-}_{migration}$ under **WAIT**-style and **MATCH**-style actions can be calculated as follows:

$$\mathbb{R}^{-}_{migration}(l_a \mapsto l_a + 1) = \mathbb{R}^{-}_{migration}(l_h \mapsto l_h + 1) \tag{5}$$

$$= -\beta H_a \cdot R_1 \cdot \frac{D_2}{H_{h,2} + \beta H_a} \cdot \frac{1}{D_1} \tag{6}$$

$$= \frac{-\beta h_2 R_1}{d(1 + \beta h_2)} \tag{7}$$

$\mathbb{R}^{-}_{cloud}$ is from renting cloud mining power. The price pr of renting cloud mining power is quantified as "the price of renting a unit of mining power for a time unit". Similar with the *mining power migration attack*, only **WAIT**-style and **MATCH**-style actions take a non-negligible time period. Therefore, $\mathbb{R}^{-}_{cloud}$ under **WAIT**-style and **MATCH**-style actions can be calculated as follows:

$$R_{BC_1}(l_a \mapsto l_a + 1) = R_{BC_1}(l_h \mapsto l_h + 1) \tag{8}$$

$$= -\beta H_a \cdot Pr \cdot \frac{D_2}{H_{h,2} + \beta H_a} \tag{9}$$

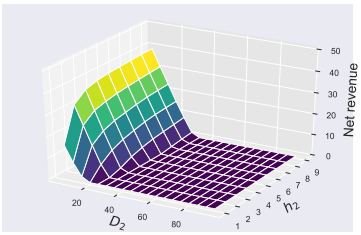
$$= \frac{-\beta h_2 D_2 Pr}{1 + \beta h_2} \tag{10}$$

2.3 Model evaluation

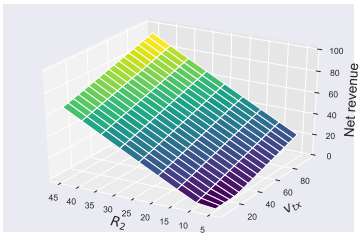
In order to identify the most important aspects on the profitability of 51% attacks, we use 51-MDP to evaluate our two 51% attacks. Together with public blockchain data, attackers can identify blockchains that are most profitable to attack, and defenders can prepare for potential 51% attacks in advance.

Experimental methodology. We implement 51-MDP using Python 2.7 and the *pymdptoolbox* library [12]. We give an upper bound $limit = 10$ for l_a and l_h. We choose $\delta = 0.2$, so the value of β can be $(0.0, 0.2, 0.4, 0.6, 0.8, 1.0)$. Concrete parameters for the evaluation are summarised in Table 5 of Appendix F. We apply the *ValueIteration* algorithm [40] with a discount value of 0.9 and an epsilon value of 0.1. We apply this discount value to encourage the adversary to finish the attack in a short time. 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. We choose a small discount factor to quantify such risk. We omit the evaluation of *cloud mining attacks* as both attacks share the same parameters D_2, h_2, R_2, v_{tx}, γ and N_c.

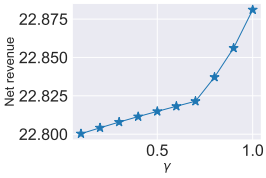
Evaluation results and analysis. Figure 2 shows the evaluation results. We defer detailed interpretation of evaluation results to Appendix B.2. To summarise, within these parameters, D_2, h_2, v_{tx}, R_2, N_c and pr have great impact the profit of 51% attacks, while γ has little impact on the profit.



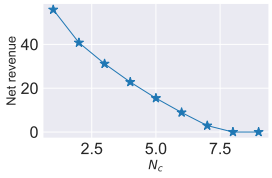
(a) Mining status h_2 and D_2.



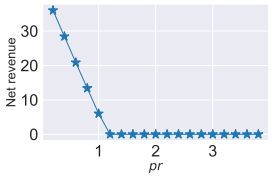
(b) Incentive v_{tx} and R_2.



(c) The adversary's network condition γ.



(d) Vigilance of the merchant N_c.



(e) Mining power price pr.

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

We observe some insights from the results. First, the attacker's profit is mainly affected by the parameters that are out of the attacker's control. The only important parameter that the adversary can control is v_{tx}, which is bound to his budget. Thus, to maximise the profit, an attacker should choose its target carefully. Once choosing the targeted blockchains, the adversary has little control over the attack.

In addition, although either the attackers or the defenders cannot fully control important parameters, monitoring them in real-time is possible. By monitoring these parameters, attackers can identify targets with most expected profit, and defenders (e.g. the cryptocurrency exchanges and the merchants) can be aware of potential attacks then perform countermeasures. For example, one of the effective countermeasures is to increase N_c, which greatly reduces the attacker's profit according to Figure 2d. In Appendix D.1 we analyse concrete countermeasures that defenders can perform.

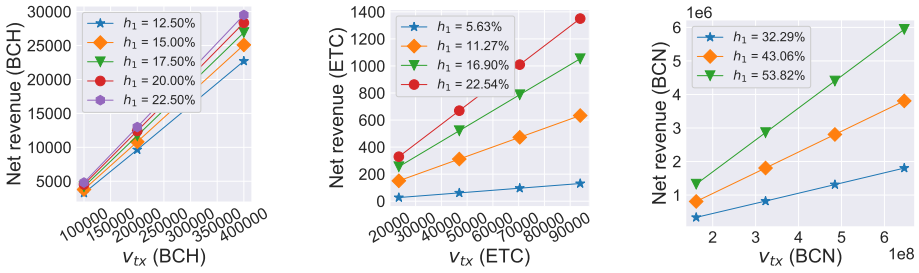
3 Evaluation of blockchains in the wild

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: 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 feasible and profitable on BTC/BCH and ETH/ETC, but it is not as effective on XMR/BCN.

In addition, we demonstrate an optimal strategy for a BTC miner to launch *mining power migration attacks* on BCH. We present The attack together with explanations and observed insights in Appendix E.

For the *cloud mining attack*, we evaluated the security of ten leading PoW-based blockchains. Our evaluation shows that the *cloud mining attacks* are feasible and profitable on most selected blockchains.

3.1 Mining power migration attacks



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

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

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 easy and profitable for a miner of BTC (or ETH) to launch a 51% attack on BCH (resp. ETC). In particular,

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

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%) [14]; and the top three mining pools in BTC are F2Pool (17.7%), Poolin (16.1%), BTC.com (11.9%) [9].

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. Meanwhile, 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.

3.2 Cloud mining attacks

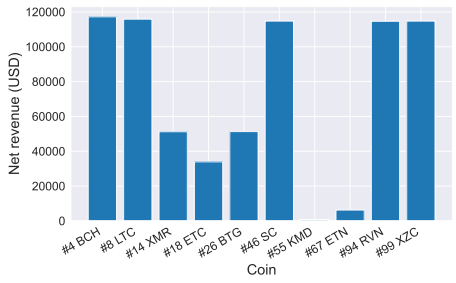


Fig. 4: Cloud mining attacks on selected 10 PoW blockchains. We use v_{tx} = \$500,000 , $h_2 = 2$ and $\gamma = 0.3$. We use the value of N_c recommended by cryptocurrency communities.

We evaluate ten leading PoW-based blockchains against the *cloud mining attack*. There are 22 PoW-based blockchains in the top 100 blockchains by market cap [13]. DigiByte and Verge use multiple mining algorithms simultaneously, and NiceHash does not support Bytom, ByteCoin, Electroneum, WaltonChain, and Aion. In addition, NiceHash does not have enough mining power to attack BTC with SHA256D, ETH with Ethash, ZEC with Equihash, DOGE with Scrypt and DASH with X11. Thus, we focus on analysing the rest ten 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 is twice of the honest mining power). We choose the value of N_c according to the recommended values from cryptocurrency community, as listed in Figure 7 in Appendix F.

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

- the attacker needs approximately \$2,000 to launch a *cloud mining attack* on ETC for an hour, and the net revenue will be \$33,899 if successful; and
- the attacker needs approximately \$2,600 to launch a *cloud mining attack* on BCH for an hour, and the net revenue will be \$117,198 if successful.

The only exception is Komodo (KMD): the attacker cannot profit much by launching *cloud mining attacks* on KMD. The reason is that the value of N_c recommended by the KMD community is 30 – much higher than other blockchains.

As shown in §2.3, increasing N_c can significantly reduce the profit of 51% attacks. We defer the detailed evaluation of KMD to Appendix B. In Appendix D, we will demonstrate in detail the impact of adjusting different parameters on the profitability of both attacks.

4 Case study: the 51% attack on Ethereum Classic

On 07/01/2019, a 51% attack happened to Ethereum Classic (ETC): the attacker double-spent transactions of more than $1.1 million on a cryptocurrency exchange Gate.io [20]. Though the mining power source remains unknown, the attack is highly suspected as a *cloud mining attack*. In this section, we investigate this 51% attack as a case of *cloud mining attacks*. We use 51-MDP to evaluate the attack and estimate the attacker's revenue. The evaluation result shows that the attacker launches the *cloud mining attack* in a fine-grained way, and obtains the theoretically optimal revenue from the attack. In Appendix C, we also analyse the attacker's behaviours, and show that the attacker's strategy is the best practice of launching *cloud mining attacks*.

4.1 The attack details

Ethereum Classic (ETC) is a PoW-based blockchain forked from Ethereum (ETH). In 07/01/2019, a 51% attack on ETC resulted in the loss of more than 1.1 million dollars. The attack lasted for 4 hours, approximately from 0:40 am to 4:20 am UTC, 07/01/2019. During the attack, the attacker repetitively created coin withdrawal transactions on the Gate.io cryptocurrency exchange [19] and launched double-spending attacks [20]. Among these attempts, 12 transactions were successfully double-spent (listed in Table 4 in Appendix F). Interestingly, the attacker later returned ETC equivalent to $100,000 back to Gate.io [2].

While the source of the mining power for this attack remains unknown, the NiceHash cloud mining platform [35] is highly suspected. One day before the attack, an anonymous person rented all available Ethash (the mining algorithm used by ETH/ETC) mining power from NiceHash [29, 31].

4.2 Evaluation

Table 8 in Appendix F summarises the attack-related data. According to Gate.io [19], during the attack's time period, $N_c = 12$ – the recommended value of the ETH community and ETC community [39]. The price of ETC and BTC was $5.32 and $4061.47, respectively. The mining difficulty of ETC was 131.80E+12, and the ratio h_2 was about 1.16. The block reward is 4 ETC coins, and the price of Nicehash mining power was 3.8290 BTC/TH/day. We keep assuming $\gamma = 0.3$ as there is no data on γ and the impact of γ is relatively small. Figure 5 shows our evaluation result. We 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.

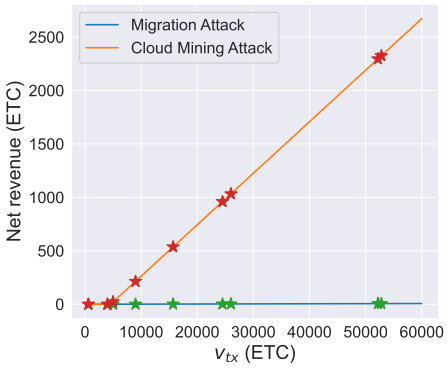


Fig. 5: Simulated 51% attack on ETC. The blue line denotes the relative reward of *cloud mining attacks*. The orange line denotes the relative reward of *mining power migration attacks* for making comparisons. We also marked different transaction amounts in the attack using dots.

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. In addition, *cloud mining attacks* are more profitable than *mining power migration attacks*. This means that renting mining power to attack ETC is much cheaper than migrating mining power from ETH. This is because both ETH and ETC use Ethash [38] as the mining algorithm. Ethash is a memory-hard function, making it GPU-friendly while ASIC-resistant [37]. Thus, any GPU can be used for mining ETH/ETC, making mining power much cheaper than that from dedicated hardware such as ASICs.

4.3 Estimating the attacker's net revenue

According to Table 4, the attacker has stolen 219,500 ETC, which is the attacker's gloss revenue. As we don't know transactions of failed 51% attack attempts, the cost of the attack is unknown. Thus, it's hard to determine the cost of attacks, and we cannot calculate the attacker's revenue directly. Nevertheless, we can apply 51-MDP to estimate the attacker's net revenue. As we know the amount of mining power of the attacker, we can estimate the success rate of attacks. With the success rate, we can estimate the total amount of transactions for failed attacks, and therefore derive the total amount of double-spending transactions. With the total amount and blockchain data as input, 51-MDP can estimate the attacker's net revenue. By using this method, we find that our estimated net revenue is approximately $84773.40, which is close to $100,000 – the value that the attacker returned to Gate.io after the attack [2].

Modelling. We first calculate the success rate of the attack. 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 $B(n_a + n_h, p)$ where n_a and n_h are the numbers of blocks that the adversary and the honest miners have 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

$$Pr(X = n_a) = Pr(n_a; n_a + N_c, p) \tag{11}$$

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

$$P = 1 - \sum_{n_a=0}^{N_c-1} Pr(n_a; n_a + N_c, p) \tag{12}$$

Then, we estimate the 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

$$\frac{R_s}{P} = \frac{R_s}{1-P} \Longrightarrow R_f = \frac{(1-P)R_s}{P} \tag{13}$$

and the estimated total net revenue R is

$$R = R_s + R_f = R_s + \frac{(1-P)R_s}{P} \tag{14}$$

Estimation. Summing profits of all successful transactions in Figure 5, the attacker's gloss revenue 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. Recall that $N_c = 12$ in ETC. From Equation 12, the success rate P of an attack can be calculated as

$$P = 1 - \sum_{n_a=0}^{N_c-1} Pr(n_a; n_a + N_c, p) \tag{15}$$

$$= 1 - \sum_{n_a=0}^{N_c-1} C_{n_a+N_c}^{n_a} p^{n_a} (1-p)^{N_c} \tag{16}$$

$$= 56.48\% \tag{17}$$

From Equation 14, we calculate the estimated net revenue R as

$$R = R_s + R_f = R_s + \frac{(1-P)R_s}{P} \tag{18}$$

$$= 9000 + \frac{(1-0.5648)\cdot 9000}{0.5648} \tag{19}$$

$$= 9000 + 6934.85 = 15934.85\ (ETCcoins) \tag{20}$$

Therefore, the attacker's net revenue is expected to be $9000 + 6934.85 = 15934.85$ ETC coins. At the time of attack, 15934.85 ETC coins is equivalent to $84773.40, which is slightly less than $100,000 – the amount that the attacker returned to Gate.io. To achieve the optimal revenue, the attacker should launch *cloud mining attacks* using the optimal strategy, which is usually fine-grained as shown in Table 3 in Appendix E. This indicates that, the attacker adopted a near optimal strategy for launching *cloud mining attacks*.

5 Related work

To our knowledge, we are the first to challenge the honest majority assumption of PoW-based blockchains in the presence of externally available mining power. Most existing papers [8, 11, 15, 16, 21, 22, 24, 26, 27, 33, 41, 45] analyse PoW-based blockchains while assuming the honest majority and omitting external factors. We summarise them in Appendix A. In this section, we compare our work with two closely related work, namely fickle mining [27, 42] and bribery attacks [10].

Fickle mining [27, 42] is that, a miner adaptively allocates mining power on two blockchains with the same mining algorithm (e.g., BTC and BCH) for extra profit. Similar to *mining power migration attacks*, fickle mining also consider miners' behaviours between multiple blockchains. While fickle mining assumes the honest majority and miners mine honestly, we consider the honest majority can be broken and rational miners can launch 51% attacks.

Bonneau et al. [10] introduce the family of bribery attacks, where an adversary bribes other miners and asks them to launch 51% attacks. They discuss two bribery attacks: one is our *cloud mining attack*, and the other is by creating a mining pool with negative fee. While Bonneau et al. [10] only informally discuss them, we formally study the *cloud mining attack* and additionally consider *mining power migration attack*. There have been new bribery attack vairants [22, 28, 43], where an adversary bribes miners to mine on a previous block and fork the blockchain. While their result is that rational miners can be bribed to break consensus, our result is that the incentive mechanism may encourage rational miners to break consensus.

6 Conclusion

In this paper, we challenge *honest majority* – the key assumption of PoW-based consensus. We propose the 51-MDP model to formalise two variants of 51%

attacks that use externally available mining power, and formally prove that the incentive mechanism in existing PoW-based blockchains usually encourage rational miners to launch 51% attacks rather than mine honestly. Of independent interest, 51-MDP can estimate the revenue of such 51% attacks, describes the attacker's strategy, and analyse attacks that consider external factors.

In the future, we will explore PoW-based consensus protocols that resist against 51% attacks using external mining power. A possible approach is to raise the threshold of 51% attacks by using accumulated historical reputation [44].

References

1. .: Security: Delayed proof of work (dpow) (2018), https://komodoplatform.com/security-delayed-proof-of-work-dpow/
2. .: Gate.io Got Back 100k USD Value Of ETC From The ETC 51% Attacker (2019), https://www.gate.io/article/16740
3. Ethereum Classic Suffers 51% Attack Again: Delisting Risk Amplified (2020), https://news.bitcoin.com/ethereum-classic-suffers-51-attack-again-delisting-risk-amplified
4. Hackers Launch Third 51% Attack on Ethereum Classic This Month (2020), https://decrypt.co/40196/hackers-launch-third-51-attack-on-ethereum-classic-this-month
5. Over $1M double-spent in latest Ethereum Classic 51% attack (2020), https://coingeek.com/over-1m-double-spent-in-latest-ethereum-classic-51-attack
6. Tweet of Ethereum Classic (2020), https://twitter.com/eth_classic/status/1299832466643931136
7. Aiyer, A.S., Alvisi, L., Clement, A., Dahlin, M., Martin, J., Porth, C.: BAR fault tolerance for cooperative services. In: SOSP (2005)
8. Arnosti, N., Weinberg, S.M.: Bitcoin: A natural oligopoly. arXiv preprint arXiv:1811.08572 (2018)
9. blockchain.com: Bitcoin Hashrate Distribution - Blockchain.info (2019), https://www.blockchain.com/en/pools
10. Bonneau, J.: 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. pp. 19–26 (2016)
11. Carlsten, M., Kalodner, H.A., Weinberg, S.M., Narayanan, A.: 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. pp. 154–167 (2016)
12. Chadès, I., Chapron, G., Cros, M.J., Garcia, F., Sabbadin, R.: MDPtoolbox: a multi-platform toolbox to solve stochastic dynamic programming problems. Ecography **37**(9), 916–920 (2014)
13. coinmarketcap.com: Top 100 Cryptocurrencies by Market Capitalization (2019), https://coinmarketcap.com
14. etherchain: Top Miners over the last 24h - etherchain.org (2019), https://www.etherchain.org/charts/topMiners
15. Eyal, I.: The Miner's Dilemma. In: 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015. pp. 89–103 (2015)

16. Eyal, I., Sirer, E.G.: 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. pp. 436–454 (2014)
17. Garay, J.A., Kiayias, A., Leonardos, N.: 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. pp. 281–310 (2015)
18. Garay, J.A., Kiayias, A., Leonardos, N.: 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. pp. 291–323 (2017)
19. gate.io: Gate.io - the gate of blockchain assets exchange (2019), https://www.gate.io
20. gate.io: Gate.io research: Confirmed the etc 51% attack and attacker's accounts - gate.io news (2019), https://www.gate.io/article/16735
21. Gervais, A., Karame, G.O., Wüst, K., Glykantzis, V., Ritzdorf, H., Capkun, S.: On the Security and Performance of Proof of Work Blockchains. In: CCS (2016)
22. Judmayer, A., Stifter, N., Zamyatin, A., Tsabary, I., Eyal, I., Gazi, P., Meiklejohn, S., Weippl, E.: Pay-to-win: Incentive attacks on proof-of-work cryptocurrencies. Cryptology ePrint Archive, Report 2019/775 (2019)
23. Kiayias, A., Russell, A., David, B., Oliynykov, R.: Ouroboros: A provably secure proof-of-stake blockchain protocol. In: Annual International Cryptology Conference. pp. 357–388. Springer (2017)
24. Kiffer, L., Rajaraman, R., Shelat, A.: 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. pp. 729–744 (2018)
25. komodoplatform.com: The Anatomy Of A 51% Attack And How You Can Prevent One (2019), https://komodoplatform.com/51-attack-how-komodo-can-help-prevent-one
26. Kwon, Y., Kim, D., Son, Y., Vasserman, E.Y., Kim, Y.: 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. pp. 195–209 (2017)
27. Kwon, Y., Kim, H., Shin, J., Kim, Y.: Bitcoin vs. bitcoin cash: Coexistence or downfall of bitcoin cash? In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 935–951. IEEE (2019)
28. Liao, K., Katz, J.: Incentivizing blockchain forks via whale transactions. In: International Conference on Financial Cryptography and Data Security. pp. 264–279. Springer (2017)
29. Messamore, W.: 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
30. Morris, D.Z.: The Ethereum Classic 51% Attack Is the Height of Crypto-Irony, https://breakermag.com/the-ethereum-classic-51-attack-is-the-height-of-crypto-irony
31. Morris, D.Z.: 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/

32. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
33. Nayak, K., Kumar, S., Miller, A., Shi, E.: 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. pp. 305–320 (2016)
34. Nesbitt, M.: Deep Chain Reorganization Detected on Ethereum Classic (ETC) (2019), https://blog.coinbase.com/ethereum-classic-etc-is-currently-being-51-attacked-33be13ce32de
35. nicehash: Nicehash - Largest Crypto-Mining Marketplace (2019), https://www.nicehash.com
36. Pass, R., Seeman, L., Shelat, A.: 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. pp. 643–673 (2017)
37. Ray, J.: Ethash Design Rationale (2018), https://github.com/ethereum/wiki/wiki/Ethash-Design-Rationale
38. Ray, J.: Dagger Hashimoto (2019), https://github.com/ethereum/wiki/wiki/Dagger-Hashimoto
39. reddit: 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
40. Ross, S.M.: Introduction to stochastic dynamic programming (2014)
41. Sapirshtein, A., Sompolinsky, Y., Zohar, A.: 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. pp. 515–532 (2016)
42. Spiegelman, A., Keidar, I., Tennenholtz, M.: Game of coins. arXiv preprint arXiv:1805.08979 (2018)
43. Winzer, F., Herd, B., Faust, S.: Temporary censorship attacks in the presence of rational miners. In: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 357–366. IEEE (2019)
44. Yu, J., Kozhaya, D., Decouchant, J., Verissimo, P.: Repucoin: Your reputation is your power. IEEE Transactions on Computers (2019)
45. Zhang, R., Preneel, B.: 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 (2019)

A Other related work

While our paper analyses PoW-based blockchains' security in the presence of externally available mining power, most existing research considers blockchains as a stand-alone system and omits external environment. We summarise existing efforts on analysing PoW-based blockchains as follows.

Formalisations of PoW-based consensus. Garay et al. [17] formalise Bitcoin's consensus under lock-step synchronous networks, and define two properties *common prefix* and *chain quality*. Pass et al. [36] extend this model to

synchronous networks. Garay et al. [18] further extend it with dynamic difficulty adjustment. Kiayias et al. [23] further defined another property on the liveness called the *chain growth*.

Evaluation frameworks of PoW-based consensus Gervais et al. [21] propose an MDP-based evaluation framework for quantitatively analyse PoW-based consensus security, with a focus on the resistance of selfish mining [16] and double-spending. Zhang et al. [45] 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 *incentive compatibility* (i.e. the net revenue lower bound of honest miners under selfish mining attacks), *subversion gain* (i.e., the profit upper bound of an adversary performing double spending), and *censorship susceptibility* (i.e., the profit loss of honest miners under censorship retaliation attacks).

Models of specific attacks on PoW-based consensus Most papers use MDP-based models [21, 24, 41, 45] or game-theoretic models [8, 11, 15, 16, 22, 26, 27, 33] to evaluate attacks on PoW-based consensus. Our 51-MDP model adopts the MDP-based approach. While similar with models in [21, 41] in terms of notations and processes, 51-MDP additionally considers externally available mining power. Our 51-MDP model is more complex than existing MDP-based models, as the number of parameters are doubled. We reduce the excessive parameters for simplifying the implementation and simulation of without losing correctness.

B Supplementary materials of 51-MDP and evaluation

B.1 Notations

Table 2 summarises all notations.

B.2 Detailed evaluation results of 51-MDP

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.

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

Table 2: Notations of parameters in 51-MDP

Symbol	Definition
BC_1, BC_2	The stronger blockchain and the weaker blockchain
D_1, D_2	Difficulty of BC_1 and BC_2
d	Fraction of BC_1's difficulty towards BC_2's difficulty, i.e., $d = \frac{D_1}{D_2}$
$H_{h,1}, H_{a,1}$	Honest and adversary's mining power on BC_1
$H_{h,2}, H_{a,2}$	Honest and adversary's mining power on BC_2
H_a, H_h	Total honest and adversary's mining power, i.e., $H_a = H_{a,1} + H_{a,2}$, $H_h = H_{h,1} + H_{h,2}$
h_1	Fraction of the adversary's mining power towards BC_1's honest mining power, i.e., $h_1 = \frac{H_a}{H_{h,1}}$
h_2	Fraction of the adversary's mining power towards BC_2's honest mining power, i.e., $h_2 = \frac{H_a}{H_{h,2}}$
R_1, R_2	Mining reward of a block on BC_1 and BC_2
r	Fraction of BC_1's mining reward of a block towards BC_2's, i.e., $r = \frac{R_1}{R_2}$
v_{tx}	Amount of the attacking transactions
γ	Propagation parameter of the adversary
pr	Renting price of a mining algorithm
β	Fraction of migrated mining power by the adversary
δ	Step of adjusting β
N_c	Number of blocks required to confirm a transaction

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.

B.3 Evaluation of KMD

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. The evaluation result in Figure 6 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, the *cloud mining attack* is still more profitable than the *mining power migration attack*. For the profitability, the reason is that KMD requires 30 blocks to confirm a transaction (i.e., $N_c = 30$) [1], which is a much higher requirement than other blockchains. As shown in §2.3, increasing N_c can significantly reduce the profit of 51% attacks.

C The attacker's strategy in the 51% attack on ETC

According to Table 4, 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 behaviour belongs to the strategies used by the attacker to maximise and stabilise his revenue, for the following reasons.

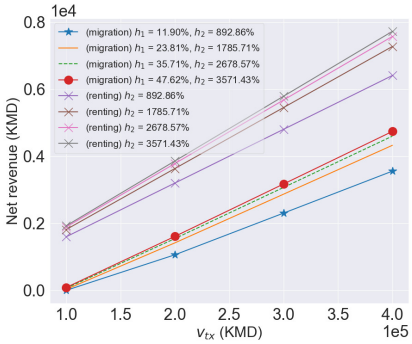


Fig. 6: Profitability of mining power migration attacks and cloud mining attacks on Komodo (KMD). We choose $\gamma = 0.3$, and $N_c = 30$ - the values recommended by KMD community.

Stabilising the revenue. First, launching multiple small double-spending attempts can stabilise the expected revenue. Double-spending attacks may fail 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.

Bypassing risk management systems. Second, this strategy may be used for bypassing risk management systems of cryptocurrency exchanges. Cryptocurrency exchanges run risk management systems to combat misbehaviours, including fraudulent payments and abnormal login attempts. A huge coin withdrawal transaction is very likely to trigger the risk management system, while multiple small transactions might be overlooked. In addition, a big transaction may lead to longer confirmation time, and a longer attack period is easier to be detected. Therefore, bypassing the risk management system is naturally a part of the attacker's strategy. According to the Gate.io report [20], 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. The attacker 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 large 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 his account. The first gap may be because the attacker was cautious when first launching the double-spending attack. The attacker double-spent a transaction of 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 it, 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 account 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, leading to the second time gap.

D 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 §4) 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 D.1.

D.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 detecting the two 51% attacks, the exchange can take several reactions to prevent them from happening. First, the

exchange can increase the number N_c of block confirmations. According to Figure 7, for the 51% attack on ETC in 2019, the attack can be avoided if increasing N_c to 18. The ETC community's action further proves the effectiveness of increasing N_c: after the last 51% attack [4], the ETC community urged to raise N_c to 10,000 [6], while it takes approximately two weeks to generate 10,000 blocks. Second, the exchange can decrease the maximum amount of cash out. Figure 2b and 7 show the impact of the transaction amount v_{tx} on the 51% attack on ETC. If the maximum amount of cash out was limited to 9,000 ETC (approximately $38340.0), then the attacker would no longer profit. Third, limiting the frequency of cash out also discourages 51% attacks. With a limited frequency of cash out, the attacker will need more time to launch attacks, and thus the attack takes more opportunity cost. Last, when the risk management system considers attacks are likely to happen, then the exchange can halt all cash out temporarily.

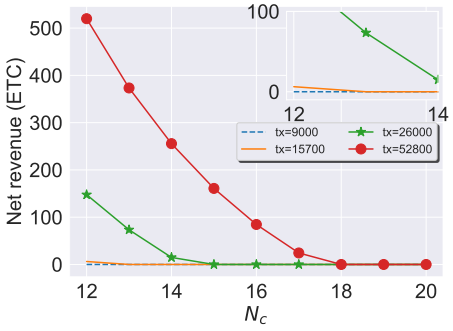


Fig. 7: Impacts of v_{tx} and N_c on the ETC attack.

D.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 minimise 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 [44] 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.

E Optimal strategy for BTC/BCH

Table 3: 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

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 §3. 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 [40] with a discount value of 0.9 and an epsilon value of 0.1 for 51-MDP.

Table 3 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 $\beta \geq \delta$).

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.

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.

F Experimental data

Table 5-9 summarise our collected data used in this paper. We fetched the blockchain data from Coinmarketcap [13] on 19 February 2019. We fetched prices of renting mining power from NiceHash [35] on 07 April 2019. For analysing the 51% attack on ETC, we fetched the attack details from [20], the blockchain data from coinmarketcap [13], and the price of renting Ethash mining power from NiceHash [35] at the time of the attack 07/01/2019. Table 4 summarises all double-spent transactions during the 51% attack on ETC [34].

In table 9, the "portion" represents the ratio of a blockchain with more mining power over the other blockchain, where the blockchain with more 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 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 mining power, the "Top Miners" show the ratio between a top miner's mining power and the blockchain's total mining power. 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 4: All 12 double-spent transactions during the 51% attack on ETC [34]. Transaction IDs and addresses are shortened.

Trans. ID	**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

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

	Notation	Default	Permuted
Mining Status	D_1	100	N/A
	D_2	10	$\{5, 10, \ldots, 100\}$
	h_1	0.1	N/A
	h_2	2.0	$\{1, 2, \ldots, 10\}$
Incentive-Related Parameters	R_1	50	N/A
	R_2	5	$\{5, 10, \ldots, 50\}$
	v_{tx}	100	$\{5, 10, \ldots, 100\}$
Adversary Network	γ	0.3	$\{0.1, 0.2, \ldots, 1.0\}$
the Vigilance of the Merchant	N_c	4	$\{1, 2, \ldots, 10\}$
Mining Power Price	pr	2	$\{0.2, 0.4, \ldots, 4\}$

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

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%