Security-Cost Efficiency of Competing Proof-of-Work Cryptocurrencies*

Kohei Kawaguchi†
Shunya Noda‡

First Version: November 30, 2021
Current Version: December 27, 2021

Abstract

Proof-of-Work cryptocurrencies consume vast energy to rule out potential attacks. Therefore, evaluating and improving the security-cost efficiency, the cost of attacking the system per unit of operating cost, is important. In this paper, we demonstrate that the stability of miners' supply of work over time is essential for the security-cost efficiency, and it is determined by the Difficulty Adjustment Algorithm (DAA) of the currency and the reward elasticity of the miner's supply of work. To this end, we develop a model of the multicurrency mining market and estimate the own- and cross-elasticity of the hash supply to the reward by exploiting the reward shock event, called halving. We use the estimated model to simulate the mining market and evaluate the security-cost efficiency. We find that Bitcoin is stable because of the inelastic miners, regardless of the DAA, whereas other smaller coins face highly elastic miners and can be stable only with efficient DAAs. Upgrading all relevant currencies' DAAs to the state-of-art one substantially improves the security-cost efficiency and saves the energy-consumption rate by 0.21 GW or 3.2% while maintaining the security level.

Keywords: Blockchain; Cryptocurrency; Proof-of-Work; SHA-256; Bitcoin; Security-Cost Efficiency; Mining Market; Algorithmic Competition; Hash Supply Elasticity

*This study is supported by the Citibank Endowment Fund (CEF20BM01), the Social Sciences and Humanities Research Council, and the Digital Economy Project at the University of Tokyo, funded by the Silicon Valley Community Foundation. The views are our own. We thank the comments of Yuichiro Kamada, Fuhito Kojima, Kosuke Uetake, Yasutora Watanabe, and participants at the Happy Hour Seminar! and the Digital Currency Workshop. We appreciate Yoshinori Hashimoto's expert advice on cryptocurrency. We thank Viky Choi, Parco Wong, Qichao Wang, and Haruo Kakehi for providing excellent research assistance.

†Contact: kkawaguchi@ust.hk, Hong Kong University of Science and Technology

‡Contact: shunya.noda@e.u-tokyo.ac.jp, The University of Tokyo and the University of British Columbia.

1 Introduction

Cryptocurrency (Nakamoto, 2008) is a decentralized transaction system for which security is guaranteed by cryptographic technology and incentive design. In cryptocurrency, agents called *miners* validate new transaction requests. A *Proof-of-Work (PoW)* cryptocurrency (Dwork and Naor, 1992; Back, 2002) prevents a miner from exploiting the system by randomly choosing the miner who takes the validation task. To implement the random assignment, PoW cryptocurrency elects a miner as the next validator with a probability proportional to the computational labor the miner supplies. The amount of computational labor provided in unit time is called the *hash rate*. A miner can dominate the validation task and exploit the system by supplying a large hash rate relative to the currency's total hash rate, while the provision of a large hash rate incurs large energy consumption (i.e., electricity cost). As such, the security of a PoW cryptocurrency is increasing in its total hash rate, and PoW inevitably consumes vast energy to rule out potential attacks.[1] Cambridge Centre for Alternative Finance estimates that in November 2021, the energy-consumption rate of Bitcoin, the largest and oldest cryptocurrency, is more than Netherlands' national electricity consumption.[2] Therefore, it is urgent to evaluate PoW cryptocurrencies' *security-cost efficiency*, which we define as the ratio of the cost of successfully attacking the system to the energy consumption for operating the system.

The hash rates change over time according to the market environment. Consequently, the cost (energy consumption) of operating currency for a period is proportional to the *average* hash rate, whereas the security level during the period is proportional to the *minimum* level of the hash rate because attackers can aim at the timing when the currency's hash rate is the lowest. If the hash rate is unstable, the minimum relative to the average hash rate is low, resulting in low security-cost efficiency. Accordingly, for evaluating the efficiency of PoW cryptocurrencies, we need to study how the hash rates respond to the market environment and when they become unstable.

PoW cryptocurrency uses the *Difficulty Adjustment Algorithm (DAA)* to determine the reward level as a function of the past transaction speed. Miners can choose which cryptocurrency to contribute by observing each currency's reward level. Thus, cryptocurrencies algorithmically compete for miners' hash supply in the mining market.[3] In 2021, numerous independent cryptocurrencies are operating using various types of DAAs. It is

[1] Alternative mechanisms, such as *Proof-of-Stake (PoS)*, have been proposed (Hinzen, John, and Saleh, 2019; Saleh, 2020; Roşu and Saleh, 2021). PoS is gathering attention because PoS can be superior to PoW in energy consumption. However, no conclusion has been reached concerning the consequence of PoS. Although some progressivist currencies, such as Ethereum, have already deployed PoS by 2021, PoW remains the most popular consensus mechanism.

[2] See `https://ccaf.io/cbeci/index`, accessed on November 28, 2021.

[3] This is as if firms compete for workers' labor supply by adjusting the wage in the labor market. The cryptocurrencies are firms, the miners are workers, and the rewards are wages. The miners can switch the currency to contribute as if workers decide for which company to work by observing firms' offers.

how the DAAs adjust the reward level and how elastically miners respond to changes in the rewards of competing currencies that characterizes the stability of the hash supply.

In this paper, we investigate (i) what determines the security-cost efficiency of PoW cryptocurrencies, and (ii) how we can improve the efficiency. Specifically, we demonstrate that the dynamic interaction between cryptocurrencies' algorithmic competition and the reward elasticity of the miners' hash supply crucially shapes the security-cost efficiency, and the efficiency can be substantially improved by changing the DAA profile. To this end, we develop a model of the mining market and a method to estimate miners' aggregate hash supply. We focus on the largest mining market that comprises *Bitcoin (BTC)*, *Bitcoin Cash (BCH)*, and *Bitcoin Satoshi Vision (BSV)*. We estimate the own- and cross-elasticity of the miners' hash supply to the short-time reward changes by exploiting an event called *halving*. This event is ideal for identifying the hash supply elasticity because it provides a large, predetermined, and discontinuous change in the reward of each currency and happens independently across currencies in a short time window. We exploit the third halving that sequentially arrived for BCH, BSV, and BTC in the spring of 2020.

We then use the estimated model to simulate the mining market after halving and evaluate the security-cost efficiency of cryptocurrencies under this large reward shock. Importantly, in contrast to the previous studies that have evaluated the security of cryptocurrencies assuming a fixed hash rate, we fully take into account the endogenous and dynamic changes of miners' hash supply. We define a novel security-cost efficiency measure, *Security per Energy Consumption (SpEC)*, as the ratio of the minimum to the average hash rate. We evaluate the SpEC of different DAAs that have been adopted by the SHA-256 cryptocurrencies, including the BTC's *original DAA*, BCH's and BSV's *CW-144* (until November 2021 for BCH), and BCH's *ASERT* (since November 2021). ASERT is acclaimed to outperform the others, but there has been no formal test of this claim. We calculate how much energy is wasted due to the inefficient DAA for a given level of security.

We find that a currency's hash rate responds positively to its own reward in the short run, whereas it does negatively to its rival currency's reward. Thus, miners supply their work in the most profitable currencies, and the cryptocurrencies are tightly connected through the mining market.

We estimate that BTC's own elasticity of hash supply is 0.63, whereas the own elasticities of BCH's and BSV's hash supply are 5.4 and 4.9, respectively. Thus, BTC, the currency with the longest history and the largest transaction volume, has the drastically lower hash supply elasticity than the newer and smaller currencies, BCH and BSV.

The estimated own elasticities reveal how these currencies maintain the stability of their hash rate. Noda, Okumura, and Hashimoto (2020) show that the original DAA (adopted by BTC) stabilizes the hash rate if and only if the own elasticity is smaller than 1. Thanks to the inelastic hash supply, BTC meets this criterion. Noda et al. (2020)

also show that CW-144 (adopted by BCH and BSV at the timing of the third halving) stabilizes the hash rate if and only if the own elasticity is smaller than 144. BCH's and BSV's own elasticity lies between 1 and 144. Therefore, BCH and BSV could not survive if they adopted the original DAA.

Noda et al. (2020) derive these thresholds of 1 and 144 in the single-currency model. The situation is worse in the multicurrency environment. We find the cross-elasticities of BCH's and BSV's hash supply to BTC's reward are -4.0 and -3.2, cross-elasticity of BCH's hash supply to BSV's reward is -1.5, and cross-elasticity of BSV's hash supply to BCH's reward is -1.2. Therefore, BCH and BSV are highly sensitive to external shocks and amplify the shock against each other. On the other hand, cross-elasticities of BTC's hash supply to BCH's and BSV's reward are only -0.2. Thus, BTC is substantially more resilient to external shocks. These findings underscore the importance of considering the connection through the miner's market for creating a functioning transaction system by PoW.

Using the estimated hash supply elasticity, we run counterfactual simulations to evaluate the security-cost efficiency of various DAA profiles. We start the simulation from just before the third BTC's halving for 60 days. After confirming that the estimated model replicates the actual mining market's pattern, we first study the effect of upgrading BTC's DAA to CW-144. We find that upgrading BTC's DAA does not substantially affect these three currencies' hash rates because BTC's hash supply is inelastic and BCH's and BSV's DAA absorbs any effect of BTC's DAA change.

We then study the effect of downgrading BCH's and BSV's DAA to the original DAA. We find that BCH and BSV collapse after the BTC halves if they use the inefficient original DAA. Moreover, the adoption of the inefficient original DAA poses a negative externality on the stability of other currencies. We find, by contrast, that if BCH replaces CW-144 with ASERT, it can further stabilize the system. Moreover, it also stabilizes BTC's and BSV's hash rate, creating a positive externality.

We measure the security-cost efficiency of cryptocurrencies under various DAA profiles in the period after BTC halving. We find that a currency can achieve a higher SpEC if its hash supply is inelastic. Regardless of the choice of DAAs, BTC, which has an inelastic hash supply, achieves a higher SpEC than BCH and BSV. We also find that a currency should abandon the original DAA and adopt ASERT to achieve a higher SpEC. The values of SpEC are consistent with the aforementioned analysis based on counterfactual simulations, providing convenient sufficient statistics of security-cost efficiency.

It is also suggestive to evaluate the security-cost efficiency by the amount of wasted energy. From the hash rate and SpEC of each profile, we can calculate the energy consumption that could be saved by changing the DAA profile from the actual one to an alternative one while maintaining the security level. Our estimate shows that by upgrading all DAAs to ASERT, we could save on average 0.21 GW or 3.2% in the period of

simulation. Although the change of SpEC is slim for BTC, the largest energy-saving is from BTC due to its size.

Cryptocurrency has been attracting economists' attention because of its various features that could enhance social welfare compared to traditional transaction systems. The decentralized transaction system may outperform the traditional ones under some circumstances (Chiu and Koeppl, 2019; Cong and He, 2019; Matsushima and Noda, 2020), or it may prevent the monopoly of a private firm operating the transaction system (Huberman, Leshno, and Moallemi, 2021). This paper contributes to the literature by evaluating and proposing the way for improving the cost efficiency of PoW cryptocurrencies.

It is well-known that a cryptocurrency system becomes less secure as the hash rate becomes lower. When the aggregate hash rate is low, an attacker can easily gain a 51 percent share to perform a double-spend attack (for the details about such attacks, see, for example, Ch. 5 of Narayanan, Bonneau, Felten, Miller, and Goldfeder, 2016).[4] Some of the other types of attacks, such as selfish mining (Eyal and Sirer, 2018) and smart/smarter mining (Goren and Spiegelman, 2019; Fiat, Karlin, Koutsoupias, and Papadimitriou, 2019), can be profitable even when attackers have a smaller share. We formalize this idea to define SpEC. Moreover, we are the first to associate a currency's security level with the currencys' algorithmic competition and miner's endogenous response in the mining market. We contribute to this literature by quantifying the sensitivity of the security level and security efficiency to the mining market competition.

The literature has not sufficiently uncovered the dynamic interaction between the cryptocurrencies' DAAs and the miners in the mining market. For example, Noda et al. (2020); Prat and Walter (2021); Shibuya, Yamamoto, Kojima, Shi, Matsuo, and Laszka (2021) focus on the BTC market and preclude other related currencies from consideration. Aggarwal and Tan (2019) develop a structural model of miners' choice for BTC and BCH mining and estimate miners' payoff function. They focus on the miners' strategic response to the emergency difficulty adjustment algorithm, a DAA employed by BCH in the first 4 months after the hard fork from BTC. Because they do not specify the payoff as a function of the winning rate, the model does not capture the dynamic interaction between the difficulty adjustment and hash supply, which is the key for the assessment of the security-cost efficiency in this paper.

[4]Indeed, several cryptocurrencies have experienced double-spending attacks. For example, Ethereum Classic was attacked in January 2019 (see https://www.bnnbloomberg.ca/ethereum-classic-movements-halted-by-coinbase-on-signs-of-attack-1.1194689, accessed on November 28, 2021) and Bitcoin Gold was attacked in May 2018 (see https://cointelegraph.com/news/bittrex-to-delist-bitcoin-gold-by-mid-september-following-18-million-hack-of-btg-in-may, accessed on November 28, 2021) and January 2020 (see https://cointelegraph.com/news/bitcoin-gold-blockchain-hit-by-51-attack-leading-to-70k-double-spend, accessed on November 28, 2021). More recently, BSV was attacked in July 2021. See https://coingeek.com/bitcoin-association-statement-zero-tolerance-for-illegal-attacks-on-the-bitcoin-sv-network/, accessed on November 28, 2021.

2 Institutional Details

2.1 Proof-of-Work Mechanics

Cryptocurrency systems store the transaction data as *blockchains*. A blockchain is a growing list of *blocks*, where a block is a collection of transactions validated by a miner. In decentralized cryptocurrency systems, anyone can work as a miner. If a single miner can create blocks too frequently, then the miner, who might be malicious, can have the power to take advantage of the system.

One way to solve this problem is to introduce a consensus mechanism called *Proof-of-Work* (PoW). The PoW consensus mechanism ensures the random selection of a block creator from active miners by assigning to miners a cumbersome task such that the probability of completing it is proportional to the computational cost expended. Because it is impossible to misinform the computational cost expended, this mechanism prevents a miner from dominating the block creation by cheating. It safeguards the cryptocurrency system as long as the fraction of malicious miners is small.

Specifically, a PoW consensus mechanism requires miners to evaluate a (cryptographic) *hash function*. The hash function is a function that maps data of arbitrary size to fixed-size values (called *hash values*). The function is designed so that a small change to the data changes the hash value extensively. As a consequence, the new hash value looks uncorrelated with the old hash value. Therefore, no one can predict the hash value unless computing the hash function. BTC, BCH, and BSV adopt *SHA-256* as a hash function.

Miners iteratively compute the hash value associated with the block data, which contains a field called *nonce*. A miner can change the nonce without changing the other block information to compute a hash value. A miner tries to find a nonce that returns a small enough hash value by changing the nonce to evaluate the hash function. By the nature of the hash function, the only way to achieve this is trial and error.

The cryptocurrency system sets a threshold value, called *target*. Once a miner finds a nonce for which the hash value of the block becomes smaller than the target, then the miner is allowed to append the block to the blockchain and obtains a *prize*. The prize is the seigniorage provided by the system and the transaction fees paid by users.

The cryptocurrency system adjusts the target by the *Difficulty Adjustment Algorithm (DAA)*. Each currency adopts a different DAA. BCH changed its DAA a few times in its history. We describe the details of DAAs in Section 2.6.

Since the hash value associated with a block data is ex ante unpredictable, computing a hash value once is equivalent to drawing a lottery. From a miner's viewpoint, the hash value returned by SHA-256 is a (pseudo) random variable that follows a discrete uniform distribution whose support is $\{0, 1, \ldots, 2^{256} - 1\}$. Then, the probability that a miner

successfully creates a new block in a trial (one hash computation) is (target)/2^{256}. We refer to this probability as the *winning rate*.

Let $w(t)$ be the winning rate of the currency in time t, and $h(t)$ be the *hash rate*, the number of hash computations done for the currency in a unit time. The winning rate is tiny and the hash rate is substantial. As a result, the block arrival for this currency is (accurately) approximated by a nonhomogeneous Poisson process where its time-t intensity is $w(t)h(t)$.

2.2 Mining

When a miner produces a new block (i.e., wins a lottery generated with PoW), the miner is rewarded with the cryptocurrency he contributed. The *prize* comprises the seigniorage paid with newly minted coins and the transaction fees paid by users. The amount of seigniorage is predetermined and is halved for every 210,000 blocks (see Section 2.5 for the detail). The seigniorage has been substantially greater than the transaction fee so far.

Miners use computers specialized in the computation of a specific hash function exclusively. Such computers are called a *mining ASIC (application-specific integrated circuit) machine*. Mining ASIC machines exist for some hash functions but not for others. For computing SHA-256, ASIC is predominant, and mining is profitable only when miners use mining ASIC machines (Taylor, 2017).

Mining machines are often compatible with multiple cryptocurrencies. For example, most SHA-256 ASIC machines list BTC, BCH, and BSV as "minable currencies" in their advertised specs. Each cryptocurrency determines the target and prize independently and its price (the exchange rate against fiat money) varies over time. Therefore, a miner's profit from investing his hash power into a cryptocurrency also varies over time. In addition, miners can shut down their machines when the variable cost of mining (mostly the electricity cost) is greater than the return. Accordingly, a miner, who already owns a mining machine, should select whether to operate and which currency to contribute dynamically.

2.3 Security Risk

Cryptocurrencies are designed on the premise that the "majority of miners" will adhere to the record-keeping protocol. Since cryptocurrency is a decentralized payment system, the system inevitably collapses if the "majority of miners" have the intention to attack the system and deviate from the protocol. This attack is called the *51% attack*. To be more precise, under PoW, an attacker needs to control the majority of the hash power on the network for a successful attack. Accordingly, the cost of carrying out this attack grows linearly with the aggregate hash rate of the target cryptocurrency.

Even when an attacker's hash rate is smaller than 50%, as long as the miner is significantly large, he can manipulate the system to make unfair profits. For example, Eyal and Sirer (2018) propose *selfish mining*, an attack to cause an "accidental" fork intentionally by keeping discoverd blocks private. Such a strategy forces honest miners to waste their hash power and enables the attacker to occupy a larger share of the hash rate compared with her/his actual hash power, leading to a larger profit.[5] Eyal and Sirer (2018) demonstrate that the BTC mining protocol (which is also adopted by BCH and BSV) is never safe if an attacker commands more than one third of the aggregate hash rate. Goren and Spiegelman (2019) and Fiat et al. (2019) propose *smart/smarter mining*,[6] an attack to shirk mining periodically to increase the winning rate to earn a larger profit. They show that an attacker can exploit the original DAA to make a profit even when her/his hash power is significantly smaller than one-third of the aggregate hash rate.[7] To make matters worse, when the attacker starts to perform smart/smarter mining, other miners will have incentives to join this attempt.

Because of these security risks, the aggregate hash rate has been regarded as an important indicator of the security level of the cryptocurrency. In Section 8, we formalize this notion to develop a security efficiency measure of cryptocurrency under various stress scenarios.

2.4 Bitcoin and Its Fork Currencies

In this paper, we focus on BTC and its fork currencies, BCH and BSV. We describe the history of these currencies.

Bitcoin (BTC) In early 2009, Nakamoto (2008) released BTC as the first cryptocurrency. The history of cryptocurrency evolves when a miner appends a new block. Each cryptocurrency has a rule that defines a valid block, and the blocks that violate the rule will be ignored by the community of miners. Consequently, all transactions recorded on invalid blocks are not regarded as processed transactions by the users.

Miners do not always agree on the rule. When a system upgrade is proposed, miners often disagree. As a dispute is provoked, the cryptocurrency community first attempts to settle it by conversation and voting. However, the negotiation occasionally breaks down and miners end up with two separate rules. Such events are called *hard fork*s. Blocks produced by one rule are not regarded as valid blocks by the other rule. After a hard fork,

[5] Shibuya et al. (2021) argue that the elastic hash supply exacerbates selfish mining. When selfish mining decreases the expected reawrd paid for honest miners, and therefore, the success probability of the selfish-mining attack is larger when the hash supply is elastic.

[6] Goren and Spiegelman (2019) and Fiat et al. (2019) are concurrent works about a similar attack, and smart/smarter mining is named by Goren and Spiegelman (2019).

[7] Profitability of smart/smarter mining crucially depends on the variable cost of mining, and therefore, the threshold hash share is not proposed as a scalar.

the two different rules generate two different transaction histories. The original currency is split into two different currencies. Since its launch, BTC has experienced a number of hard forks.

Bitcoin Cash (BCH) The largest fork currency, BCH, was hard-forked from BTC in August 2017. The community proposed two different upgrades to increase the ability to handle transactions. The first proposal was to adopt SegWit,[8] which aimed to reduce the data size of each transaction. The second proposal simply suggested increasing the block size. The community disagreed; those who followed the first proposal implemented SegWit on BTC, and the others initiated a hard fork and produced a fork currency, BCH. BCH allowed blocks of 8 MB (whereas BTC's original block size was 1 MB) and did not adopt the SegWit protocol. Later, in May 2018, BCH quadrupled its block size to 32 MB.

Bitcoin SV (BSV) In November 2018, the BCH community was about to enable an opcode for supporting the usage of smart contracts. A faction of the community rejected this proposal, insisting that such additions would ruin the vision of Satoshi Nakamoto (Nakamoto, 2008). They initiated another hard fork, and BCH was split into BCH and BSV, where SV stands for Satoshi Vision. BSV disabled the number of transactions that had been introduced after the launch of the original BTC, while the block size limit was increased to 128 MB.

In the beginning, these two factions attempted to prevent a currency fork by wiping out the blockchain produced by the other faction. To this end, BCH and BSV did not dare to implement *replay protection*, a protocol that enables users to send transaction requests to only one of these two currencies. In other words, at that time, when a user wanted to send 1 BCH to another user, she or he must also have sent 1 BSV to the same person. This period is called the *hash war*. During the period of the hash war, BCH and BSV were not truly two different currencies, but a single currency with ongoing factional strife. Miners had an incentive to support the currency they themselves supported, thereby preventing the other currency from continuing. Therefore, the structure of the miner's profit maximization problem was very different from that of normal times. The hash war started on November 15, when BCH and BSV forked, and ended on November 23, when BSV declared that it would implement replay protection and agreed to become a fork currency independent of BCH. Since then, BTC, BCH, and BSV have continued to coexist as independent currency systems, although they share the same origin.

We focus on the BTC and its forks for a number of reasons.

[8] See https://bips.dev/141/, accessed on November 28, 2021.

Economic Significance These currencies are economically significant. While various cryptocurrencies have been proposed, BTC still makes up a dominant share of the cryptocurrency market; as of November 2021, the market capitalization of BTC is $1 trillion, while that of Ethereum, the second-largest cryptocurrency, is $500 billion. Furthermore, the other two focused currencies, BCH and BSV are also relatively large—among all PoW currencies, BCH and BSV have the fourth and eighth largest market capitalization, respectively. The markets of cryptocurrencies that adopt other hash functions are significantly smaller.

Exhaustiveness While there also exist other cryptocurrencies that use SHA-256 as their hash function, their market capitalization is negligible. For example, in July 2021, the fourth largest SHA-256 cryptocurrency (after BSV), *Peercoin*, only had $30 million market capitalization, which was less than 0.01 percent of the BTC market capitalization. Since SHA-256 cryptocurrencies other than BTC, BCH, and BSV are all too small for industrial miners to mine, we can safely exclude them from the miners' choice set.

Similarity Since these three currencies share the same origin, they have a similar structure. In particular, they use the same (i) mining puzzle, (ii) specification of block headers, and (iii) targeted block arrival rate. Therefore, from the perspective of miners, it should be easy to switch from one currency to another. Indeed, some mining pools, joint groups of miners who combine miners' resources for risk sharing, provide options to automatically mine the most profitable currency among these three.[9] Moreover, this feature enables us to easily compare DAAs adopted by these currencies.

ASIC Dominance All of these three currencies use SHA-256 as their hash function, for which mining ASIC machines are by far more efficient than all-purpose computers. Thanks to this feature, we can safely assume that miners use some of the mining ASIC machines, for which we can obtain detailed data about advertised specs. This feature enables us to evaluate energy consumption (W) spent supplying a unit hash rate (H/s).

2.5 Halving

To prevent inflation, these three currencies limit the total amount of coins that will be minted in the long run to 21 million. To achieve this goal, these currencies reduce the number of coins issued for each block creation, which is equal to the amount of the seigniorage prize awarded to the block creator, geometrically. Specifically, these three currencies halve the seigniorage prize every 210,000 blocks (approximately four years).

[9]For example, f2pool's profit switching pool selects the most profitable currency from BTC, BCH, and BSV. Source: https://f2pool.io/mining/guides/how-to-mine-bitcoin/, accessed on November 7, 2021.

The schedule of halving is determined when BTC was launched. Therefore, halving is foreseeable but exogenous in the sense that its occurrence is independent of the latest market conditions.

The first and second halving arrived at BTC on November 28, 2012, and July 9, 2016, respectively. At that time, BCH and BSV were not yet hard forked from BTC. The latest halving is the third halving. The third halving occurred in the 2020 spring and reduced the seigniorage reward from 12.5 units (BTC, BCH, and BSV) to 6.25 units. Block 630,000 arrived at BCH on April 8, at BSV on April 10, and at BTC on May 11. The timing is slightly different across currencies because BTC, BCH, and BSV produce new blocks independently after the hard fork, while it is similar because all of the three aim at producing a block every ten minutes.

2.6 Difficulty Adjustment

As described in Subsection 2.2, the winning rate is a parameter value that is selected by the currency system (through the choice of the target). The objective of the parameter selection is to stabilize the frequency of block arrivals. BTC and its fork currencies aim at producing a new block every ten minutes, on average. Since the block arrival follows a nonhomogeneous Poisson process with intensity $w(t)h(t)$, the aim of currency k is to achieve

$$\frac{1}{w(t)h(t)} = 10 \text{ (minutes)} =: T^*. \tag{1}$$

Because the state variables that determine the hash rate $h(t)$ change over time, the currency must dynamically adjust its winning rate $w(t)$ to achieve this goal. This activity is called difficulty adjustment.

Using the data about the sequences of past winning rates and the *timestamps* (the time at which each block is produced), cryptocurrency systems algorithmically adjust their winning rates. Because the winning rate is updated only when the currency produces a block, slightly abusing the notation, we denote $w(l)$ by the winning rate for producing lth block of the currency, and $t(l)$ by the timestamp of the lth block to describe DAAs.

Various DAAs have been implemented so far. We describe the basic structures of these algorithms. For implementation reasons, the algorithms actually implemented are slightly different from the mathematical equations we introduce here. For full details, read the original code of these algorithms.

Original DAA The *original DAA* has been used by BTC since its launch. It adjusts the winning rate every 2,016 blocks, which corresponds to every two weeks if every block

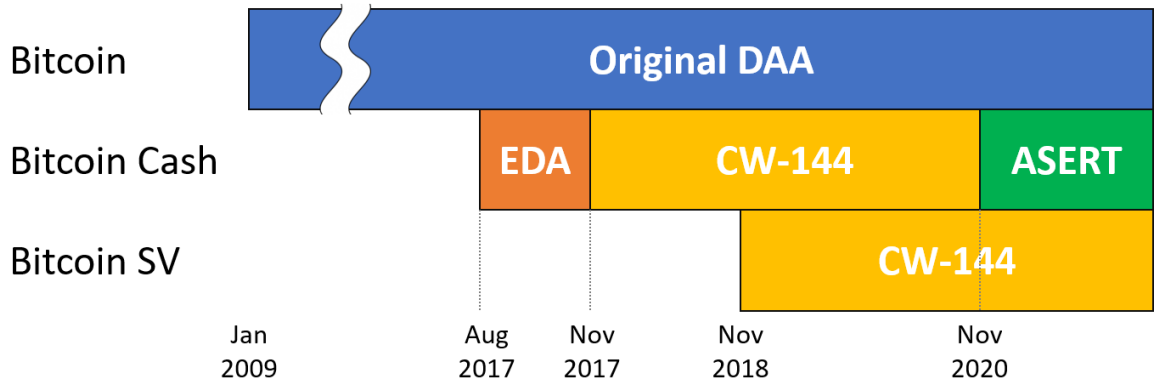


Figure 1: The history of DAAs adopted by cryptocurrency systems.

is produced in exactly ten minutes. The adjustment rule is:

$$w(l+1) = \begin{cases} \dfrac{t(l) - t(l-2016)}{2016 \times T^*} \cdot w(l) & \text{if } l \equiv 0 \mod 2016; \\ w(l) & \text{otherwise.} \end{cases} \tag{2}$$

The hash supply elasticity is the key to the stability of a cryptocurrency. Noda et al. (2020) show that the original DAA fails to stabilize the winning rate if the reward elasticity of the hash supply is smaller than 1.

EDA (Emergency Difficulty Adjustment Algorithm) The *emergency difficulty adjustment algorithm* (EDA) was implemented into BCH when it was hard-forked from BTC in August 2017. At that time, no one could foresee the appropriate level of the winning rate after the hard fork. The BCH community was concerned about the possibility that the initial level of the winning rate was too low for BCH to survive until the reward rate was adjusted to an appropriate level. To enhance the flexibility of the difficulty adjustment, EDA adopts an additional rule besides the difficulty adjustments performed by the original DAA. EDA shoots up the winning rate for block l by 20 percent if the time difference between the $(l-6)$th block and the $(l-12)$th block was more than 12 hours.

EDA was vulnerable to miners' strategic behavior. Some miners were said to have strategically triggered the emergency adjustment by suspending their activities. By doing so, miners can intentionally generate easy blocks, which provide a high reward rate to them. Indeed, BCH suffered from the fluctuation of block times and decided to introduce a new DAA to resolve this problem.[10] Consequently, EDA was abandoned in November 2017.

[10]See https://www.bitcoinabc.org/2017-11-01-DAA/, accessed on November 28, 2021

CW-144 *CW-144* is introduced as a successor of EDA by BCH in November 2017.[11] Unlike the original DAA and EDA, CW-144 updates the winning rate for every single block, using the 144-block moving average of block times and the inverse winning rate. The update rule of CW-144 is approximately given as follows:

$$w(l+1) = \frac{t(l) - t(l-144)}{T^* \times \sum_{l'=l-144}^{l} \frac{1}{w(l')}}. \tag{3}$$

CW-144 exhibited high performance in the pretesting, and the testing team unanimously recommended it as a new DAA. Noda et al. (2020) proved that CW-144 adjusts the block arrival rate to the targeted level asymptotically as long as the own reward elasticity of the hash supply is smaller than 144.

BSV is hard-forked from BCH in October 2018. While BSV made several updates after its launch, BSV still uses CW-144 as its DAA as of August 2021, while some supporters of BSV propose to move back to the original DAA in the future for an ideological reason. In November 2020, BCH decided to upgrade its DAA further, and therefore, CW-144 is no longer adopted by BCH.

ASERT (Absolutely Scheduled Exponentially Rising Targets) While CW-144 adjusts the winning rate more smoothly than the original DAA and EDA, its simple moving average design has been criticized for causing periodic winning-rate oscillations.

To address this issue, BCH decided to introduce a new DAA, named *ASERT (absolutely scheduled exponentially rising targets)*. ASERT determines the winning rate using the following formula:

$$w(l+1) = w(l) \cdot \exp\left(\frac{t(l) - t(l-1) - T^*}{\bar{T}}\right), \tag{4}$$

where

$$\bar{T} := 2,880 \text{ (minutes)} = 2 \text{ (days)} \tag{5}$$

is an algorithm parameter (called *half life*).

It compares the block height and elapsed time since the reference block to the target values for adjusting the winning rate. By doing so, it avoids the periodicity caused by the autoregressive adjustment rule of CW-144. According to the upgrade proposal,[12] among all the candidate algorithms, ASERT performed even better than CW-144 in various criteria.

[11] See https://reviews.bitcoinabc.org/D601, accessed on November 28, 2021.

[12] See https://read.cash/@jtoomim/bch-upgrade-proposal-use-asert-as-the-new-daa-1d875696, accessed on November 28, 2021.

3 Data

3.1 Source

There are four main sources of data: (i) the source code of cryptocurrency systems, (ii) the full history of blockchains obtained by setting up a full node for each cryptocurrency system, (iii) the cryptocurrency price (exchange rate against fiat money) obtained from Yahoo! Finance, and (iv) the advertised specifications of ASIC machines obtained from ASIC Miner Value.[13] By combining these data, we obtain a time-series data set of the full histories of BTC, BCH, and BSV over the 11-year period, between January 2009 and September 2020.

Source Code The ledgers of BTC, BCH, and BSV are managed in a decentralized manner, and literally any party can work as a recordkeeper (miner). Accordingly, the history of the consensus rules of cryptocurrency systems, which are written as computer programs, are disclosed publicly.[14] We obtained the full specifications of DAAs and the timings of updates from these source codes.

Blockchain Data The same as the source code, the full history of these blockchains are publicly disclosed, and any party can obtain it by setting up a full node (i.e., declaring to work as a miner). A blockchain is a collection of blocks, and each block contains data about (i) the timestamp (the reported time at which the block was produced), (ii) the target, or equivalently, the winning rate, and (iii) the amount of the prize paid to the block creator.

Cryptocurrency Prices Major cryptocurrencies, including the three currencies we focus on, have been actively traded for fiat money. These pieces of information can be downloaded from Yahoo! Finance's web page.[15]

ASIC Machine Specification The website, ASIC Miner Value, describes the advertised specification of ASIC machines including the target hash function, release date, hash power, and energy consumption.

3.2 Data Construction

We clean the blockchain data of BTC, BCH, and BSV to obtain the currency and block-height level data set. We convert the bits information in a block into the winning rate

[13] See https://www.asicminervalue.com/, accessed on December 16, 2021.

[14] BTC: https://github.com/bitcoin/bitcoin, accessed on November 7, 2021;
BCH: https://github.com/Bitcoin-ABC/bitcoin-abc, accessed on November 7, 2021;
BSV: https://github.com/bitcoin-sv/bitcoin-sv, accessed on November 7, 2021.

[15] Yahoo! Finance https://finance.yahoo.com/, accessed on November 7, 2021.

Table 1: Summary statistics: Currency and block-height level blockchain data since 2019

Currency	Variable	N	Mean	Sd	Min	Max
BTC	Winning rate	1.38e+05	1.93e-23	9.36e-24	9.3e-24	4.56e-23
	Block time (s)	1.38e+05	594	599	0.5	8.35e+03
	Hash rate (H/s)	1.38e+05	5.97e+20	3.55e+21	4.31e+18	1.87e+23
BCH	Winning rate	1.31e+05	7.77e-22	2.31e-22	2.85e-22	1.4e-21
	Block time (s)	1.31e+05	601	790	0.5	1.91e+04
	Hash rate (H/s)	1.31e+05	2.05e+19	1.03e+20	8.24e+16	5.66e+21
BSV	Winning rate	1.36e+05	1.7e-21	8.39e-22	3.14e-22	4.96e-21
	Block time (s)	1.36e+05	603	806	0.5	1.9e+04
	Hash rate (H/s)	1.36e+05	1.03e+19	5.24e+19	3.96e+16	3.6e+21

Table 2: Summary statistics: Currency and date level price data since 2019

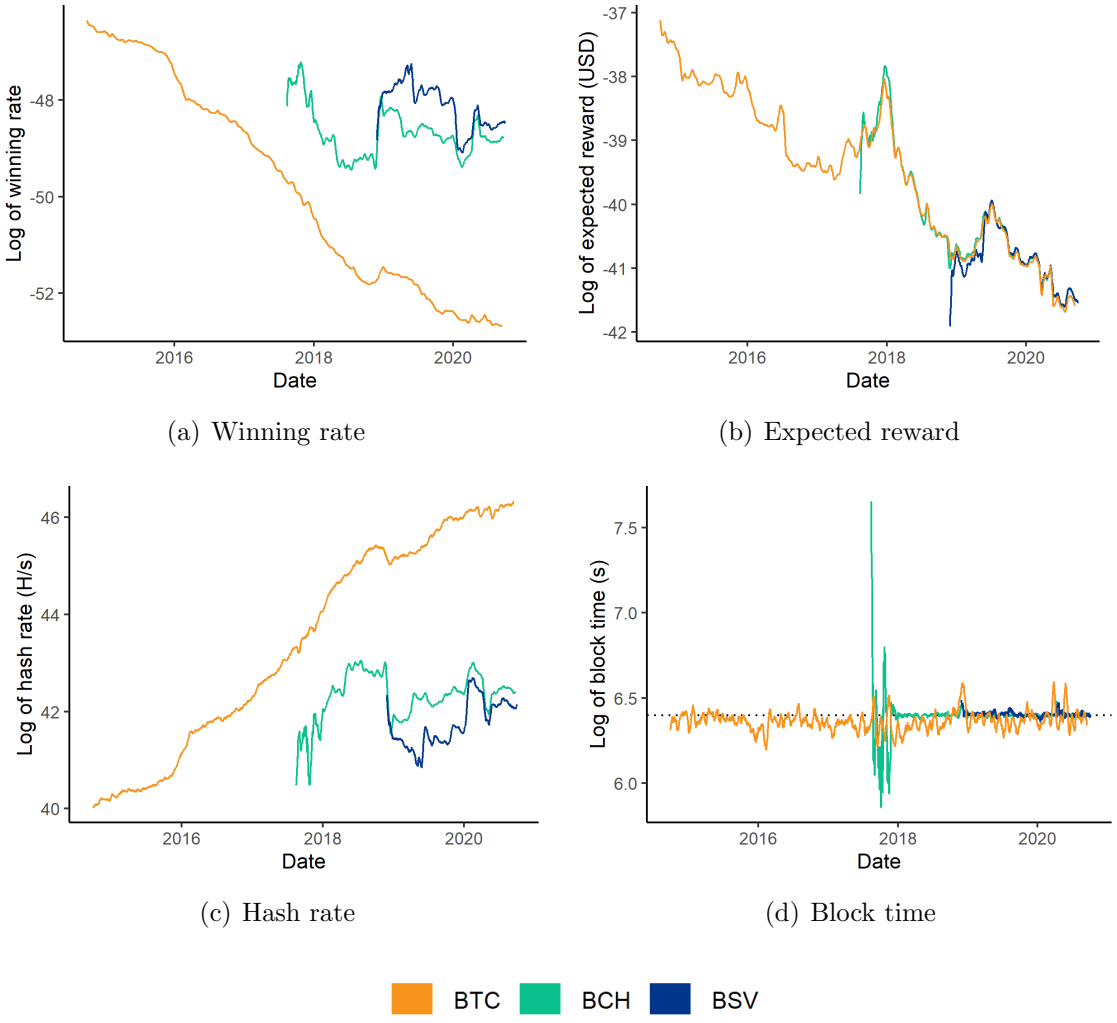
Currency	Variable	N	Mean	Sd	Min	Max
BTC	Price (USD)	878.0	15845.4	15512.4	3411.1	63208.9
BCH	Price (USD)	878.0	334.0	198.2	110.8	1436.8
BSV	Price (USD)	861.0	164.7	70.4	52.7	422.5

set for the block and calculate the time needed to generate the block as the difference between the block's and the preceding block's time stamps. We estimate the hash rate supplied to the block as the inverse of the product of the winning rate and the block time. This produces a currency block-height level data of winning rate, block time, and hash rate from the beginning of the history of BTC, January 3, 2009, to the data we collected blockchain information, August 5, 2021.

We then clean Yahoo! Finance data to obtain daily exchange rate data for each currency. We take the simple average of the daily lowest and highest price of a unit of currency in the U.S. dollar and refer to it as the price of the currency. The exchange rate data for BTC is available since September 17, 2014. The exchange rate data of BCH and BSV are available since their hard fork from BTC, August 2, 2014, and November 15, 2018, respectively.

3.3 Summary Statistics

Table 1 summarizes the currency and block-height level blockchain data since 2019, when the three currencies became available. During this period, there are 138 thousand BTC blocks, 131 thousand BCH blocks, and 136 thousand BSV blocks. Because they target the same 600 second block generation time, the numbers of generated blocks during a



(a) Winning rate

(b) Expected reward

(c) Hash rate

(d) Block time

Figure 2: 14-day moving averages of blockchain data

period are close.

The winning rate of BTC is on average 1.93×10^{-23}. The average winning rates of BCH and BSV are 7.77×10^{-22} and 1.70×10^{-21}. BTC is the hardest, followed by BCH and BSV. BTC's winning rate is the most stable with a standard deviation of 9.36×10^{-24}, although they use an inefficient DAA.

The average hash rate of BTC is 5.97×10^{20} and 2.05×10^{19} and 1.03×10^{19} for BCH and BSV. Thus, the hash supply to the BTC is approximately 10 times greater than to BCH and BSV.

The average block times are 594, 601, and 603 seconds for BTC, BCH, and BSV, hovering around the target 600 seconds. However, the variations are large. The standard deviations of the block times are 599, 790, 806 seconds, and the longest block time has the order of 10^3 for BTC while 10^4 for BCH and BSV. Thus, BTC is the most stable in the block time as well.

Table 2 summarizes the currency and date level price data since 2019. The average price of BTC since 2019 is \$15,845. At the peak, the price is \$63,209. The average prices of BCH and BSV are \$334 and \$165 with peaks of \$1,437 and \$423.

Figure 2 shows the 14-day moving averages of each currency's winning rate, expected reward, hash rate, and block time. The expected reward is the prize of mining multiplied by the winning rate and the price in the U.S. dollar, representing the expected payoff in the U.S. dollar of supplying a unit of hash for a miner.

Panel (a) shows that BTC's winning rate has constantly declined over time, reflecting the increase of the aggregate hash supply of cryptocurrencies. The winning rate of BCH and BSV does not have such a trend.

Panel (b) indicates that the expected reward is closely tied across BTC, BCH, and BSV. This comovement is not trivial because the currencies' DAAs do not explicitly target this. Miners' arbitrage across currencies is essential for this result. Because miners compare the currencies' expected rewards and the DAAs attempt to stabilize the hash supply, the winning rates are adjusted to reduce the difference in the expected reward. If miners do not arbitrage across currencies, we should not observe this pattern in data.

We have additional two implications from this figure. First, it indicates that the expected reward measured by the contemporaneous exchange rate approximates the incentive for miners, even though miners can hold the currency and sell in the future. Second, nevertheless, there are few periods during which the expected reward of BCH and BSV diverges from BTC's expected reward, suggesting a certain degree of adjustment failure.

Panel (c) shows that the hash supply to BTC has steadily grown despite the decrease in the winning rate and expected reward. The hash supply of BCH and BSV does not have such a trend. Panel (d) shows that the block time has been stable for BTC and BSV. BCH's block time was initially highly unstable, possibly due to the adoption of

EDA, but gradually became stable.

4 Effects of Halving

The third halving, which arrived for BCH on April 8, 2020, for BSV on April 10, and for BTC on May 11, offers the ideal setting to identify own and cross hash supply response to the expected rewards. First, the halving creates a large discontinuous change in the expected reward to miners by cutting the prize by half. Second, these events are predetermined by the design of the currencies and the timing is exogenously determined by the arrival time of each currency's 630,000th block. Miners' dynamic decisions including investment in ASIC machines can change in anticipation of the event, but their static decision, like the decision of which currency to mine, has little reason to react anticipating the arrival of halving. Therefore, no other discontinuous change is likely. Third, these events happened within 1 month for three currencies with some time intervals. These features enable us to identify the own and cross hash supply elasticity to the expected reward holding other things equal.

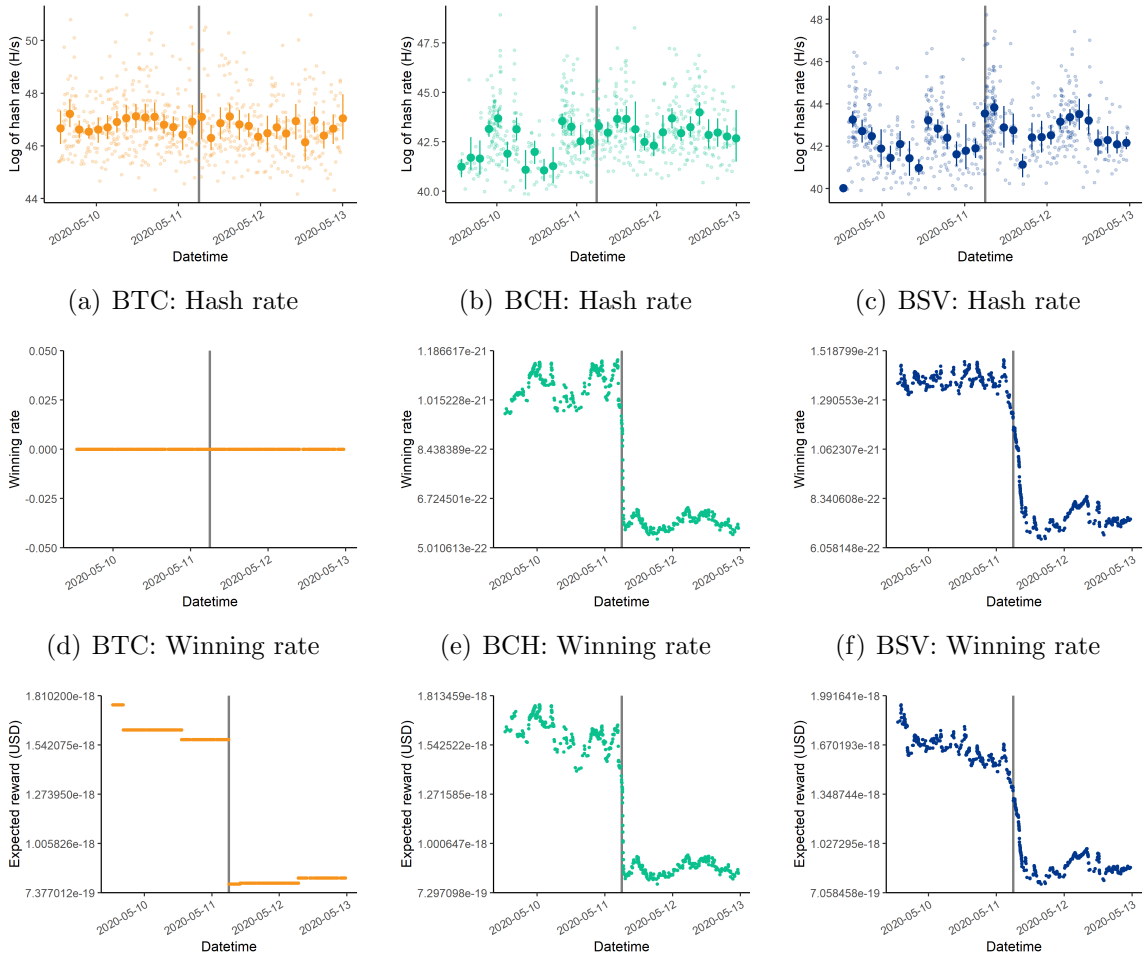
Before estimating the hash supply function using the variations around the third halving, we demonstrate how the hash supply reacts to the own and cross shocks and how the DAAs adjust the winning rate to the hash supply shocks. By doing so, we establish the fact that cryptocurrencies using the same hash function are closely connected to each other through the miner's hash supply market and that we must incorporate this fact for evaluating the stability and security of cryptocurrencies.

4.1 The Effects of BTC Halving

We examine how miners and BTC, BCH, and BSV's DAAs react to the halving of BTC. Panels (a), (b), and (c) of Figure 3 show how the hash rates of BTC, BCH, and BSV evolved for a window of 2-day before and after the halving. It plots the raw data and their binned scatter averages with the 95% confidence intervals. At this scale, the hash rate does not look sensitively reacting to the BTC halving, although the hash rate of BCH and BSV showed a slight increase for half a day after the halving.

However, this does not mean that miners are inelastic to the reward change. The opposite is true: This happened because the miners are elastic and the CW-144 of BCH and BSV successfully absorbed the external shocks coming from the BTC halving.

Panels (d), (e), and (f) of Figure 3 show how the winning rates are adjusted in the same time window. It only plots the raw data for visibility. We find that the winning rates of BCH and BSV quickly adjusted at the moment of BTC halving, with a slight delay in BSV. Strikingly, the winning rate of BTC is completely unchanged during this time window, due to the inflexibility of the original DAA. Because the original DAA does



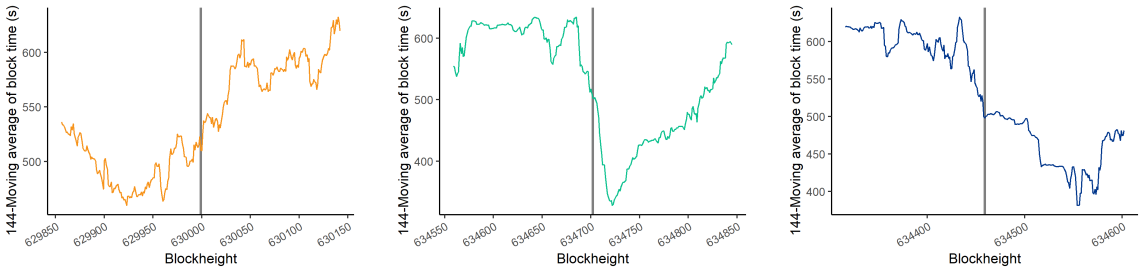
(a) BTC: Hash rate (b) BCH: Hash rate (c) BSV: Hash rate

(d) BTC: Winning rate (e) BCH: Winning rate (f) BSV: Winning rate

(g) BTC: Expected reward (h) BCH: Expected reward (i) BSV: Expected reward

Figure 3: Hash rate, winning rate, and expected reward around the BTC halving

Note: In hash rate figures, binned averages and their 95% confidence intervals are drawn on the raw data points. In winning rate and expected reward figures, only raw data points are drawn.



(a) BTC (b) BCH (c) BSV

Figure 4: 144-block moving averages of block time around the BTC halving

Table 3: The effects of BTC halving on hash rate and block time

	Log of hash rate (H/s)		
	BTC	BCH	BSV
Estimate	-0.119	0.375	0.909
Std.error	(0.132)	(0.13)	(0.129)
Nobs.left	4188	4011	3977
Nobs.right	3755	4020	4046
H.left	5.87e+05	7.18e+05	7.71e+05
H.right	9.28e+05	7.56e+05	6.26e+05
B.left	9.45e+05	1.26e+06	1.26e+06
B.right	1.45e+06	1.25e+06	1.04e+06
Order.regression	2	2	2
Kernel	Triangular	Triangular	Triangular
Bwselect	Msetwo	Msetwo	Msetwo

Note: Standard errors are in the parentheses. The estimates are the local average treatment effects at the BTC halving time identified by a regression discontinuity design. The bandwidth is chosen for both sides of the cutoff by using Calonico, Cattaneo, and Titiunik (2014). H.left and H.right are bandwidth for the estimation of the regression functions and B.left and B.right are for the estimation of the bias. The bias is corrected and the standard errors are robust to the asymptotic variance due to the bias term using Calonico et al. (2014). Local quadratic functions are used to fit the regression functions with a triangular kernel function.

not adjust the winning rate, the expected reward of BTC is cut by half due to halving. Panel (g) of Figure 3 confirms the drop of BTC's expected reward.

This causes a shift of hash supply from BTC to BCH and BSV, increasing BTC's block time and decreasing BCH and BSV's block time. Figure 4 focus on the short 144-block-height windows before and after halving. Note that this window is shorter than that of Figure 3: 144 blocks are produced in 1 day, if each block is produced in (targeted) 10 minutes. To make it comparable to the CW-144's input, we take the 144-block-height moving average of the block time. It is clear from the figure that the moving average of BTC's block time increases after the halving and BCH's and BSV's decrease.

Because the block time increased, CW-144 of BCH and BSV immediately cut the winning rate until the mining incentives become indifferent across currencies. This creates the sharp drop of BCH and BSV's winning rates in Panels (e) and (f) and the expected reward in Panels (h) and (i) as a consequence.

The block time dropped first at BCH and then at BSV. This suggests that miners for BTC first shifted to BCH and then to BSV because the BCH's CW-144 adjusted the winning rate. This is the reason for the delay in the response of BSV's winning rate.

We confirm these observations by a sharp regression discontinuity design applied to the BTC halving timing. We use the date time at the second level as a running variable.

We use a local quadratic polynomial with a triangular kernel function for estimating the outcome regression functions. We choose the optimal bandwidths minimizing the mean squared error separately for both sides of the cutoff for estimating the regression functions and bias terms, correct the bias in the estimate, and use the robust standard errors according to Calonico et al.'s (2014) method. The analysis is implemented by their *rdrobust* package in R (Calonico, Cattaneo, Farrell, and Titiunik, 2017).

Table 3 summarizes the estimation results. First, the halving does not cause any statistically significant impact on BTC's hash rate and block time. On the contrary, at the moment of halving, the hash rate of BCH and BSV statistically significantly increased by 0.38 and 0.91%, respectively. If the winning rate was not adjusted, the block time of BCH and BSV should have missed the target for a longer period.

This analysis reveals multiple important facts regarding a miner's hash supply and the function of a DAA. First, miners are elastic to the changes in the expected reward. Second, BTC, BCH, and BSV, which use the same SHA-256 hash function, are substitutes from a miner's perspective. In other words, these cryptocurrencies are connected through the hiring market of miners and shocks in a currency propagate to other currencies. Third, the cross-hash supply elasticity of BCH and BSV to BTC's reward is substantially higher than the own hash supply elasticity of BTC, amplifying the original shock. Fourth, nevertheless, the efficient CW-144s of BCH and BSV successfully manage the instability created by a large reward shock to BTC. Fifth, BTC could survive this shock because of the low own hash supply elasticity despite the inefficient original DAA. Finally, the choice of DAA by a currency has externalities to other currencies. The adoption of efficient DAA by BCH and BSV should have benefited BTC: if BCH and BSV used inefficient DAA, BTC should have lost more hash supply for a longer period. On the other hand, the adoption of inefficient DAA by BTC should destabilize the hash supply to BCH and BSV.

Therefore, considering the connection through the miner's market is essential for creating a functioning transaction system by PoW, because shocks in a currency propagate to other currencies through miners' behaviors and are even amplified depending on the DAAs used by other currencies. Specifically, the evaluation of the efficiency, stability, and security of a PoW cryptocurrency must be performed under the knowledge of miner behavior and the market structure of competing cryptocurrencies. To this end, we lay out a model of miners' hash supply behavior and evaluate the performance of DAAs by counterfactual simulations based on the estimated model.

4.2 The Effects of BCH and BSV Halving

We perform the same analysis around the BCH halving on April 8, 2020, and BSV halving on April 10. The figures and tables are in the Appendix. Figures A1 and A3

are the counterparts of Figure 3 for BCH and BSV halving. Figures A2 and A4 are the counterparts of Figure 4. Tables A1 and A2 are the counterparts of Table 3. Because these halvings happened in a short period, we write the timing of the main halving by the solid vertical line and the other by the dashed vertical line.

We only briefly discuss the patterns around the two halvings. Table A1 shows that the BCH's hash rate dropped by 0.7% by BCH halving and BSV's hash rate increased by 0.4% both statistically significantly, but BTC's hash rate was unchanged. Table A2 shows that BSV's hash rate dropped by 1.5% and BCH's hash rate increased by 1.89% both statistically significantly, whereas BTC's hash rate was unchanged. It confirms that BTC's hash rate is inelastic to both internal and external reward shocks, whereas BCH's and BSV's hash rates are sensitive to both shocks.

5 Model

5.1 Environment

We consider a continuous-time environment in which there are $K \in \mathbb{Z}_{++}$ cryptocurrencies that use the same hash function. Time is indexed by $t \in \mathbb{R}_+$. Each cryptocurrency is labeled by $k \in \{1, \ldots, K\}$. The unit of account of cryptocurrency k is TKN(k). For simplicity, we unify all fiat money and denote its unit of account by USD.

We analyze the behavior of the *hash rate*, the total number of hash computations exerted within a unit time by miners. We denote the hash rate of currency k in time t by $h(k, t; \theta)$ (H/s). θ is a parameter that determines miners' behavior.

Computing a hash value for mining is equivalent to drawing a lottery. We denote the probability that a miner successfully creates a new block of currency k with one hash computation in time t by $w(k, t)$ (block/H). We call $w(k, t)$ the *winning rate* of currency k in time t.

When a miner creates a new block of currency k in time t, he obtains a *prize* of $m(k, t)$ (TKN(k)/block). For simplicity, we ignore transaction fees and assume that the prize is solely provided as seigniorage. This assumption has been valid historically: In these three cryptocurrencies, the seigniorage reward has been a dominant component of the prize.[16] The price of currency k in time t is denoted by $e(k, t)$ (USD/TKN(k)).

Currency k's block height in time t is denoted by $l(k, t) \in \mathbb{Z}_{++}$, and when a block arrives in time t, $l(k, t)$ is incremented by one. Conversely, we denote the time in which currency k's block l arrives by $t(k, l)$. Formally, $t(k, l)$ is defined by $t(k, l) = \min_{t'}\{l(k, t') = l\}$.

[16] For example, in BTC, transaction fees per block have been less than 10% of the seigniorage reward for most of the period. After the halving, while the level of transaction fees increased slightly, the total transaction fees per block were at most 17.2% of the seigniorage reward after the third halving. See https://btc.com/stats/fee, accessed on November 28, 2021.

We define $r(k,t) \coloneqq w(k,t)m(k,t)e(k,t)$ (USD/H) as the *expected reward rate* from investing one hash into currency k in time t, denominated in fiat money. The expected reward rate represents the expected revenue from supplying a hash.

Prize $m(k,t)$ is predetermined by the halving policy, and price $e(k,t)$ is exogenously determined. The DAA determines the winning rate $w(k,t)$ on the basis of the timestamps $t(k,l)$ and winning rates $w(k,t(k,l))$ written in blocks that have arrived up to then. (The functional forms of DAAs are stated in Subsection 2.6, and therefore, we do not duplicate them here.)

Miners choose which currency to contribute or not to work by comparing the expected rewards, which results in the aggregate hash rate $h(k,t;\theta)$. We assume that each miner is infinitesimal, and therefore, the aggregate variables, w and e are exogenous to the individual miner. In this situation, a miner's profit-maximizing strategy is to mine the currency k in time t if and only if (i) $k \in \arg\max_{k'} r(k',t)$ and (ii) $r(k,t)$ is larger than the miner's variable cost. Here, for each miner's time-t decision making, only the profile of current expected rewards, $(r(k,t))_{k=1}^{K}$, is relevant. Building upon this observation, we assume that, for each k, currency k's time-t aggregate hash supply function $h(k,t;\theta)$ is a function of the profile of time-t expected rewards, $(r(k',t))_{k'=1}^{K}$. We also note that Figure 2.b supports this assumption in the sense that the level of expected rewards of SHA-256 cryptocurrencies is largely aligned due to arbitrage.

When the winning rate is small and the hash rate is large, the number of blocks generated in a time interval approximately follows a nonhomogeneous Poisson process whose arrival rate in time t is $w(k,t)h(k,t)$; i.e.,

$$l(k,\bar{t}) - l(k,\underline{t}) \sim Poisson\left(\int_{\underline{t}}^{\bar{t}} w(k,s)h(k,s)ds\right), \tag{6}$$

or equivalently,

$$t(k,l+1) - t(k,l) \sim Exp\left(\int_{\underline{t}}^{\bar{t}} w(k,s)h(k,s)ds\right). \tag{7}$$

In reality, the assumption is satisfied for the SHA-256 market: As described in Table 2, BTC's winning rate is in the order of 10^{-23}, and BTC's hash rate is in the order of 10^{20}. The block arrivals are independent across currencies.

5.2 Epoch

While we have developed a continuous-time model, most variables of interest are discrete in nature. Currency k's winning rate $w(k,\cdot)$ and prize $m(k,\cdot)$ are associated with the blocks of currency k; thus, $w(k,\cdot)$ and $m(k,\cdot)$ are changed only when currency k's block

arrives. While currency k's price $e(k,\cdot)$ and miners' cost factors (such as the electricity price they face) can change any time, as long as blocks arrive frequently, it is also innocuous to assume that these variables change only when a block arrives. In the SHA-256 market, the DAAs aim to produce blocks every 10 minutes. This is sufficiently shorter than one day (1,440 minutes), the update frequency of the cryptocurrency price data. Since the hash supply $h(k,t,\theta)$ is a function of the expected reward profile at that time point, $(r(k',t))_{k'=1}^{K}$, $h(k,\cdot;\theta)$ is also updated only when a block arrives.

To run the estimation efficiently, we introduce a method to restructure the continuous-time model into a discrete-epoch model. An *epoch* is an interval that shares identical state profiles. An epoch ends when a block arrives in some currency k, because upon the arrival of a new block, the variables, such as the winning rate, could be updated. Epochs are indexed by $n = 0, 1, \ldots$, and epoch n ends and epoch $n+1$ starts when a block of currency k arrives for some $k \in \{1, \ldots, K\}$. Because variables only change at the epoch level, an epoch is sufficient to index time. Accordingly, the time index of $m(k,t), e(k,t), w(k,t)$, and $h(k,t;\theta)$ is replaced with epoch index n to obtain $m(k,n), e(k,n), w(k,n)$, and $h(k,n;\theta)$.

We let $t(n)$ be the physical time at which epoch n starts, and $T(n)$ be the physical time length of epoch n, i.e., $T(n) \coloneqq t(n+1) - t(n)$. In epoch n, the arrival of currency k's block follows a *homogeneous* Poisson process with intensity $w(k,n)h(k,n;\theta)$—within an epoch, the winning rate and the hash rate maintain the same value, and therefore, the block arrival rate is homogeneous. Since there are K currencies, K independent Poisson processes are running, where the intensity of k-th process is given by $w(k,n)h(k,n;\theta)$. Hence, the *epoch* arrival, which adds up all the block arrivals across Kf currencies, follows the merged Poisson process with intensity $\sum_{k=1}^{K} w(k,n)h(k,n;\theta)$. Epoch n ends when the first tick of the merged Poisson process, and its distribution is given by an exponential distribution with the same intensity. Accordingly, we have

$$T(n) \coloneqq t(n+1) - t(n) \sim Exp\left(\sum_{k=1}^{K} w(k,n)h(k,n;\theta)\right). \tag{8}$$

Let $a(k,n) \in \{0,1\}$ be a binary variable that specifies whether epoch n ends by the arrival of currency k; i.e.,

$$a(k,n) = \begin{cases} 1 & \text{if } l(k,n+1) = l(k,n) + 1; \\ 0 & \text{if } l(k,n+1) = l(k,n), \end{cases} \tag{9}$$

where $l(k,n)$ be the block height of currency k in epoch n. Then, by the property of the

merged Poisson process, we have

$$\Pr(a(k,n)=1|T(n)) = \frac{w(k,n)h(k,n;\theta)}{\sum_{k'=1}^{K} w(k',n)h(k',n;\theta)}. \tag{10}$$

This conditional probability is independent of $T(n)$: Regardless of the epoch length, the probability that the epoch ends by currency k's block arrival is proportional to currency k's block arrival rate $w(k,n)h(k,n;\theta)$.

5.3 Likelihood of Block Arrival Events

We observe the length of each epoch (i.e., $T(n)$ for all n), and the currency whose block arrival terminated each epoch (i.e., the values of $a(k,n)$ for all k and n). Because the underlying hash supply determines the likelihood of these events, we can infer the parameters of the hash supply function by a maximum likelihood estimation.

Consider an event like "epoch n ends by currency k's block arrival (i.e., $a(k,n)=1$) and epoch n lasts for $T(n)$ seconds". This event happens if and only if (i) the Poisson clock of currency k, whose intensity is $w(n)h(k,n;\theta)$, ticks when $T(n)$ seconds passed, and (ii) no other Poisson clocks have ticked by then.

The probability density that event (i) occurs is

$$w(k,n)h(k,n;\theta)\exp\left(-w(k,n)h(k,n;\theta)T(n)\right). \tag{11}$$

The probability that each currency $k' \neq k$ does not have a block arrival until $T(n)$ seconds passed is given by $\exp\left(w(k,n)h(k,n;\theta)\right)$. Since block arrivals are independent across currencies, the probability that event (ii) occurs is

$$\exp\left(-\sum_{k'\neq k} w(k',n)h(k',n;\theta)T(n)\right). \tag{12}$$

Therefore, the probability density of the whole event is

$$w(k,n)h(k,n;\theta)\exp\left(-\sum_{k'=1}^{K} w(k',n)h(k',n;\theta)T(n)\right). \tag{13}$$

Hence, the likelihood of $\{a(\cdot,n),T(n)\}_{n=1}^{N}$ as a function of hash supply parameter θ conditional on the history $\{m(\cdot,n),e(\cdot,n),w(\cdot,n)\}_{n=1}^{N}$ is:

$$L(\theta;\{a(\cdot,n),T(n)\}_{n=1}^{N}) \tag{14}$$

$$= \prod_{n=1}^{N}\left[\sum_{k=1}^{K} a(k,n)w(k,n)h(k,n;\theta)\right]\exp\left(-\sum_{k'=1}^{K} w(k',n)h(k',n;\theta)T(n)\right). \tag{15}$$

We approximate the hash supply function by a log-log linear function:

$$h(k,n;\theta,\bar{h}) = \bar{h}(n) \cdot \exp\left(\alpha_k + \sum_{k'=1}^{K} \beta_{k',k} \log r(k',n)\right), \tag{16}$$

where $\theta \coloneqq (\alpha, \beta)$ is the set of parameters, and $\bar{h}(t)$ is the time trend, which captures the growth of the total capacity of mining ASIC. Here, $\beta_{a,b}$ represents currency b's hash-supply elasticity of currency a's expected reward rate.

The maximum likelihood estimator $\hat{\theta}$ is a maximizer of likelihood function (15). The estimation of the hash supply is analogous to labor supply estimation. Because of this analogy, one may be concerned with the endogeneity of "wages", that is, winning rates. This is not a problem in the current context, because there is no unobserved heterogeneity in the DAAs that determines the winning rate. The winning rate is perfectly determined by the observed history of block time and there is no unknown parameter in the DAAs. We replicated each currency's DAAs in R and confirmed this perfect match. Because the distribution of the winning rate conditional on the observed history is independent of the hash supply parameter θ, maximizing the full likelihood is equivalent to maximizing (15).

We focus on the period from 28 days before the BCH halving to 28 days after the BTC halving, which is about three months long. As demonstrated in Section 4, halving brings a large exogenous shock to the expected reward rate within a short period. The observation in this period enables us to examine miners' short-term operation decisions in response to the expected reward while ignoring miners' long-term investment decisions. We assume that our period of observation is short enough that no miner could increase their capacity significantly during this period. This assumption enables us to regard $\bar{h}$ as a constant function, and we normalize it to $\bar{h} = 1$.

5.4 Estimation of the Exchange Rate Process

We assume that the exchange rate process is independent of block arrival, and the daily log increment of the exchange rate follows an i.i.d. distribution. Specifically, we assume that the daily exchange rate is determined by the following Lévy process.

$$\log e(d+1) - \log e(d) = \mu + \sigma\epsilon(d), \tag{17}$$

where d is the day, μ is the daily drift rate, σ is the daily volatility, and $\epsilon(t)$ is the i.i.d. shock. Because the empirical distribution of the log increment in the data exhibits a fatter tail than a normal distribution, we fit a t-distribution to $\epsilon(d)$.

We first calculate the sample mean and standard deviation of the log increment of the exchange rate to estimate the drift rate μ and the volatility σ. Using the estimated

Table 4: Estimation results of hash supply function

	Log of hash rate		
Num epoch = 38046	BTC	BCH	BSV
Intercept	52.9	49.9	47.8
	(1.97)	(1.99)	(1.97)
Log of expected reward of BTC	0.626	-3.98	-3.19
	(0.103)	(0.113)	(0.106)
Log of expected reward of BCH	-0.24	5.39	-1.54
	(0.0953)	(0.127)	(0.0929)
Log of expected reward of BSV	-0.223	-1.22	4.87
	(0.0977)	(0.0765)	(0.118)

Note: Standard errors are in the parentheses. The hash rate is in H/s and the expected reward is in the U.S. dollar.

drift rate $\hat{\mu}$ and volatility $\hat{\sigma}$, we compute the standardized log increments,

$$\frac{\log e(d+1) - (1+\hat{\mu}) \log e(d)}{\hat{\sigma}}, \tag{18}$$

and fit them to t-distribution to obtain the degree of freedom by the maximum likelihood estimation.

6 Estimation Results

6.1 Hash Supply Function

Table 4 summarizes the maximum likelihood estimation results of the hash supply function. As we expected, for all currencies, the on-diagonal elements, which denote the own reward elasticity of the hash supply, are positive. This is because when a currency provides miners with a better expected reward, then more miners switch to contribute to the currency. In contrast, the off-diagonal elements, which denote the cross-reward elasticity of the hash supply, are negative. This is because the expected reward of rival currencies is the value of the outside option. These results articulate the importance of the miners' short-term operation decisions—real-world miners are actively searching for which currency to contribute, taking account of the profile of expected rewards.

BTC's own reward elasticity of hash supply is 0.626, indicating that BTC's hash supply is highly inelastic. The cross elasticities, i.e., BCH's and BSV's reward elasticities of BTC's hash supply are -0.24 and -0.223, respectively. Therefore, BCH's and BSV's reward changes hardly affect BTC's hash supply. As shown in Panels (a) and (g) of Figure 3, even though BTC's halving cut the reward by half and the original DAA failed

Table 5: Estimation results of exchange rate process

Curency	Mean	Sd	Df
BTC	0.0017	0.0318	8.4798
BCH	0.0001	0.0654	6.9181
BSV	0.0009	0.0681	6.2921

Note: The mean and standard deviation of the daily log increment of the exchange rate to the U.S. dollar are calculated. The standardized log increments are fitted to a t-distribution and the degree of freedom is estimated by the maximum likelihood method.

to adjust the winning rate in the short run, BTC's hash rate was stable. Thus, BTC has a large number of loyal miners who continue supplying their hash power even when other currencies offer higher expected rewards.

The loyal miners' choice can be rational. BTC is the oldest and largest cryptocurrency. BTC's market capitalization is approximately 100 times larger than BCH and BSV. The thick USD-BTC exchange market makes it easy to convert the mining prize into fiat money. This may be creating a premium for miners.

By contrast, the hash supplies of BCH and BSV are highly elastic. BCH's and BSV's own reward elasticity of hash supply are 5.39 and 4.87, respectively. These values are more than eight times greater than BTC's own reward elasticity. The hash supplies of BCH and BSV are highly sensitive to other currencies' rewards. BTC's and BSV's reward elasticity of BCH's hash supply are -3.98 and -1.22, respectively. BTC's and BCH's reward elasticity of BSV's hash supply are -3.19 and -1.54, respectively. These estimates imply that a non-negligible share of miners contributing to BCH and BSV are switching miners, who actively search for the most profitable currency to mine.

The elasticity of hash supply is crucial for the stability of the block arrival rate. Using a single-currency model, Noda et al. (2020) showed that the original DAA stabilizes the block arrival rate in the long run if and only if the (own) elasticity is smaller than one. While BTC's hash supply satisfies this condition, BCH's and BSV's do not. By contrast, CW-144 is stable in a broader class of environments—CW-144 stabilizes the block arrival rate in the long run if and only if the (own) elasticity is smaller than 144, where 144 originates from the window for taking the moving average. For all SHA-256 currencies, the estimated elasticity is smaller than 144. This finding implies that the adoption of CW-144 is essential for the survival of BCH and BSV. We expect that if BCH and BSV had kept using the original DAA, BCH and BSV would have failed to survive the third halving shock.

6.2 Exchange Rate Process

Table 5 shows the estimation result of the exchange rate process. The standard deviation of log increment is 0.03 for BTC and 0.07 for BCH and BSV. The degree of freedom of the error terms is 8.5 for BTC and 6.9 and 6.3 for BCH and BSV, respectively. It shows that the exchange rates of BCH and BSV are more volatile and have a fatter tail than the BTC's exchange rate.

Our estimation results imply that BCH and BSV face more difficult challenges than BTC not only in terms of the elasticity of hash supply but also in terms of volatile exchange rates. While the halving shock is a rare event (it occurs only once four years), changes in the exchange rate regularly occur. These exchange-rate shocks change the expected reward denominated in fiat money, influencing miners' incentives for supplying the hash rate. Due to larger standard deviations and fatter tails, BCH and BSV have experienced bigger shocks more frequently than BTC. Their DAAs need to handle these regular shocks, and therefore, a higher ability to adjust the winning rate is demanded. This would be another reason why BCH is actively searching for a better DAA whereas BTC has retained its original DAA.

7 Counterfactual DAAs

7.1 Simulation Setting

In this section, we provide counterfactual analyses that demonstrate the performance of the three DAAs, the original DAA, CW-144, and ASERT. We focus on the economy's response to the largest shock—the third BTC halving. We assume that the three SHA-256 currencies, BTC, BCH, and BSV, experienced the actual history until block 629,999, which is the block produced right before the third halving. Starting from block 630,000, these three currencies install counterfactual DAAs (if any) and the evolution of their blockchains and exchange rates is simulated using our model. To produce counterfactual histories, we use the model described in Section 5 and the estimated parameters shown in Section 6.

Since the exchange rate is independent of block arrival, we generate the evolution of the exchange rate in advance, separately. We draw $\epsilon(d)$ from a t-distribution to generate a counterfactual history of the daily exchange rate data. Then, we perform linear interpolation to compute the exchange rate at each physical time, t. When the epoch n starts at time $t(n)$, then the exchange rate at time $t(n)$ will be regarded as the exchange rate associated with that epoch.

We describe how we generate an epoch n. The state variables, the physical time at which the epoch n starts $t(n)$, the winning rate $w(\cdot, n)$, prize $m(\cdot, n)$, and exchange rate $e(\cdot, n)$, are passed from the previous epoch. Substituting these variables into the

Table 6: Summary of Simulation Scenarios in Subsection 7.2.

	BTC	BCH	BSV
Scenario 1 (actual)	Original DAA	CW-144	CW-144
Scenario 2	CW-144	CW-144	CW-144
Scenario 3	Original DAA	Original DAA	CW-144
Scenario 4	Original DAA	Original DAA	Original DAA

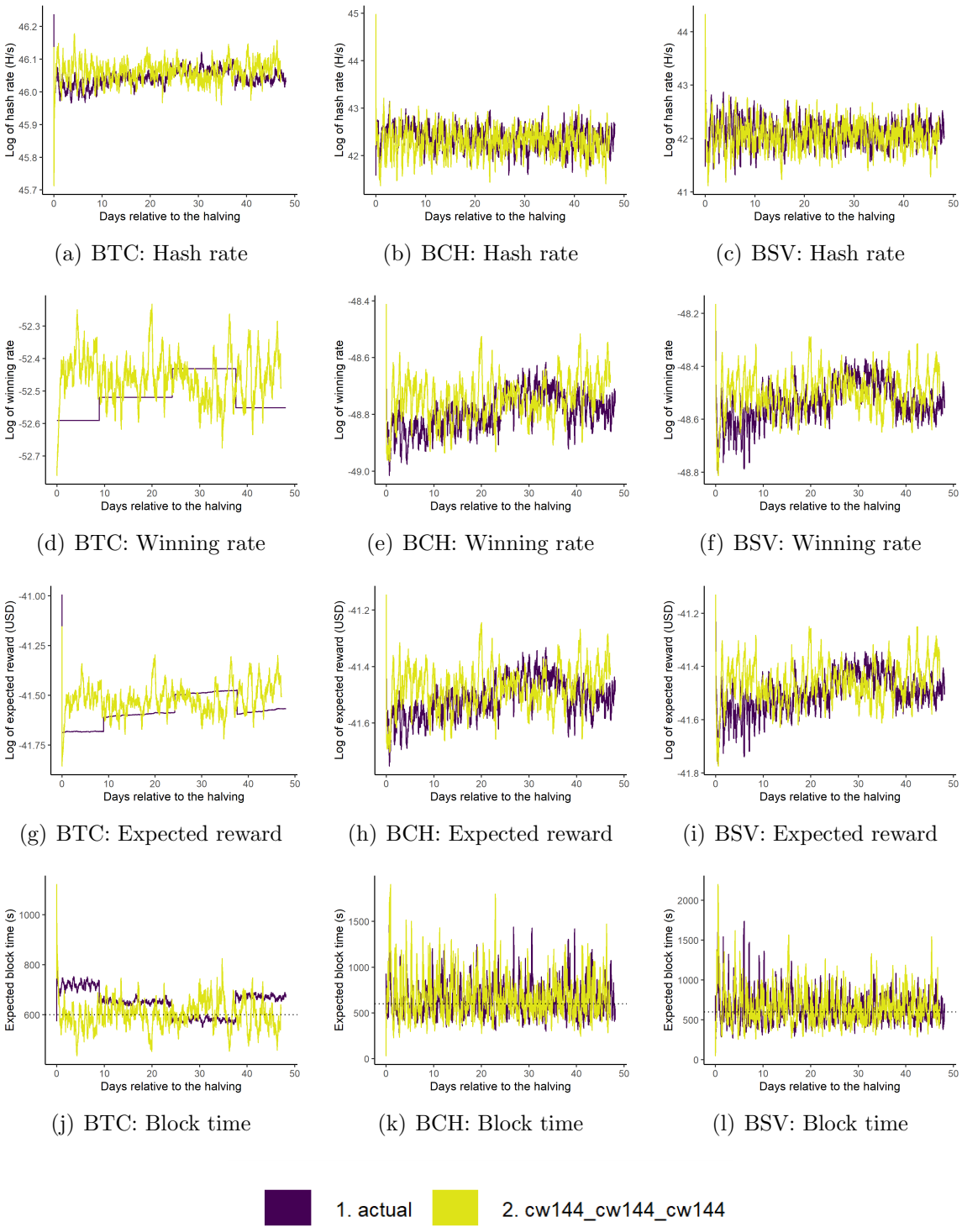
estimated hash supply function (16), we compute the hash rate in the epoch n, $h(\cdot, n; \theta)$. We first draw the epoch length $T(n)$ using (8). Afterward, we determine the currency for which a block arrives using (10). Using the DAA specified in Subsection 2.6, we update and obtain the winning rate $w(k, n+1)$ of the currency that produces a new block. Similarly, we refer the pre-determined rule to update the prize $m(k, n+1)$. For all currencies $k' \neq k$, the winning rate and prize are retained the same, $w(k', n+1) = w(k', n)$ and $m(k', n+1) = m(k', n)$, because these values could be updated only when a block arrives to the currency. The physical time at which the new epoch $n+1$, starts, is given by $t(n+1) = t(n) + T(n)$, and we refer the exchange rate at that time, which is drawn beforehand, to determine $e(k, n+1)$. Now, all state variables are updated, and we proceed to the epoch $n+1$.

Using this procedure, we generate a counterfactual history of the SHA-256 economy for 60 days. We display the realized path of a single simulation because the patterns look qualitatively similar across realizations. Our aim is to elucidate the relationship between the choice of DAAs and the stability of the economy.

7.2 DAA Choice and Stability

We first examine what would have happened if BTC upgrades the DAA to CW-144 and if BCH and BSV downgrade the DAA to the original DAA. Specifically, we consider the following four scenarios: (i) the actual DAA profile, that is, BTC uses the original DAA, and BCH and BSV use CW-144, (ii) BTC upgrades, that is, all currencies use CW-144, (iii) BCH downgrades, that is, BTC and BCH use the original DAA, and BSV uses CW-144, and (iv) BSV also downgrades, that is, all currencies use the original DAA. Table 6 summarizes the DAA profile of these scenarios.

Figure 5 depicts the simulation results of Scenarios 1 and 2. The behavior of variables under the actual DAA profile (Scenario 1) is similar to the real economy. While BTC's winning rate adjustment is slow (Panel d, Scenario 1), BTC's hash rate remains stable (Panel a, Scenario 1) since BTC's hash supply is inelastic. By contrast, while BCH's and BSV's hash supplies are elastic since CW-144 quickly adjusts their winning rates (Panels b and c, Scenario 1), BCH's and BSV's block arrival rates also quickly return to the desired levels (Panels k and l, Scenario 1). This is how these three currencies survive the



(a) BTC: Hash rate (b) BCH: Hash rate (c) BSV: Hash rate

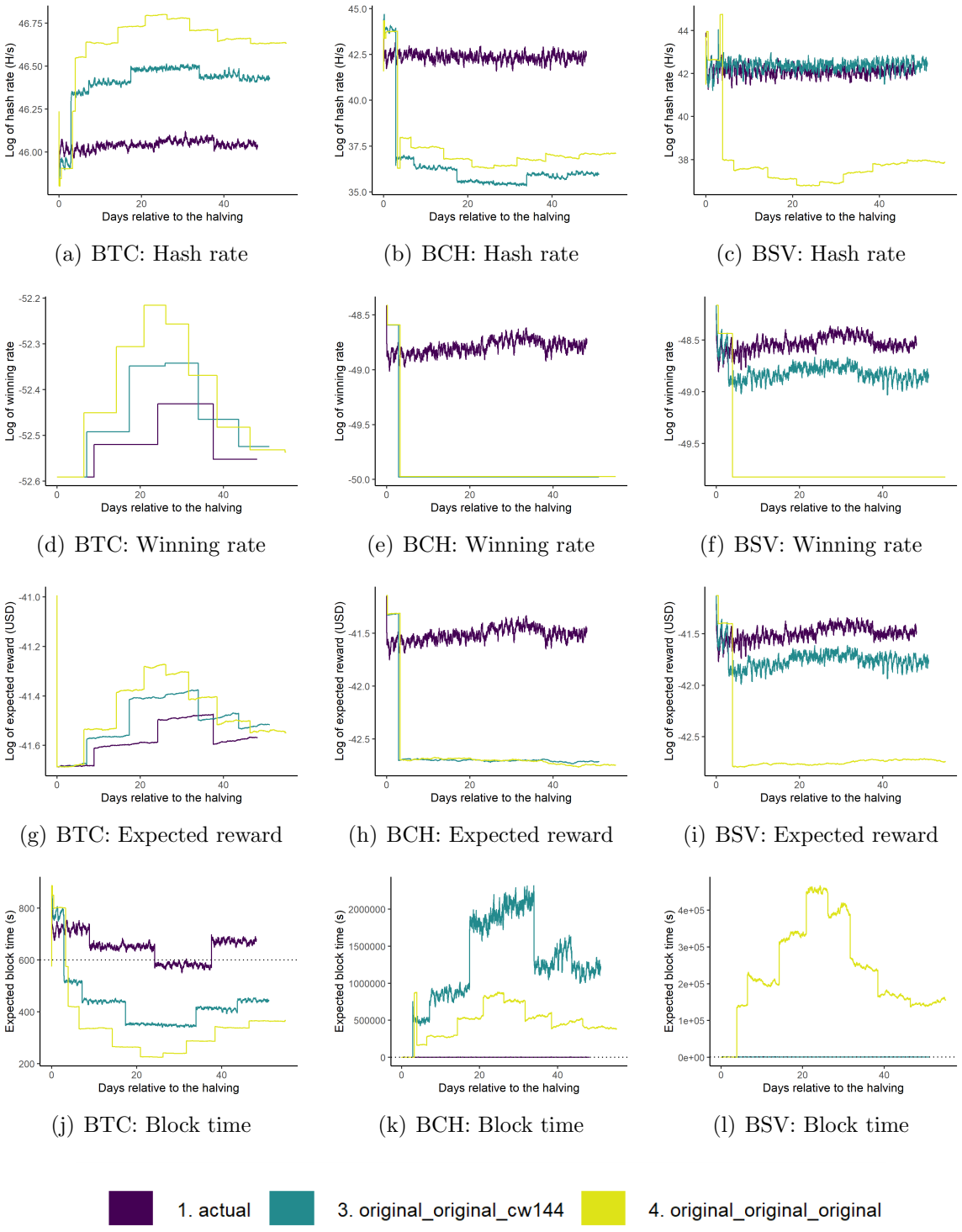
(d) BTC: Winning rate (e) BCH: Winning rate (f) BSV: Winning rate

(g) BTC: Expected reward (h) BCH: Expected reward (i) BSV: Expected reward

(j) BTC: Block time (k) BCH: Block time (l) BSV: Block time

Figure 5: Counterfactual Simulations: Upgrading to CW-144

Note: The figure shows the counterfactual simulation results starting from the 1 block before the BTC halving. A line shows a single path of the simulation under the specified scenario. The scenario name represents the DAA of BTC, BCH, and BSV, respectively.



(a) BTC: Hash rate (b) BCH: Hash rate (c) BSV: Hash rate

(d) BTC: Winning rate (e) BCH: Winning rate (f) BSV: Winning rate

(g) BTC: Expected reward (h) BCH: Expected reward (i) BSV: Expected reward

(j) BTC: Block time (k) BCH: Block time (l) BSV: Block time

Figure 6: Counterfactual Simulations: Downgrading to the Original DAA

Note: The figure shows the counterfactual simulation results starting from the 1 block before the BTC halving. A line shows a single path of the simulation under the specified scenario. The scenario name represents the DAA of BTC, BCH, and BSV, respectively.

third halving shock.

Upgrading BTC's DAA to CW-144 (Scenario 2) does not substantially change the behavior of the variables, but there are a few noticeable effects. First, BTC's winning rate (Panel d, Scenario 2) and expected reward (Panel g, Scenario 2) are more vibrating as an immediate consequence of the DAA change. Second, BTC's block time is also more vibrant but is on average more centered around the target time of 600 seconds (Panel j, Scenario 2). Third, the DAA change does not affect BCH's and BSV's block time (Panels k and l, Scenario 2). This is because the winning rate and expected reward of BCH and BSV adjust with the BTC's DAA change (Panels e, f, h, i, Scenario 2).

On the contrary, downgrading the DAA of BCH and BSV to the original DAA causes a catastrophic impact on the SHA-256 market. Figure 6 displays the simulation results of Scenarios 3 and 4 compared to Scenario 1. If BCH downgrades the DAA to the original DAA, BCH's block time soars up and never returns to normal during the simulation period (Panel k, Scenario 3). This is because the original DAA fails to adjust the winning rate. The lines of Scenario 3 in Panels b and e explain what happens to the BCH in this scenario. BTC's halving makes the mining of BCH highly profitable. Nevertheless, the periodic nature of the original DAA prevents the immediate adjustment of the winning rate. As a result, the winning rate of BCH remains excessively high for a few days after halving (The first plateau of Scenario 3 of Panel e). As a consequence, the miners rush into BCH (The first plateau of Scenario 3 of Panel b), causing a massive block generation in a short time. Thus, when BCH accumulates 2,016 blocks, the original DAA finds it urgent to cut the winning rate, resulting in an overshoot of the winning rate cut (Panel e, Scenario 3). Because the expected reward rate is extremely low (Panel h, Scenario 3), miners flee from BCH and the hash rate sharply drops (Panel b, Scenario 3). The block time goes extremely high, and it never accumulates the next 2,016 blocks needed to adjust the winning rate, putting an end to BCH.

Interestingly, BCH's downgrading to the original DAA destabilizes the other two currencies as well. BTC's block time at first jumps up and then drops deep below the target level (Panel j, Scenario 3) because the hash rate flees to BCH at first and then comes back (Panel d, Scenario 3). Although it is hard to see in the current scale, the same side effect arises in BSV's block time (Panel l, Scenario 3). This confirms the negative externality of DAA adoption to other currencies in the same mining market.

Further downgrading BSV's DAA to the original DAA causes a similar problem. BSV's block time soars up and never returns to normal (Panel l, Scenario 4) because the winning rate maintains high and then sharply drops (Panel f, Scenario 4), and the hash rate of BSV first jumps up and then disappears (Panel f, Scenario 4). This poses a nontrivial externality on the stability of BTC and BCH. It poses a negative externality on BTC because BTC's block time further diverges from the target (Panel j, Scenario 4). However, it has a positive externality on BCH, because BCH's block time becomes

Table 7: Summary of Simulation Scenarios in Subsection 7.3

	BTC	BCH	BSV
Scenario 1 (actual)	Original DAA	CW-144	CW-144
Scenario 5 (new)	Original DAA	ASERT	CW-144
Scenario 6 (new)	ASERT	ASERT	ASERT

slightly shorter than Scenario 3 (Panel k, Scenario 4). Because BSV also fails to adjust the winning rate, some of the hash rates return to BCH.

In Section 4, we hinted that the adoption of DAA by a currency may have an eternality on the stability of the other currencies using the same hash function. This section's analysis thus confirmed this by counterfactual simulations. Therefore, the evaluation of the efficiency of cryptocurrency has to be jointly conducted at the mining market level under multiple stress test scenarios.

7.3 Performance of ASERT

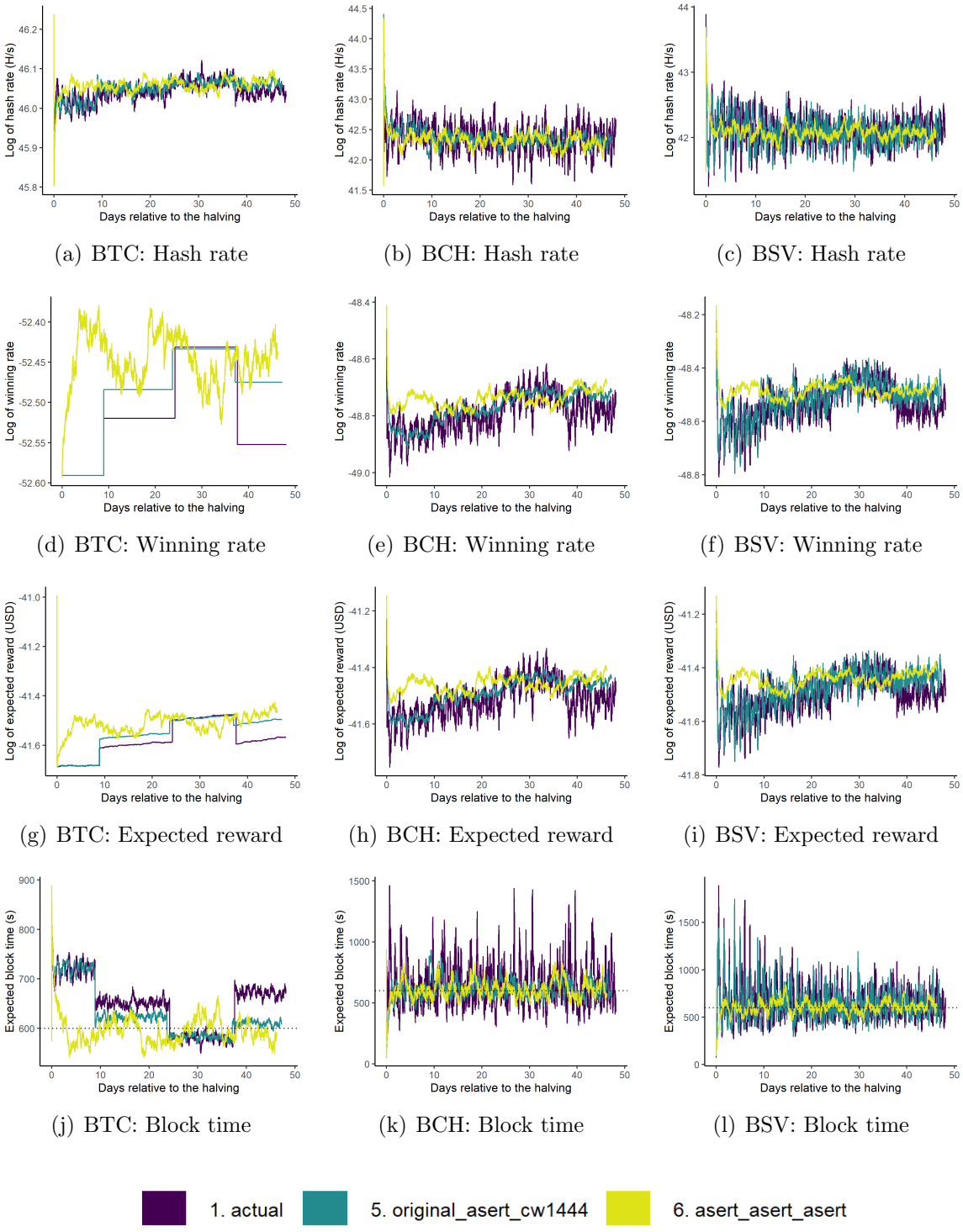
In November 2020 (after the third halving), BCH updated its DAA again—BCH adopted ASERT by abandoning CW-144. ASERT aims to further eliminate the periodic oscillations of the winning rate and hash rate. The BCH community expects that ASERT strengthens the stability of the block arrival rate.

To test the performance of ASERT compared to CW-144, we consider an additional counterfactual scenario: Scenario 5, BTC uses the original DAA, BCH uses ASERT, and BSV uses CW-144 already at the timing of BTC third halving and Scenario 6, all currencies use ASERT.

Figure 7 shows the simulation results under Scenarios 1, 5, and 6. For BCH, both CW-144 and ASERT quickly adjust the winning rate to an appropriate level (Panel e). By immediately adjusting the winning rate, they successfully change the miners' behavior and stabilize the hash rate at the desired level (Panel b). Consequently, both DAAs successfully absorb the BTC halving shock and quickly set the block arrival rate at the targeted level (Panel k).

Nevertheless, there is a noteworthy difference in the performance of CW-144 and ASERT. The variability of all variables, the hash rate, winning rate, expected reward, and block time of BCH, are substantially smaller under ASERT than CW-144. Although subtle, BCH's adoption of ASERT also creates a non-negligible positive externality on the stability of BTC and BSV. BTC's hash rate is more stable when BCH adopts ASERT (Panel a, Scenario 5) instead of CW-144 (Panel a, Scenario 1), even though the BTC's cross-elasticity to BCH is small. The hash rate of BSV is also slightly more stable when BCH adopts ASERT (Panel c, Scenario 5).

If all currencies adopt ASERT, the block time quickly converges to around the target



(a) BTC: Hash rate (b) BCH: Hash rate (c) BSV: Hash rate

(d) BTC: Winning rate (e) BCH: Winning rate (f) BSV: Winning rate

(g) BTC: Expected reward (h) BCH: Expected reward (i) BSV: Expected reward

(j) BTC: Block time (k) BCH: Block time (l) BSV: Block time

Figure 7: Counterfactual Simulations: Upgrading to ASERT

Note: The figure shows the counterfactual simulation results starting from the 1 block before the BTC halving. A line shows a single path of the simulation under the specified scenario. The scenario name represents the DAA of BTC, BCH, and BSV, respectively.

level and the variation around the target becomes substantially small as seen by Scenario 6 in Figure 7.

The additional stability created by ASERT is important because it is directly related to the cost of attacking the currency, as discussed in Section 2.3. In the next section, we introduce a measure of the security efficiency of cryptocurrency that formalizes this idea and compare the efficiency of different DAA profiles.

8 Security per Energy Consumption (SpEC)

8.1 Foundation of SpEC

If we only aim at stabilization of the block arrival rate, then the absolute size of the aggregate hash rate can be targeted at an arbitrary level. However, if we want to prevent a potential attack on the cryptocurrency, then the aggregate hash rate must be sufficiently large, because various attacks, such as the 51% attack, become possible if the attacker has a sufficiently large hash rate compared to the aggregate hash rate of other miners. The large aggregate hash rate means that the capability of ASIC machines necessary for a successful attack becomes unrealistically large.

A large aggregate hash rate, nevertheless, is costly because it requires higher energy consumption. The Cambridge Centre for Alternative Finance estimates that the total electricity consumption of BTC in June 2021, which is immediately after the third halving, amounts to 4.45 TWh. Cryptocurrencies are usually not equipped with any mechanism to determine the appropriate level of hash rate considering the trade-off between the security level and energy consumption. Therefore, the current aggregate hash rate and the security level can be excessive.

In this paper, we do not determine what the appropriate level of security or target aggregate hash supply is, because it requires a more comprehensive database of attacks on cryptocurrencies. Instead, we study the security efficiency of a cryptocurrency—the security level achieved per unit of energy consumption. Regardless of the security level choice, we still need to determine and adopt a system that achieves the target security level with minimum energy consumption.

To enable this, we propose a new measure for quantifying the ratio of the security level to the energy consumption and refer to the measure as *Security per Energy Consumption (SpEC)*. The major difference from the existing analysis is that we incorporate the endogenous response of miners and the algorithmic competition of cryptocurrencies by estimating and simulation the mining market model. Existing studies have assumed a constant hash rate. As we have demonstrated in Section 4, however, this assumption is implausible. As the winning rate adjusts reflecting the block time, the exchange rate changes reflecting the market condition, and the expected reward changes as a conse-

quence, then the hash rate supplied by miners also changes. This creates a moment when the aggregate hash rate is temporarily low and the system is particularly vulnerable to potential threats. Therefore, a security efficiency measure must incorporate the relation between the system and the endogenous variability of the hash rate into account. In addition, the security efficiency must be measured under various stress scenarios.

Operating Cost We measure the energy consumption for operating cryptocurrency for a period by the *average* hash rate during the period. If the power efficiency of mining ASIC and the price of the energy faced by miners are almost constant during the period, then the hash rate determines the amount of energy consumption and the management cost of the currency. Accordingly, the time average of the aggregate hash rate measures the level of the average energy consumption per unit time.

Attacking Cost as Security Level Conceptually, we measure the security level of a currency during a period by the *minimum* hash rate during the period. According to the security-game literature (e.g., Tambe, 2011), the security level of a system should be evaluated at its minimum level, because attackers are looking for opportunities such that attack is the easiest. In cryptocurrency, because the approximate hash rate is publicly observed, an attacker can find the best moment to carry out the attack. If the power efficiency of mining ASIC and the price of the energy faced by miners are almost constant during the period, then the minimum hash rate determines the energy consumption for attacking the currency, which is the security level. As we have shown so far, the hash rate endogenously fluctuates over time, and its fluctuation depends on the DAA profile of the mining market. Therefore, even if the average hash rate is similar, the minimum hash rate, i.e., the security level, can be significantly different.

Practically, we use the bottom 5th percentile of the hash rate during the simulation period as an indicator of the attacking cost. In the history of cryptocurrency, no attack has been reportedly implemented in an instance. It usually takes some time to complete the attack. Therefore, it would be more appropriate to measure the attacking cost by the minimum level below which the hash rate stays for a certain period. The choice of 5% is arbitrary: it can be 1% or 10%, depending on the context.

8.2 Calculation of SpEC

The security level per energy consumption is measured by the bottom 5th percentile of the hash rate per average hash rate during a period.

We denote the index of simulation paths by $p = 1, \ldots, P$. For each simulation path p, our procedure generates an epoch-level data. We denote the currency k's hash rate of epoch n in simulation path p by $h(k, n; p)$, and the duration of epoch n by $T(n; p)$.

Currency k's *energy consumption index of simulation path* p, denoted by $EC(k;p)$, is defined by

$$EC(k;p) \coloneqq \frac{\sum_{n=1}^{N(p)} T(n;p)h(k,n;p)}{\sum_{n=1}^{N(p)} T(n;p)}, \tag{19}$$

where $N(p)$ denotes the total number of epochs in path p. We define currency k's energy consumption by

$$EC(k) \coloneqq \frac{1}{P}\sum_{p=1}^{P} EC(k;p). \tag{20}$$

To define the security index, we first sort the epochs in ascending order of the hash rate. Formally, we define $\bar{n}(i;k,p)$ such that

$$h[k,\bar{n}(1;k,p);p] \leq h[k,\bar{n}(2;k,p);] \leq \cdots \leq h[k,\bar{n}(N(p);k,p);p], \tag{21}$$

i.e., $\bar{n}(i;k,p)$ is the epoch at which currency k's hash rate takes ith smallest value in path p. Next, we compute the epoch in which the hash rate is at the bottom 5th percentile, i.e., to find $i^*(k,p)$ such that

$$i^*(k,p) \coloneqq \min\left\{ i : \sum_{j\leq i} T[\bar{n}(j;k,p);p] \geq 0.05 \cdot \sum_{n=1}^{N(p)} T(n;p) \right\}. \tag{22}$$

Currency k's *security index of simulation path* p, denoted by $Security(k;p)$, is defined by

$$Security(k;p) \coloneqq h[k,\bar{n}(i^*(k,p);k,p);p]. \tag{23}$$

We define currency k's *security index* by

$$Security(k) \coloneqq \frac{1}{P}\sum_{p=1}^{P} Security(k). \tag{24}$$

Currency k's *Security per Energy Consumption index (SpEC)* is defined as

$$SpEC(k) \coloneqq \frac{Security(k)}{EC(k)}. \tag{25}$$

The numerator and the denominator are both measured by the hash rate (H/s). If we divide them by the hash rate of an ASIC machine, multiply the electricity consumption per second, and multiply the electricity price, they are converted into the electricity cost of the machine for attacking and operating the currency. If the distribution of the machine specification and electricity price is the same among attackers and ordinary miners, then the multiplied terms are canceled out between the numerator and denominator. Then, SpEC can be interpreted as the electricity cost for attacking the currency per the

electricity cost for operating the currency.

8.3 SpEC after BTC Halving

Table 8: SpEC after the BTC Halving

	DAA			SpEC		
	BTC	BCH	BSV	BTC	BCH	BSV
Scenario 1	Original	CW-144	CW-144	0.955	0.661	0.668
Scenario 2	CW-144	CW-144	CW-144	0.949	0.583	0.622
Scenario 3	Original	Original	CW-144	0.897	0.012	0.650
Scenario 4	Original	Original	Original	0.592	0.036	0.051
Scenario 5	Original	ASERT	CW-144	0.978	0.850	0.696
Scenario 6	ASERT	ASERT	ASERT	0.978	0.848	0.860

Note: For each scenario, we simulate 96 paths of blockchain for 60 days from one block before the BTC halving. SpEC can be interpreted as the electricity cost for attacking the currency per the electricity cost for operating the currency.

We conduct simulations according to the procedure described in Section 7.1. We consider the profile of DAAs analyzed in Section 7.2 and 7.3. For each scenario, we simulate 96 paths (i.e., $P = 96$) for 60 days from one block before the BTC halving (i.e., $\sum_{n=1}^{N(p)} T(n;p) = 60$ (days) for all p). The number of 96 is due to the number of cores of our computer. Table 8 summarizes SpEC of the three currencies under these scenarios.

In all scenarios, BTC achieves the best (i.e., highest) SpEC among the three currencies. For Scenarios 1, 2, 3, and 5, BTC's SpEC is roughly equal to or larger than 0.9, whereas BCH and BSV achieve at most 0.7 with CW-144 (BSV, Scenario 5) and 0.850 with ASERT (BCH, Scenario 5). In Scenario 4, SpEC is relatively low for all currencies, but BTC still achieves the highest SpEC. This is because BTC's hash supply is inelastic and its hash rate is extremely stable regardless of DAA. Indeed, BTC's SpEC does not improve by upgrading to CW-144 (Scenario 2).

By contrast, SpEC of BCH and BSV is highly sensitive to the choice of DAA. The original DAA is by far the worst: In Scenario 3, where BCH results in a catastrophic outcome (see Section 7.2), BCH's SpEC is only 0.012. Similarly, in Scenario 4, where both BCH and BSV collapse, BCH's and BSV's SpEC are merely 0.036 and 0.051, respectively. The analysis reveals that the high security level of BTC is mainly due to the inelastic hash supply and not the DAA. On the contrary, it is the efficient DAA that provides security to BCH and BSV.

The original DAA does not only create a moment of excessively low hash rate but also a moment of excessively high hash rate. When the winning rate is excessively high, the aggregate hash supply increases. However, it does not contribute to the security, because

Table 9: Energy Consumption Saving after the BTC Halving

(a) Energy-Saving

	DAA			Saving (GW)			
	BTC	BCH	BSV	BTC	BCH	BSV	Total
Scenario 1	Original	CW-144	CW-144	0.00	0.00	0.00	0.00
Scenario 2	CW-144	CW-144	CW-144	-0.05	-0.03	-0.01	-0.08
Scenario 3	Original	Original	CW-144	-0.66	-4.02	-0.01	-4.69
Scenario 4	Original	Original	Original	-5.68	-1.47	-0.66	-7.81
Scenario 5	Original	ASERT	CW-144	0.17	0.02	0.00	0.20
Scenario 6	ASERT	ASERT	ASERT	0.17	0.02	0.02	0.21

(b) Operating Cost

	DAA			Operating cost (GW)			
	BTC	BCH	BSV	BTC	BCH	BSV	Total
actual	Original	CW-144	CW-144	6.26	0.16	0.12	6.54

Note: For each scenario, we simulate 96 paths of blockchain for 60 days from one block before the BTC halving. It calculates the energy consumption saving (GW) due to the changes in the DAA profile.

attackers can wait until the difficulty is adjusted to a low level. Therefore, the high hash rate results in a waste of energy.

CW-144 brings a moderately high SpEC to BCH and BSV. For all scenarios where the currency uses CW-144, SpEC is approximately 0.6 to 0.7. ASERT is even better: BCH's SpEC in Scenario 5 where BCH uses ASERT is 0.85, which is a great improvement from SpEC of 0.66 in Scenario 1 where BCH uses CW-144. ASERT attenuates the variability of the hash rate more efficiently than CW-144. Thus, BCH's adoption of ASERT should have reinforced its security level. Scenario 6 shows that the SpEC of the three currencies substantially improves if all of them adopt ASERT.

The SpEC in the table confirms the externality of DAA choice on the stability and security of other currencies. We find that BTC's SpEC is substantially lower in Scenarios 3 (0.897) and 4 (0.592), where BCH and both BCH and BSV adopt the inefficient original DAA. Although subtle, comparing Scenarios 5 and 1, we find that BTC's and BSV's SpEC increase to 0.978 and 0.696 when BCH adopts ASERT from 0.955 and 0.668. Thus, the adoption of an inefficient DAA lowers the SpEC of other currencies, whereas the adoption of an efficient DAA strengthens the security and stability of other currencies.

8.4 Energy-Saving

It is also suggestive to evaluate the cost efficiency by the amount of wasted energy. If the actual DAA profile of $SpEC_1(k)$ is changed to an alternative DAA profile of $SpEC_n(k)$, then the energy consumption can be saved by $EC_1(k) \times [SpEC_1(k)/SpEC_n(k) - 1]$ while maintaining the security level at $Security_1(k)$. Because this is in terms of the hash rate (H/s), we divide it by the power efficiency (H/J) of mining ASIC machines to convert to the saved energy consumption per second (J/s = W). We use the average power efficiency of ASIC machines that were released between July 1, 2018, and June 30, 2020, for this calculation. Table 9 shows the energy saved by changing the DAA profile from the actual one. By upgrading all DAAs to ASERT, we could save on average 0.21 GW or 3.2% (= 0.21/6.54) of the actual operating cost. Although the change of SpEC is slim for BTC, the largest energy-saving is from BTC due to its size.

8.5 SpEC after BCH and BSV Halving

Tables A3 and A5 are the SpEC after BCH and BSV halving, corresponding to Table 8 of BTC halving. Tables A4 and A6 are the saved energy consumption after BCH and BSV halving, corresponding to Table A6. In reality, BSV halving followed a few days after BCH halving, but the simulation considers each halving separately.

We briefly describe the main takeaways. First, we observe that the BCH's and BSV's own halving shocks more significantly lowered BCH's and BSV's SpEC. Second, the own halving shock is hard to absorb even with CW-144. In the real world, the halving of BCH and BSV happened in a short period, and this might have helped these currencies to restore stability. Third, SpEC with ASERT remains high. ASERT clearly outperforms CW-144 at this level of reward shock. Fourth, BCH's and BSV's SpEC of the original DAA read high. However, this is because, during the simulation period, they have never been able to adjust the winning rate according to the hash rate lowered by halving. In the next difficulty adjustment, the winning rate will soar up and the hash rate will follow. Then, the original DAA will waste a large amount of energy, and BCH's and BSV's SpEC under these scenarios will be worsened. However, the effect is not yet seen in the 60-day window.

9 Concluding Remarks

We studied the structure of PoW cryptocurrencies' mining market and investigated the stability, security, and energy consumption of cryptocurrencies as a decentralized transaction system. To do so, we focused on BTC, BCH, BSV, the largest cryptocurrency group that adopts the same hash function (SHA-256), and analyzed how their DAAs and miners respond to a large exogenous shock on the expected reward, the third halving.

We found that miners responded to changes in the reward and cryptocurrencies algorithmically competed for miners' computational work in the mining market. We introduced a novel measure of security efficiency, SpEC, and found that BTC achieved the high SpEC because of the inelastic hash supply and not because of the DAA. By contrast, BCH's and BSV's hash supply was highly elastic and stabilized only with the efficient DAA such as CW-144. If BCH and BSV had used the original DAA at the timing of the third halving, then BCH and BSV would have collapsed. The analysis also revealed the externality of DAA choice on the stability and security of other currencies.

According to the analysis, ASERT was the state-of-the-art DAA and clearly outperformed the original DAA and CW-144 in terms of the stability of the hash rate and the security level achieved per unit of energy consumption. BTC's original DAA could cause serious trouble if the hash supply of BTC turned to be elastic in the future.

There are several limitations to our analysis. First, we identified the hash supply elasticity in the reduced form but did not attribute it to the structural factors. For example, the low hash supply elasticity of BTC may be due to the thick exchange market. Second, although we estimated the aggregate hash supply by exploiting the third halving, which was the largest shock ever as of 2021, it was not sure how miners could react to a substantially greater shock to the expected reward. With a greater shock, the miners' behavior can be different—BTC's loyal miners may shut down their machines to save the electricity cost, and therefore, BTC may face an elastic hash supply problem. Third, this paper's analysis was limited to relatively large cryptocurrencies. Smaller currencies have attempted to stabilize their hash rate using technologies other than DAA, for example, by adopting ASIC-resistant hash functions and accepting multiple hash functions for mining. Finally, the analysis was limited to PoW cryptocurrencies. Extending the analysis of security efficiency to cryptocurrencies using general consensus mechanisms would be interesting future research.

References

Aggarwal, V. and Y. Tan (2019): "A Structural Analysis of Bitcoin Cash's Emergency Difficulty Adjustment Algorithm," Working Paper.

Back, A. (2002): "Hashcash-a Denial of Service Counter-Measure," Technical Report.

Calonico, S., M. D. Cattaneo, M. H. Farrell, and R. Titiunik (2017): "Rdrobust: Software for Regression-Discontinuity Designs," *The Stata journal*, 17, 372–404.

Calonico, S., M. D. Cattaneo, and R. Titiunik (2014): "Robust Nonparametric Confidence Intervals for Regression-Discontinuity Designs," *Econometrica*, 82, 2295–2326.

Chiu, J. and T. V. Koeppl (2019): "Blockchain-Based Settlement for Asset Trading," *Review of Financial Studies*, 32, 1716–1753.

Cong, L. W. and Z. He (2019): "Blockchain Disruption and Smart Contracts," *Review of Financial Studies*, 32, 1754–1797.

Dwork, C. and M. Naor (1992): "Pricing via Processing or Combatting Junk Mail," in *Annual International Cryptology Conference – CRYPTO' 92*, 139–147.

Eyal, I. and E. G. Sirer (2018): "Majority Is Not Enough: Bitcoin Mining Is Vulnerable," *Communications of the ACM*, 61, 95–102.

Fiat, A., A. Karlin, E. Koutsoupias, and C. Papadimitriou (2019): "Energy Equilibria in Proof-of-Work Mining," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 489–502.

Goren, G. and A. Spiegelman (2019): "Mind the Mining," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 475–487.

Hinzen, F. J., K. John, and F. Saleh (2019): "Proof-of-Work's Limited Adoption Problem," Working Paper.

Huberman, G., J. D. Leshno, and C. Moallemi (2021): "Monopoly without a Monopolist: An Economic Analysis of the Bitcoin Payment System," *Review of Economic Studies*.

Matsushima, H. and S. Noda (2020): "Mechanism Design with Blockchain Enforcement," Working Paper.

Nakamoto, S. (2008): "Bitcoin: A Peer-to-Peer Electronic Cash System," Bitcoin White Paper.

Narayanan, A., J. Bonneau, E. Felten, A. Miller, and S. Goldfeder (2016): *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Princeton University Press.

Noda, S., K. Okumura, and Y. Hashimoto (2020): "An Economic Analysis of Difficulty Adjustment Algorithms in Proof-of-Work Blockchain Systems," in *Proceedings of the 21st ACM Conference on Economics and Computation*, 611–611.

Prat, J. and B. Walter (2021): "An Equilibrium Model of the Market for Bitcoin Mining," *Journal of Political Economy*, 129, 2415–2452.

Roşu, I. and F. Saleh (2021): "Evolution of Shares in a Proof-of-Stake Cryptocurrency," *Management Science*, 67, 661–672.

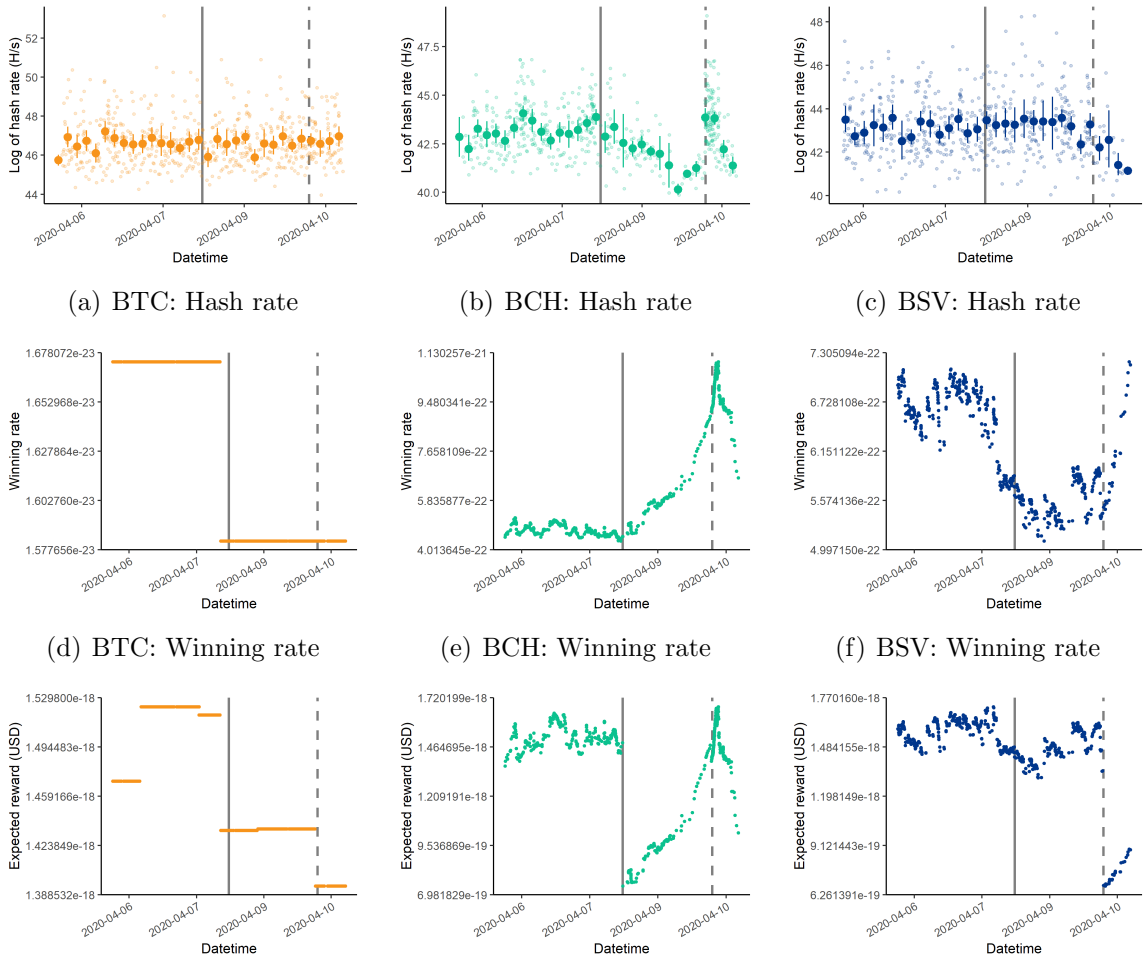
Saleh, F. (2020): "Blockchain without Waste: Proof-of-Stake," *Review of Financial Studies*, 34, 1156–1190.

Shibuya, Y., G. Yamamoto, F. Kojima, E. Shi, S. Matsuo, and A. Laszka (2021): "Selfish Mining Attacks Exacerbated by Elastic Hash Supply," in *Financial Cryptography and Data Security*, 269–276.

Tambe, M. (2011): *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*, Cambridge University Press.

Taylor, M. B. (2017): "The Evolution of Bitcoin Hardware," *Computer*, 50, 58–66.

A Appendix



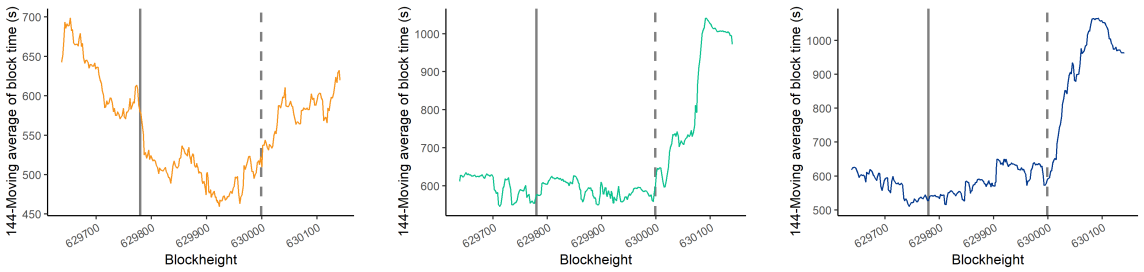
(a) BTC: Hash rate (b) BCH: Hash rate (c) BSV: Hash rate

(d) BTC: Winning rate (e) BCH: Winning rate (f) BSV: Winning rate

(g) BTC: Expected reward (h) BCH: Expected reward (i) BSV: Expected reward

Figure A1: Hash rate, winning rate, and expected reward around the BCH halving

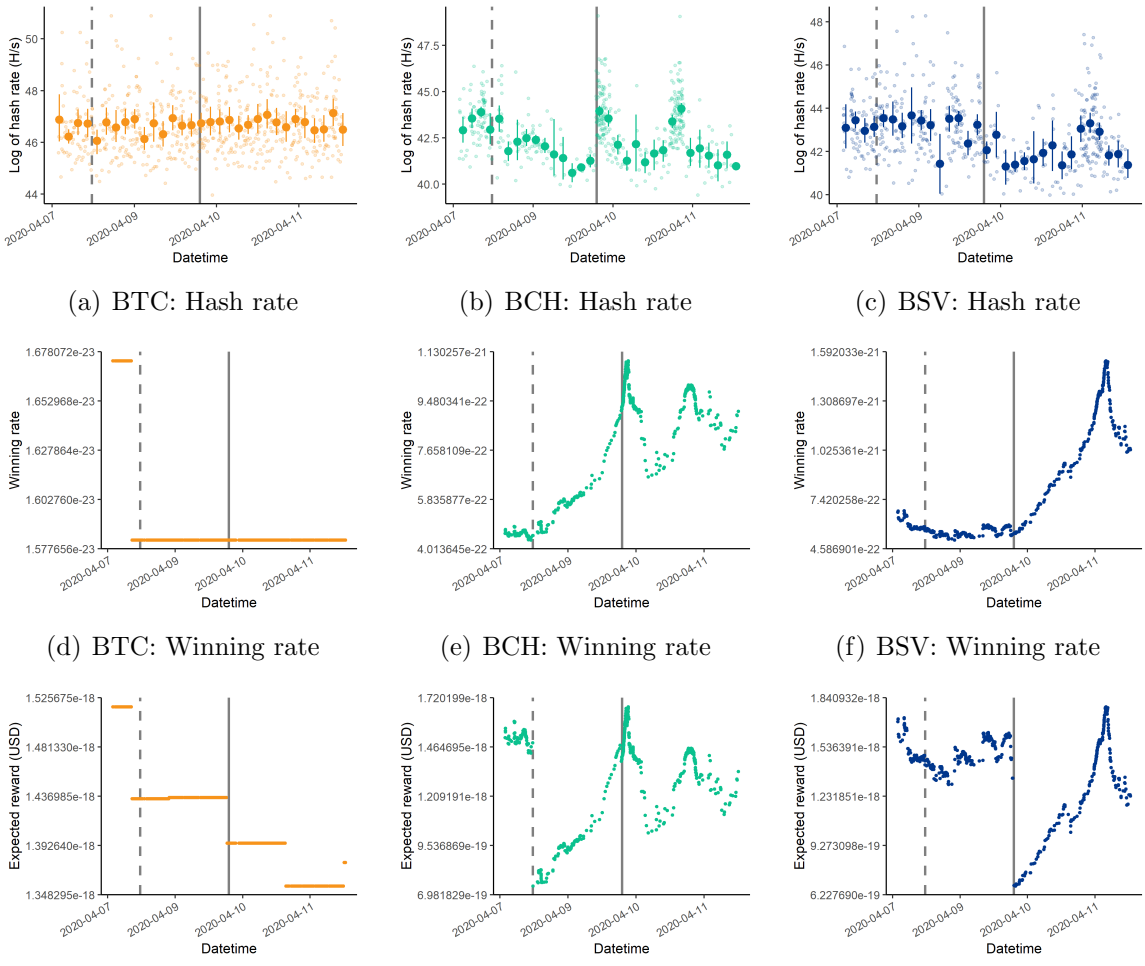
Note: In hash rate figures, binned averages and their 95% confidence intervals are drawn on the raw data points. In winning rate and expected reward figures, only raw data points are drawn. The solid vertical line is the timing of BCH halving and the dashed vertical line is the timing of BSV halving.



(a) BTC (b) BCH (c) BSV

Figure A2: 144-block moving averages of block time around the BCH halving

Note: The solid vertical line is the timing of BCH halving and the dashed vertical line is the timing of BSV halving.



(a) BTC: Hash rate (b) BCH: Hash rate (c) BSV: Hash rate

(d) BTC: Winning rate (e) BCH: Winning rate (f) BSV: Winning rate

(g) BTC: Expected reward (h) BCH: Expected reward (i) BSV: Expected reward

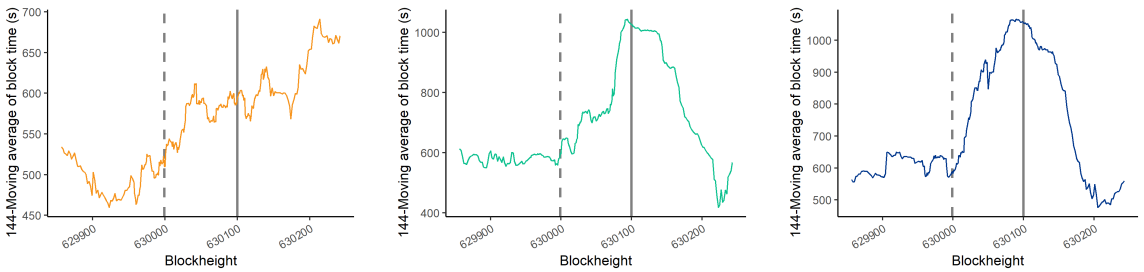
Figure A3: Hash rate, winning rate, and expected reward around the BSV halving

Note: In hash rate figures, binned averages and their 95% confidence intervals are drawn on the raw data points. In winning rate and expected reward figures, only raw data points are drawn. The solid vertical line is the timing of BSV halving and the dashed vertical line is the timing of BCH halving.

Table A1: The effects of BCH halving on hash rate and block time

	Log of hash rate (H/s)		
	BTC	BCH	BSV
Estimate	-0.0951	-0.712	0.415
Std.error	(0.151)	(0.155)	(0.187)
Nobs.left	3769	4030	4025
Nobs.right	4227	3926	3909
H.left	6.82e+05	9.47e+05	4.48e+05
H.right	4.4e+05	4.23e+05	3.03e+05
B.left	1.03e+06	1.45e+06	7.86e+05
B.right	7.57e+05	9.17e+05	6.7e+05
Order.regression	2	2	2
Kernel	Triangular	Triangular	Triangular
Bwselect	Msetwo	Msetwo	Msetwo

Note: Standard errors are in the parentheses. The estimates are the local average treatment effects at the BCH halving time identified by a regression discontinuity design. The bandwidth is chosen for both sides of the cutoff by using Calonico et al. (2014). H.left and H.right are bandwidth for the estimation of the regression functions and B.left and B.right are for the estimation of the bias. The bias is corrected and the standard errors are robust to the asymptotic variance due to the bias term using Calonico et al. (2014). Local quadratic functions are used to fit the regression functions with a triangular kernel function.



(a) BTC (b) BCH (c) BSV

Figure A4: 144-block moving averages of block time around the BSV halving

Note: The solid vertical line is the timing of BSV halving and the dashed vertical line is the timing of BCH halving.

Table A2: The effects of BSV halving on hash rate and block time

	Log of hash rate (H/s)		
	BTC	BCH	BSV
Estimate	0.0905	1.89	-1.5
Std.error	(0.126)	(0.24)	(0.229)
Nobs.left	3789	3905	4037
Nobs.right	4219	4014	3945
H.left	7.74e+05	1.91e+05	4.85e+05
H.right	6.17e+05	4.54e+05	2.95e+05
B.left	1.16e+06	5.38e+05	8.22e+05
B.right	9.64e+05	7.12e+05	6.38e+05
Order.regression	2	2	2
Kernel	Triangular	Triangular	Triangular
Bwselect	Msetwo	Msetwo	Msetwo

Note: Standard errors are in the parentheses. The estimates are the local average treatment effects at the BSV halving time identified by a regression discontinuity design. The bandwidth is chosen for both sides of the cutoff by using Calonico et al. (2014). H.left and H.right are bandwidth for the estimation of the regression functions and B.left and B.right are for the estimation of the bias. The bias is corrected and the standard errors are robust to the asymptotic variance due to the bias term using Calonico et al. (2014). Local quadratic functions are used to fit the regression functions with a triangular kernel function.

Table A3: SpEC after the BCH Halving

	DAA			SpEC		
	BTC	BCH	BSV	BTC	BCH	BSV
Scenario 1	Original	CW-144	CW-144	0.957	0.259	0.599
Scenario 3	Original	Original	CW-144	0.943	0.543	0.693
Scenario 4	Original	Original	Original	0.872	0.155	0.067
Scenario 5	Original	ASERT	CW-144	0.975	0.767	0.708
Scenario 6	ASERT	ASERT	ASERT	0.975	0.808	0.861

Note: For each scenario, we simulate 96 paths of blockchain for 60 days from one block before the BCH halving. SpEC can be interpreted as the electricity cost for attacking the currency per the electricity cost for operating the currency.

Table A4: Energy Consumption Saving after the BCH Halving

	DAA			Saving (GW)			
	BTC	BCH	BSV	BTC	BCH	BSV	Total
Scenario 1	Original	CW-144	CW-144	0.00	0.00	0.00	0.00
Scenario 3	Original	Original	CW-144	-0.11	0.00	0.03	-0.08
Scenario 4	Original	Original	Original	-0.83	-0.04	-6.26	-7.13
Scenario 5	Original	ASERT	CW-144	0.13	0.08	0.03	0.23
Scenario 6	ASERT	ASERT	ASERT	0.13	0.08	0.05	0.26

Note: For each scenario, we simulate 96 paths of blockchain for 60 days from one block before the BCH halving. It calculates the energy consumption saving (GW) due to the changes in the DAA profile.

Table A5: SpEC after the BSV Halving

	DAA			SpEC		
	BTC	BCH	BSV	BTC	BCH	BSV
Scenario 1	Original	CW-144	CW-144	0.959	0.546	0.432
Scenario 3	Original	Original	CW-144	0.933	0.181	0.344
Scenario 4	Original	Original	Original	0.834	0.010	0.096
Scenario 5	Original	ASERT	CW-144	0.966	0.811	0.328
Scenario 6	ASERT	ASERT	ASERT	0.975	0.843	0.781

Note: For each scenario, we simulate 96 paths of blockchain for 60 days from one block before the BSV halving. SpEC can be interpreted as the electricity cost for attacking the currency per the electricity cost for operating the currency.

Table A6: Energy Consumption Saving after the BSV Halving

	DAA			Saving (GW)			
	BTC	BCH	BSV	BTC	BCH	BSV	Total
Scenario 1	Original	CW-144	CW-144	0.00	0.00	0.00	0.00
Scenario 3	Original	Original	CW-144	-0.22	-0.64	-0.14	-1.00
Scenario 4	Original	Original	Original	-1.36	-8.48	-0.15	-9.98
Scenario 5	Original	ASERT	CW-144	0.05	0.04	-0.17	-0.08
Scenario 6	ASERT	ASERT	ASERT	0.11	0.04	0.04	0.20

Note: For each scenario, we simulate 96 paths of blockchain for 60 days from one block before the BSV halving. It calculates the energy consumption saving (GW) due to the changes in the DAA profile.