Voting Games to Model Protocol Stability and Security of Proof-of-Work Cryptocurrencies

Sanjay Bhattacherjee[1(✉)] and Palash Sarkar[2]

[1] Institute of Cyber Security for Society and School of Computing, Keynes College, University of Kent, Canterbury CT2 7NP, UK
s.bhattacherjee@kent.ac.uk

[2] Applied Statistics Unit, Indian Statistical Institute, 203, B.T. Road, Kolkata 700108, India
palash@isical.ac.in

https://www.kent.ac.uk/computing/people/3156/bhattacherjee-sanjay, https://www.isical.ac.in/~palash/

Abstract. We model the *protocol stability* and the *security* of proof-of-work cryptocurrencies using voting games. The first game, which we call the Rule Game, pertains to the scenario where the cryptocurrency miners engage in a voting procedure to accept or reject a proposal for change of the cryptocurrency protocol. The second game, which we call the Attack Game, refers to the scenario where a group of miners can form a coalition to launch a 51% attack on the system and consequently change a portion of the history of the underlying blockchain, thus defeating its promise of immutability. For the Attack Game, we define progressively granular notions of security all of which are based on the key concept of minimal winning coalitions from voting game theory. For both the Rule Game and the Attack Game, we show practical applicability of tools from voting game theory using a snapshot of real world data for Bitcoin. In particular, this highlights the fragile nature of the security of Bitcoin with respect to 51% attacks.

Keywords: Voting games · Cryptocurrency · Bitcoin (BTC) · Preventive power · Protocol change · 51% attack · Security

1 Introduction

Since the proposal of Bitcoin (BTC) by the eponymous Satoshi Nakamoto [29], many cryptocurrencies have been proposed. Bitcoin though, remains the most popular and valuable cryptocurrency. The underlying blockchain technology has found numerous applications as well. There are multiple dimensions to research on blockchains. In this work, we model (1) the procedure to change the protocol and (2) the security of proof-of-work cryptocurrencies using voting games. The ensuing analysis shows the practical applicability of tools from voting game theory in the scrutiny of cryptocurrency systems.

F. Fang et al. (Eds.): GameSec 2022, LNCS 13727, pp. 297–318, 2023.
https://doi.org/10.1007/978-3-031-26369-9_15

Voting Games

Voting Games are typically used to model and analyse the decision-making procedure in scenarios that involve several entities or players. Of practical interest are weighted majority voting games. In such games, each player has a pre-assigned weight. A coalition of players wins if the sum of the weights of all the players in the coalition is at least a certain pre-specified fraction q of the total sum of the weights of all the players.

Voting Games in Proof-of-Work Cryptocurrencies

A proof-of-work cryptocurrency has an underlying linear blockchain data structure. Blocks are added to this structure by individual miners and mining pools who find the proof-of-work for a new block as a solution to a computational puzzle. For simplicity, we use the name *miner* for any mining entity - an individual or a pool. The more the computational power or the "hash rate" (explained in [3]) invested by a miner, more are its chances to succeed in mining a new block. This feature indicates the presence of an implicit voting game in cryptocurrencies.

In this work, we model two key functional aspects of a proof-of-work cryptocurrency system using voting games. By adding new blocks to the system, a miner participates as a player in both these games. The weight of a miner in these games is proportional to the fraction of the total network's hash rate that it controls. The winning threshold q is fixed depending upon the functional aspect that is being modelled.

The first functional aspect that we model is the procedure to change the *rules governing the cryptocurrency* described as *protocols*. The miners and other participating entities of the system follow the protocols. Being a decentralised and distributed system, the protocols are not determined or distributed by any central authority. They are initially agreed upon by the miners at the time of inception and as part of the specification, there is a well-defined procedure to change the protocols through consensus of the miners.

In the context of Bitcoin, a protocol modification happens through a Bitcoin Improvement Proposal (BIP). We show that a BIP [7] can be viewed as a weighted majority voting game among the miners where the winning threshold q is 0.95 (although there are instances of lower thresholds being used as well [6]). More generally, we use the term Rule Game to denote the voting games arising in the context of protocol change of any cryptocurrency (details in [3]). The purpose of having a high winning threshold in a Rule Game is to achieve near unanimity for a protocol change to take place. From the viewpoint of voting games, having a high winning threshold results in several miners becoming blockers.

The core principles of the Bitcoin blockchain have been emulated to varied extents by perhaps all cryptocurrencies that have been developed thereafter. Just like BIP, protocol changes requiring consensus are a very common phenomenon in Ethereum [16] and other proof-of-work blockchains [26,30].

It is perhaps natural that a procedure to achieve consensus among multiple agents gives rise to situations of cooperation and conflict and hence is a topic of

certain game theoretic interest. When the miners are in agreement over the protocols, we say that the cryptocurrency is *stable*. Proposals for protocol change that lead to conflicts among miners make the system less stable. An extreme consequence of disagreement between the miners over protocol changes is the *forking* of the cryptocurrency into two separate ones. A fork splits the network and even though some of the miners may continue to mine on both the chains, it certainly reduces the hash rate invested in both cryptocurrencies as compared to the original one. This makes both chains more vulnerable to attacks compared to the original chain. There are of course other socio-economic implications of rifts between miners over protocol changes leading to reduced stability. Some of the highly debated cryptocurrency protocol changes include SegWit [5] that led to a new cryptocurrency Bitcoin Cash (BCH) forked from the BTC chain, and the split between Ethereum and Ethereum Classic [37] after the DAO attack. This shows that protocol stability is a major challenge for proof-of-work cryptocurrencies. We use the Coleman preventive power measure (as argued in [3]) to capture the influence of a miner in a Rule Game for protocol stability.

The second functional aspect that we model is the immutability of the underlying blockchain data structure ensured by the miners not attacking the system. If a coalition of miners acquire at least 51% of the hash rate of the system, then with non-negligible probability, such a coalition can engage in changing a substantial part of the blockchain and consequently double spending of the currency. At any point of time, some of the miners may be attempting to attack the system while the others are preventing it. We model this as a voting game between the miners with the winning threshold $q = 0.51$. We call this an Attack Game.

There have been several instances of 51% attacks [1] on various cryptocurrencies, including Bitcoin [36], Bitcoin Gold [9], Ethereum Classic [13], Bitcoin SV [8], Verge [10], Litecoin Cash [27], Vertcoin [28], and many more. These attacks have led to owners of the respective cryptocurrencies suffer huge losses. The Attack Game and its detailed characterisation and analysis is thus extremely important for the cryptocurrency space.

Our analysis of the Attack Game is based on questions formulated around the crucial notion of minimal winning coalitions. For example, one may wish to know the minimum cardinality of any minimal winning coalition; another relevant question would be the minimum cardinality of any minimal winning coalition containing a particular miner. We define related notions of security to provide concrete answers to such questions.

The top level security notion that we introduce is that of the cryptocurrency being $\mathfrak{c}$-secure, where $\mathfrak{c}$ is the maximum integer such that there is no minimal winning coalition for the Attack Game of size $\mathfrak{c}$ or lower. In other words, this means that no coalition of $\mathfrak{c}$ or lesser number of miners will be able to attack the cryptocurrency. Starting from this security notion, we define progressively granular notions of security finally leading to the following security consideration. Suppose L is a set of 'large' miners, i.e., miners who control a significant amount of the network hash rate. Let (S_1, S_2) be a partition of L. We consider the

scenario where the miners in S_1 are trying to compromise the system, but, the miners in S_2 are not doing so. Then the question is how much support do the miners in S_1 require from miners outside L to attack the system. We define the cryptocurrency to be $(L, \mathfrak{c}_1, \mathfrak{c})$-secure if for any subset S_1 of L of size $\mathfrak{c}_1$, more than $\mathfrak{c}$ miners outside of L are required to form a coalition with S_1 and successfully attack the system.

To the best of our knowledge, the game theoretic modelling, the definitions of security and the mechanisms for analysing the stability of a cryptocurrency that we introduce have not appeared before either in the voting game or in the cryptocurrency literature.

Snapshot Analysis

To actually compute the power measures and the minimal winning coalitions in a cryptocurrency game, it is required to obtain the weights of the miners which are their hash rates. The values of the hash rates are not directly available. Instead, they need to be estimated. A simple estimate can be obtained based on the assumption that the hash rate is proportional to the number of blocks mined by a miner in a given interval of time. This provides a snapshot estimate of the actual hash rate of the miners.

For Bitcoin, we use such a snapshot estimate and show how to perform a meaningful analysis of the Rule and the Attack Games. The results for the Attack Game throw light on the vulnerability of Bitcoin. For example, in the time interval that we have considered, there were 8 different coalitions each consisting of 5 known miners such that any of these coalitions could have won the Attack Game. Given that the Bitcoin market is worth billions of dollars, it is a disconcerting thought that there can be several coalitions of a small number of parties who can disrupt the whole system. While a cryptocurrency does not have any central authority, the power to compromise the system residing in the hands of a few parties certainly detracts from the purposed goal of a completely decentralised system.

Related Works: An introduction to voting games can be found in [11] and an extensive overview of the topic is provided in [18]. The idea of measuring power in a voting game was introduced in [34,35]. The use of swings to measure voting power of players was suggested in [2,33]. Later work by [14] provided alternative proposals for capturing the notions of preventive and initiative powers of players. Voting power measures based on the idea of minimal winning coalitions were suggested in [15,20,21]. We refer to [24,31] for surveys on voting games.

There is a fairly large and growing literature on cryptocurrencies. Some game theoretic aspects of cryptocurrencies have already been studied [19,22,23,25]. To the best of our knowledge, the applications of weighted majority voting games to model protocol stability and security of cryptocurrencies have not been considered earlier.

2 Background and Preliminaries

In this section we provide a high level overview of proof-of-work cryptocurrencies and voting games to help understand how voting games arise in these cryptocurrencies. For further details on blockchains, the reader may consult [12,32]. More details on voting games can be found in [11,18].

2.1 Blockchain Basics

A typical *blockchain* is fundamentally a linear data structure that is built and maintained by a distributed system of computing nodes connected through peer-to-peer networking. The data structure is composed of a "chain of blocks" $\mathbf{B} = \{B_0, B_1, \ldots\}$. Each block B_i is made up of a header h_i and a body b_i. For $i \in \{1, 2, \ldots\}$, the header h_i of every block B_i contains a digest of the body b_i and the hash value $\mathcal{H}(h_{i-1})$ where $\mathcal{H}$ is a cryptographically secure hash function and h_{i-1} is the header of the previous block B_{i-1} in the chain. A cryptocurrency system $\mathfrak{C}$ is a blockchain where the information on the creation and transfer of the currency is stored in the body of each block.

In Bitcoin, a key criteria for validity of a new block B_r is that the hash of its header $\mathcal{H}(h_r)$ should have a certain number of leading zeros also called the *difficulty*. The node that produced B_r is called the *miner* of that block. The Bitcoin protocol allows the miner to include a random number or *nonce* η_r in the header h_r of B_r so that its hash $\mathcal{H}(h_r)$ has the required difficulty. The nonce thus found through mining is called *proof-of-work*. Finding the proof-of-work is a computational challenge, and the probability of success is proportional to the number of hashes that the miner can compute per second (called its *hash rate*). Miners may increase their individual hash rates to increase the number of blocks they are able to mine and thus their profits thereof. Multiple miners may come together to *pool* their computing resources and mine blocks together. This increases the probability of successfully mining the block and results in more profits within a specified time frame. For the purpose of this paper, we do not distinguish between miners and mining pools.

In general for any blockchain system, we use the name *miner* to denote an entity that creates a block, and the term *weight* to denote the amount of computational resource (hash rate) they have invested in the system. The weight is their contribution of resources in running and maintaining the system.

Blockchain Security. The primary security promise offered by a blockchain system is the immutability of its history. As new blocks are appended to $\mathbf{B}$, the existing blocks get farther away from the last block of $\mathbf{B}$. If a block $B_i \in \mathbf{B}$ is altered to B_i' by an attacker, its header h_i and the hash $\mathcal{H}(h_i)$ will change. The new hash $\mathcal{H}(h_i')$ will not tally with the hash value $\mathcal{H}(h_i)$ stored in the header h_{i+1} of B_{i+1}.

To alter a proof-of-work blockchain, the attacker will have to find a new proof-of-work η_i' for this altered block B_i'. To make the system consistent, $\mathcal{H}(h_i')$ will then have to be saved in B_{i+1} changing it to B_{i+1}'. So now the attacker

has to find a new proof-of-work η'_{i+1} for B'_{i+1} to be saved in the header of B_{i+2} and so on. Thus a single change in B_i will result in a cascade of changes to all subsequent blocks in the chain until the last block in the blockchain. If an attacker controls *more than half* of the (computational) resources of the network, it can launch such an attack on **B**. Such an attack is thus called the "*51% attack*". As mentioned in Sect. 1, there are many known instances of such attacks on cryptocurrencies.

Blockchain Protocol Stability. A decentralised blockchain system is completely maintained by its miners. The maintainability of a blockchain includes the ability to change the blockchain protocol itself. Typically, a protocol change is proposed on some off-chain forum like a common web-portal, mailing list, etc. that is popularly followed. The block headers contain a set of bits that are mapped to a change proposal though an off-chain mechanism as well. For every new block created, its creator denotes their support/opposition to the currently active protocol change proposals by setting/unsetting the corresponding bits. Only when an overwhelming majority of the blocks added to **B** within a predetermined period of time have support in favour of a change, it is incorporated into the protocol. As discussed in Sect. 1, protocol change proposals have resulted in several debates between contending groups of miners and in the worst case of disagreements have even led to the forking of blockchains.

2.2 Voting Game Basics

Let $N = \{A_1, A_2, \ldots, A_n\}$ be a set of n players. A subset of N is called a coalition and the power set of N, i.e., the set of all possible coalitions is denoted by 2^N. A voting game G comprising of the players in N is given by its characteristic function $\Psi_G : 2^N \rightarrow \{0, 1\}$ where a winning coalition is assigned the value 1 and a losing coalition is assigned the value 0. The set of all winning coalitions is denoted by $W(G)$ and the set of all losing coalitions is denoted by $L(G)$. For a finite set S, $\#S$ will denote the cardinality of S.

For any $S \subseteq N$, $A_i \in N$ is called *swing* in S if $A_i \in S$, $\Psi_G(S) = 1$ but $\Psi_G(S \setminus \{A_i\}) = 0$. The number of subsets $S \subset N$ such that A_i is a swing in S will be denoted by $m_G(A_i)$. A coalition $S \subseteq N$ is called a *minimal winning coalition* if $\Psi_G(S) = 1$ and there is no $T \subset S$ for which $\Psi_G(T) = 1$. A coalition $S \subseteq N$ is called a *minimal blocking coalition* [4] in G if $\Psi_G(N \setminus S) = 0$ and for any non-empty $T \subset S$, $\Psi_G(N \setminus T) = 1$. A player A_i is called a blocker if $\{A_i\}$ is a minimal blocking coalition.

Definition 1. *Consider a triplet $(N, \mathbf{w}, q)$, where $N = \{A_1, \ldots, A_n\}$ is a set of players, $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ is a vector of non-negative weights with w_i being the weight of A_i and q is a real number in $(0, 1)$. Let $w_S = \sum_{A_i \in S} w_i$ denote the sum of the weights of all the players in the coalition $S \subseteq N$. Then,*

$w_N = \sum_{A_i \in N} w_i$. *A weighted majority voting game* $G = (N, \mathbf{w}, q)$ *is defined by the characteristic function* $\Psi_G : 2^N \to \{0, 1\}$ *as follows.*

$$\Psi_G(S) = \begin{cases} 1 & \text{if } w_S/w_N \geq q, \\ 0 & \text{otherwise.} \end{cases}$$

3 Voting Games Arising from Proof-of-Work Cryptocurrencies

Proof-of-work cryptocurrencies give rise to at least two weighted majority voting games. We first describe the common features of both the games.

The Players, Their Weights and the Winning Threshold: The miners and the mining pools are the players in the game. We will simply write miner to mean either an individual miner or a mining pool. Intuitively, the weight of a player is its ability to mine a new block. In a proof-of-work based system, suppose there are k miners having hash rates $h_1, \ldots, h_k$ with the total hash rate h of the system being equal to $h = h_1 + \cdots + h_k$. *The weights of the miners are the hash rates* $h_1, \ldots, h_k$. For any positive real number λ, it is possible to use $\lambda h_1, \ldots, \lambda h_k$ as the weights without changing the characteristic function of the game. The winning threshold depends on the game as explained in Sects. 3.1 and 3.2.

Approximations of the Hash Rates of the Players: Being a decentralised and distributed system, the hash rates of the miners of a proof-of-work cryptocurrency system are not directly available. However, the proportion of blocks contributed by a miner to the system should indicate its proportion of the total hash rate.

Several Internet sites provide the number of blocks mined by various miners in a given time period. From this, it is possible to obtain an estimate of the hash rate of the miners. Suppose that for a given time period, a list $(A_1, b_1), \ldots, (A_k, b_k)$ is available indicating that the miner A_i has mined b_i blocks in that time period. It is reasonable to assume that the fraction of blocks mined by A_i in a given time period is proportional to h_i/h. Under this assumption, an estimate of the proportional hash rate of the miner A_i can be taken to be b_i/b where $b = b_1 + \cdots + b_r$. Since for a particular time period, b is constant, the weight of a miner can be taken to be the number of blocks it has mined in the given time period. The choice of this time period is not definite. It should not be too long since then miners who had been active earlier, but are no longer active will get positive weights. Neither should it be too short as then the estimate would not be accurate.

The suggested method of approximating the weights of the miners has a limitation. The actual weight of a miner is its hash rate while the approximate weight of a miner vote is taken to be the number of blocks that it is able to mine in the given time interval. While this number is expected to be proportional to the hash rate of the miner, it is not an exact correspondence. For example, it

is possible that miners with low weights are unable to mine any block in the required time interval. As a result, the approximate weights of these miners will be zero, even though they have positive hash rates. While this is indeed an issue, for the miners with high hash rates, the proportion of mined blocks would be quite close to the proportional hash rates.

We note that the theoretical aspects of our work are not dependent on the method employed to obtain estimates of the hash rates of the miners. The theory that we develop could be equally well applied to hash rates estimated using some other method.

3.1 The Rule Game

The procedure for protocol change in a proof-of-work cryptocurrency has been briefly described in Sects. 1 and 2. For Bitcoin, this is done through a BIP. The Rule Game arising from a BIP occurs as follows. The difficulty (minimum number of leading zeros) of the hash value for a valid Bitcoin block is fixed for every 2016 blocks. Such a window of 2016 blocks is called the *target period* that typically lasts for 2 weeks. A BIP has to be decided upon within 26 consecutive target periods (around a year's time). Once started and before time-out, each of the 26 target periods creates a new Rule Game for the BIP. The winning threshold for a BIP is typically 95%. So at least 95% of the 2016 blocks in a target period must indicate support for the BIP (by setting the respective bit for the BIP in the block header) for it to be considered as accepted and active. So BIP games are played during fixed time intervals which are the periods of constant difficulty. Coalitions of players can form for the activation (or blocking) of a BIP. The interests of the members of such a coalition would be aligned, i.e., all of them would benefit (or suffer) in the same manner if a BIP is activated.

Simultaneous Voting Games: Several BIPs could be under consideration at the same point of time. In any target period, a miner who mines a new block has to indicate its preference for all of these BIPs. So in each time period a number of voting games are being simultaneously played. If the outcomes of the BIPs are unrelated, then the effect of simultaneous voting games can be captured by considering the voting games to be played sequentially. While some BIPs can indeed be unrelated, it is unlikely that BIPs under consideration will always be unrelated. The interaction between the outcomes can create complex voting and coalition strategies among the miners. For example, a miner may indicate support for a BIP only if some other miners indicate support for some other BIP.

Repeated Voting Games: Voting for a BIP takes place in at most 26 consecutive target periods. A BIP may not receive adequate support in a particular target period. However, this does not mean that the BIP has failed. It will again be open for voting in the subsequent target period. This process continues until the BIP gets locked-in, or, it times out after the 26 target periods. This feature is again very different from conventional voting game scenarios where once a motion fails, it is not taken up for voting any more.

A miner may mine several blocks in the time period over which voting takes place. We have assumed that the miner indicates its support or opposition to a protocol change proposal in all the blocks in a consistent manner. This seems to be a reasonable assumption. We do not know if there is any situation where a miner in a given time interval may indicate support to a proposal in some of the mined blocks and indicate opposition to the same proposal in the other mined blocks.

3.2 The Attack Game

In this game, the goal of a player or a coalition of players is to get control of the network by ensuring that the total sum of their hash rates is at least 51% of the entire hash rate of the network. So the winning threshold in this game is 51%.

A set of miners may form a coalition whereby they pool their computational resources so that the combined hash rate of the coalition becomes 51%. Such a coalition can attempt to launch a double spending attack on the network and agree to divide the income from the double spending among themselves in accordance with some criterion. It is possible that different coalitions of players can achieve the 51% threshold.

Continuously Playable Game: The Attack Game has the potential of being played at any point of time. There is no fixed time when the game is to be played. If we assume that the players are constantly trying to maximise their profits, then they are potentially exploring coalitions which will increase the hash rate. The aspect of the Attack Game whereby it is always possible to be played is not present in more conventional weighted majority voting games which are played at certain points of time and with adequate notice.

Remark: We have taken 51% as the winning threshold for the Attack Game. It has been suggested that the Bitcoin system can be attacked with even lower threshold [17]. The actual value of the winning threshold is not important for the method of analysis outlined in this work. So even though we later work with only the 51% threshold, a similar analysis can be done with other thresholds.

4 Security Notions for Analysis of the Attack Game

Let $\mathfrak{C}$ be a cryptocurrency system and let G be an Attack Game for $\mathfrak{C}$. For users of $\mathfrak{C}$ a basic question is whether $\mathfrak{C}$ is secure against the 51% attack, or, more formally whether the pair $(\mathfrak{C}, G)$ is secure. The question that arises is how to define security for the pair $(\mathfrak{C}, G)$? Of course, if there is a single miner in G having weight 51% or more of the total weight, then $\mathfrak{C}$ is clearly insecure. A single miner, however, may not have sufficient hash rate to be able to compromise the system. Then one needs to consider a coalition of miners who may wish to attack $\mathfrak{C}$. So any minimal winning coalition in the Attack Game G can mount a successful attack on $\mathfrak{C}$. Consequently, the number of minimal winning coalitions in G provide the number of ways in which $\mathfrak{C}$ can be attacked. It is unlikely

that all possible minimal winning coalitions can actually form. More granular information provides better understanding of the security of $\mathfrak{C}$.

Denoting mw_c to be the number of minimal winning coalitions of size c, we are essentially looking for the distribution (c, mw_c) for $c = 1, \ldots, n$. For example, if $\mathsf{mw}_1 > 0$, then a single miner can win the Attack Game. So one measure of security is the maximum value of c such that $\mathsf{mw}_c = 0$. This would ensure that $(\mathfrak{C}, G)$ is secure against a coalition of c or less number of miners. This leads to the following definition.

Definition 2. *Let $\mathfrak{C}$ be a cryptocurrency system and $G = (N, \mathbf{w}, q)$ be an Attack Game for $\mathfrak{C}$. Then $(\mathfrak{C}, G)$ is said to be $\mathfrak{c}$-secure if $\mathfrak{c} = \max\{c : \mathsf{mw}_c = 0\}$. Equivalently, $(\mathfrak{C}, G)$ is said to be $\mathfrak{c}$-secure if $\mathsf{mw}_c = 0$ for all $c \leq \mathfrak{c}$ and $\mathsf{mw}_{\mathfrak{c}+1} \neq 0$.*

It is perhaps intuitive that $(\mathfrak{C}, G)$ provides the maximum security against the Attack Game if all miners in G have equal weights. We prove a formalisation of this statement in [3]. For a cryptocurrency $\mathfrak{C}$, one may ask for the maximum $\mathfrak{c}$ such that $(\mathfrak{C}, G)$ is $\mathfrak{c}$-secure where the maximum is taken over all possible Attack Games G having n players and the sum of the weights of the players is w_N. Elementary arguments show that when the players have the same weight, the maximum value of $\mathfrak{c}$ is $\lceil nq \rceil - 1$.

It is possible to consider the Attack Game from the viewpoint of a particular player A. Suppose A wishes to win the Attack Game. Then a relevant question for A is the minimum number of other players it needs to form a coalition with. This is captured by considering minimal winning coalitions containing A. More generally, instead of a single miner A, one can consider a coalition S and ask how many other miners are required to win the Attack Game.

For any subset S, denote by $\mathsf{mw}_c(S)$ the number of minimal winning coalitions of cardinality c which contain all elements of S. The distribution $(c, \mathsf{mw}_c(S))$ is of interest. The maximum value of c such that $\mathsf{mw}_c(S) = 0$ is a measure of security of $(\mathfrak{C}, G)$ with respect to the subset S. It indicates the minimum number of other miners that the coalition S will need to collude with to compromise the system. This leads to the following definition.

Definition 3. *Let $\mathfrak{C}$ be a cryptocurrency system and $G = (N, \mathbf{w}, q)$ be an Attack Game for $\mathfrak{C}$. Then $(\mathfrak{C}, G)$ is said to be $\mathfrak{c}$-secure with respect to S if $\mathfrak{c} = \max\{c : \mathsf{mw}_c(S) = 0\}$. Equivalently, $(\mathfrak{C}, G)$ is said to be $\mathfrak{c}$-secure with respect to S if $\mathsf{mw}_c(S) = 0$ for all $c \leq \mathfrak{c}$ and $\mathsf{mw}_{\mathfrak{c}+1}(S) \neq 0$.*

If $S = \{A\}$ is a singleton set consisting of a single player A, then we can talk about $(\mathfrak{C}, G)$ to be $\mathfrak{c}$-secure with respect to the player A. If $(\mathfrak{C}, G)$ is $\mathfrak{c}$-secure, then it is not difficult to argue that $\mathfrak{c} = \min_{A \in N} \max\{c : \mathsf{mw}_c(A) = 0\}$.

So far, we have assumed that all coalitions are possible. In a realistic setting, it is reasonable to postulate that not all coalitions will form. There could be two competing miners who will not be part of any coalition. More generally, one can consider two disjoint coalitions S_1 and S_2 and consider the scenario where the miners in S_1 wish to win the Attack Game but, the miners in S_2 do not wish to compromise $\mathfrak{C}$.

For a positive integer c, we define $\mathsf{mw}(S_1, S_2, c)$ to be the number of minimal winning coalitions in G of cardinalities c containing all elements of S_1 and no element of S_2.

Definition 4. *Let $\mathfrak{C}$ be a cryptocurrency system and $G = (N, \mathbf{w}, q)$ be an Attack Game for $\mathfrak{C}$. Let S_1 and S_2 be two subsets of N. Then $(\mathfrak{C}, G)$ is said to be $\mathfrak{c}$-secure with respect to the pair (S_1, S_2) if $\mathfrak{c} = \max\{c : \mathsf{mw}(S_1, S_2, c) = 0\}$.*

Remarks:

1. For the pair $(\mathfrak{C}, G)$, $\mathsf{mw}(\emptyset, \emptyset, c) = \mathsf{mw}_c$.
2. For any miner A, $\mathsf{mw}(\{A\}, \emptyset, c)$ is the number of minimal winning coalitions in G containing A and having cardinalities equal to c. Consequently, $(\mathfrak{C}, G)$ is $\mathfrak{c}$-secure with respect to A if and only if $(\mathfrak{C}, G)$ is $\mathfrak{c}$-secure with respect to the pair $(\{A\}, \emptyset)$.
3. For any subset S of miners, $\mathsf{mw}(\emptyset, S, c)$ is the number of minimal winning coalitions in G not containing any element of S and having cardinalities equal to c. Consequently, $(\mathfrak{C}, G)$ is $\mathfrak{c}$-secure with respect to the pair $(\emptyset, S)$ if the size of any minimal winning coalition in G not containing any element of S is at least $\mathfrak{c} + 1$. By leaving out a set of miners, we ask for the possibility of the system being compromised by some coalition of the other miners. The maximum value of c such that G is $\mathfrak{c}$-secure with respect to the pair $(\emptyset, S)$ provides a measure of security of the system against coalitions of miners who are not in S.

Definition 5. *Let $\mathfrak{C}$ be a cryptocurrency system and $G = (N, \mathbf{w}, q)$ be an Attack Game for $\mathfrak{C}$. We say that $(\mathfrak{C}, G)$ is $(\mathfrak{c}_1, \mathfrak{c}_2, \mathfrak{c})$-secure, if*

$$\mathfrak{c} = \max\{c : \mathsf{mw}(S_1, S_2, c) = 0 \text{ for all subsets } S_1, S_2 \subseteq N \text{ with } \#S_1 \leq \mathfrak{c}_1,\ \#S_2 \leq \mathfrak{c}_2\}.$$

Remarks:

1. If $S_1 = S_2 = \emptyset$, then there are no constraints and in this case $\mathsf{mw}(S_1, S_2, c) = \mathsf{mw}_c$. $(\mathfrak{C}, G)$ is $(0, 0, \mathfrak{c})$-secure if the size of any minimal winning coalition in G is at least $\mathfrak{c}$.
2. If $S_1 = \{A_i\}$ and $S_2 = \emptyset$, then $\mathsf{mw}(S_1, S_2, c)$ is the number of minimal winning coalitions of cardinality c containing A_i in G.
   $(\mathfrak{C}, G)$ is $(1, 0, \mathfrak{c})$-secure if for any miner A_i, the size of any minimal winning coalition containing A_i is at least $\mathfrak{c}$.
3. $(\mathfrak{C}, G)$ is $(0, 0, \mathfrak{c})$-secure if and only if the cardinality of any minimal winning coalition in G is at least $\mathfrak{c} + 1$. On the other hand, $(\mathfrak{C}, G)$ is $(1, 0, \mathfrak{c})$-secure if and only if the cardinality of any minimal winning coalition in G containing at least one miner is at least $\mathfrak{c} + 1$. Since a minimal winning coalition must contain at least one miner, it follows that G is $(0, 0, \mathfrak{c})$-secure if and only if $(\mathfrak{C}, G)$ is $(1, 0, \mathfrak{c})$-secure.

4. If $S_1 = \{A_i\}$ and $S_2 \neq \emptyset$, then $\mathsf{mw}(S_1, S_2, c)$ is the number of minimal winning coalitions of cardinality c in G containing A_i, but, not containing any element of S_2.
   $(\mathfrak{C}, G)$ is $(1, 1, \mathfrak{c})$-secure if for any two miners A_i and A_j, the size of any minimal winning coalition containing A_i but not containing A_j is at least $\mathfrak{c}$.
5. If $S_1 = \emptyset$ and $S_2 \neq \emptyset$, then $\mathsf{mw}(S_1, S_2, c)$ is the number of minimal winning coalitions of cardinality c in G not containing any element of S_2.
   $(\mathfrak{C}, G)$ is $(0, \mathfrak{c}_2, \mathfrak{c})$-secure if for any set S_2 of size at most $\mathfrak{c}_2$, the size of any minimal winning coalition not containing any element of S_2 is at least $\mathfrak{c}$.

Typically, in a cryptocurrency system the set of miners can be roughly divided into two sets, those having "large" hash rates and those have significantly smaller hash rates. Let L be such a set of "large" miners. Any successful attack is likely to involve the miners in L. On the other hand, it is also quite unlikely that all the miners in L will collude. So one can consider a partition (S_1, S_2) of L where the miners in S_1 are part of the coalition attacking the system while the miners in S_2 are not part of this coalition, i.e., $L = S_1 \cup S_2$ and $S_1 \cap S_2 = \emptyset$. The relevant question is what is the minimum number of miners outside L (i.e., in $N \setminus L$) who need to form a coalition with the miners in S_1 to win the Attack Game?

Let N be a set of miners and L be a subset of N. For any subset S of L, by $\mathsf{mw}_L(S, c)$ we will denote the number of minimal winning coalitions in G containing S which are disjoint from $L \setminus S$ and have cardinalities equal to c.

Definition 6. *Let $\mathfrak{C}$ be a cryptocurrency system and $G = (N, \mathbf{w}, q)$ be an Attack Game for $\mathfrak{C}$. Let L be a subset of N. We say that $(\mathfrak{C}, G)$ is $(L, \mathfrak{c}_1, \mathfrak{c})$-secure if*

$$\mathfrak{c} = \max\{c : \mathsf{mw}_L(S, c) = 0 \text{ for all subsets } S \subseteq L \text{ with } \#S = \mathfrak{c}_1\}.$$

If $(\mathfrak{C}, G)$ is $(L, \mathfrak{c}_1, \mathfrak{c})$-secure, then the following is ensured. Consider any partition of L into S and $L \setminus S$ with $\#S = \mathfrak{c}_1$ and suppose that the coalition S does not collude with any miner in $L \setminus S$. Then to win the Attack Game the coalition S must collude with at least $\mathfrak{c} - \#S$ miners from $N \setminus L$.

Remark: Suppose $\mathfrak{C}$ is a cryptocurrency and G is an Attack Game for $\mathfrak{C}$. Suppose B is any minimal blocking coalition in G. Then any winning coalition for G must contain at least one miner from B. This has practical implications. Suppose that at some point of time $\mathfrak{C}$ is indeed attacked, then it is certain that at least one of the miners in B must have been involved in the attack. Given the pseudonymity of participants in a blockchain network, identifying attackers is a challenge. The above formalisation of the Attack Game could be a useful tool to either reduce the set of suspects or in certain cases even pin-point the set of attackers. We leave the details for future work.

5 A Snapshot Analysis of Bitcoin

For the snapshot analysis, we consider the blocks mined during the period July 2021 to June 2022 as shown in Table 1. For both the Rule Game and the Attack

Game, the players are the miners and as explained in Sect. 3, the weight of a miner is its hash rate estimated from the number of blocks mined by it during this one-year period. We have obtained the data from https://blockchair.com/. Our analysis can be applied equally well if the hash rates are estimated using some other methods, for any other meaningful time period and for any proof-of-work cryptocurrency.

Table 1. Miners (22 known) and the number of blocks they are known to have mined during July 2021 to June 2022.

Miner#	Miner Name	#blocks	Miner#	Miner Name	#blocks
01	Unknown	13808	02	AntPool	8287
03	F2Pool	7454	04	ViaBTC	6078
05	Binance	5645	06	Poolin	4810
07	Foundry USA Pool	3204	08	SlushPool	2955
09	Huobi	331	10	SBICrypto	325
11	EMCD	261	12	Bitdeer	212
13	MaraPool	142	14	OKEX	139
15	BTC.com	113	16	SpiderPool	34
17	Solo CKPool	9	18	50BTC	6
19	SigmaPool	6	20	OKKONG	5
21	mmpool	3	22	BTC.TOP	3
23	KanoPool	1			

The data in Table 1 attributes the highest number of blocks to "Unknown". This means that the identities of the miners of these blocks are not known. It is most likely that it is not a single entity which mined these blocks. So in the computation of the voting powers, it is not appropriate to consider "Unknown" as a player. Let U be the set of all miners in the group "Unknown". We handle the miners in U in the following manner.

Suppose the total weight of the components $\mathbf{w}$ is w_N out of which the miners in U have a total weight of w. Suppose that a fraction p of the total weight of the miners in "Unknown" play to win the game while the other $(1-p)$ fraction of the total weight of the miners in "Unknown" try to block the winning. By considering different values of p in $[0, 1]$, it becomes possible to study the effect of the "Unknown" miners on the game. To capture this idea we make the following definition.

Definition 7. *Given the game $G = (N, \mathbf{w}, q)$, a player U with weight w and $p \in [0, 1]$, we define the game $G^{(p)}$ with respect to U as $G^{(p)} = (N \backslash \{U\}, \mathbf{w}_{\overline{U}}, q^{(p)})$ where $q^{(p)} = (q \cdot w_N - p \cdot w)/(w_N - w)$ and w_N is the sum of the weights of all the players in the original game G. Here $\mathbf{w}_{\overline{U}}$ denotes the weight vector obtained from $\mathbf{w}$ by leaving out the entry corresponding to U.*

The miners in "Unknown" are not present in $G^{(p)}$ so the total weight of the miners in $G^{(p)}$ is $w_N - w$. To win, a coalition in the original game G needed to

have weight at least $q \cdot w_N$. So in $G^{(p)}$, to win a coalition needs to have weight at least $q \cdot w_N - p \cdot w$.

In the game $G = (N, \mathbf{w}, q)$ obtained from Table 1, there are a total of $n = 23$ miners with the weight vector $\mathbf{w}$ as given in Table 1 and $q = 0.95$. The value of w_N is 53831 and "Unknown" miners have total weight of $w = 13808$. In the game $G^{(p)}$, the group U is removed from the game while the threshold $q^{(p)}$ is modified depending upon the value of p.

5.1 Computation of Voting Powers in the Rule Game

There is a large literature suggesting a variety of indices on how to measure the power or influence of a player in a voting game. We refer to [18] for discussions on this vast subject and to [11] for a textbook level introduction. We consider the Coleman preventive power measure which has been defined in [14]. Under this measure, the power of a player A_i in a game G is defined as follows.

$$\mathsf{CP}_G(A_i) = \frac{m_G(A_i)}{\#W(G)}.$$

The value of $\mathsf{CP}_G(A_i)$ is at least 0 and at most 1. We get $\mathsf{CP}_G(A_i) = 0$ if and only if A_i is not a swing in any coalition (a dummy player with $m_G(A_i) = 0$). We have $\mathsf{CP}_G(A_i) = 1$ if and only if A_i is present in every winning coalition (a blocker with $m_G(A_i) = \#W(G)$). Further, $\mathsf{CP}_G(A_i)$ is monotonically non-decreasing with the weight of A_i. For the Rule Game, the property of a miner being a blocker is of crucial interest. To the best of our knowledge, among the various power measures available in the literature, CP is the only power measure which assigns the maximum value of 1 to a blocker and is monotonically non-decreasing with the weights of the players. Due to these two reasons, we suggest that CP is an appropriate measure for measuring the power of a player in the Rule Game.

Let $G = (N, \mathbf{w}, q)$ where the sum of the weights of all the players in G is w_N. Let U be a player with weight w. For $p \in [0, 1]$ consider the game $G^{(p)}$ with respect to U. A player A_i in $G^{(p)}$ of weight w_i is a blocker if and only if $w_i > (1 - q)w_N + (p - 1)w$. So whether a player is a blocker depends on the value of p. It may happen that for a certain value of p, the player is a blocker, but, fails to be a blocker for a different value of p. For a specified value of p, the set of blockers in $G^{(p)}$ is fixed.

We consider the game $G^{(p)}$ for various values of p. The power profile given by CP for $G^{(p)}$ for various values of p is shown in Table 2. In Table 2 a value of 1 in the (i, p) cell indicates that player number i is a blocker in $G^{(p)}$. For $p = 0$, when none of the "Unknown" miners support the protocol change, the largest 7 miners are blockers (the protocol cannot be changed without their support) while the remaining 15 miners are all dummies with no say in the matter. As p increases, the number of blockers decreases. In Table 2, the numbers of blockers are 7, 5, 5, 2, and 1 corresponding to the values of $p = 0, 0.1, 0.2, 0.3$ and 0.4. For the other values of p, none of the known players are blockers. As the fraction of

miners in "Unknown" who support the protocol change increases, the blocking capability of the other players go down. More generally, in Table 2, with increase in p, the power of any particular player decreases monotonically.

Table 2. Values of the Coleman preventive power index of the different players in the Rule Game for July 2021 to June 2022 and for various values of p.

Player#	Player	wt	p										
			0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
02	AntPool	8287	1	1	1	1	1	0.941	0.864	0.876	0.759	0.731	0.641
03	F2Pool	7454	1	1	1	1	0.998	0.799	0.864	0.711	0.719	0.603	0.621
04	ViaBTC	6078	1	1	1	0.996	0.716	0.797	0.635	0.625	0.560	0.548	0.409
05	Binance	5645	1	1	1	0.913	0.713	0.783	0.593	0.580	0.527	0.457	0.408
06	Poolin	4810	1	1	1	0.536	0.713	0.494	0.591	0.488	0.478	0.383	0.364
07	Foundry USA Pool	3204	1	0.959	0.335	0.518	0.435	0.465	0.354	0.338	0.242	0.282	0.244
08	SlushPool	2955	1	0.780	0.332	0.518	0.435	0.405	0.328	0.308	0.241	0.252	0.222
09	Huobi	331	0	0.117	0.001	0.044	0.007	0.061	0.018	0.049	0.023	0.042	0.017
10	SBICrypto	325	0	0.117	0.001	0.043	0.007	0.060	0.018	0.048	0.023	0.042	0.016
11	EMCD	261	0	0.094	0.001	0.038	0.007	0.047	0.014	0.037	0.017	0.032	0.013
12	Bitdeer	212	0	0.076	0.001	0.031	0.007	0.039	0.012	0.030	0.015	0.026	0.011
13	MaraPool	142	0	0.057	0.001	0.022	0.005	0.026	0.007	0.020	0.009	0.017	0.007
14	OKEX	139	0	0.056	0.001	0.022	0.005	0.025	0.007	0.019	0.009	0.017	0.007
15	BTC.com	113	0	0.045	0.001	0.017	0.004	0.020	0.006	0.016	0.008	0.014	0.006
16	SpiderPool	34	0	0.017	0.001	0.008	0.001	0.007	0.001	0.005	0.002	0.004	0.001
17	Solo CKPool	9	0	0.002	0	0.001	0	0.002	0.001	0.001	0.001	0.001	0.001
18	50BTC	6	0	0.001	0	0.001	0	0.001	0	0.001	0.001	0.001	0
19	SigmaPool	6	0	0.001	0	0.001	0	0.001	0	0.001	0.001	0.001	0
20	OKKONG	5	0	0.001	0	0.001	0	0.001	0	0.001	0	0.001	0
21	mmpool	3	0	0.001	0	0	0	0.001	0	0	0	0	0
22	BTC.TOP	3	0	0.001	0	0	0	0.001	0	0	0	0	0
23	KanoPool	1	0	0	0	0	0	0	0	0	0	0	0
#blockers			7	5	5	2	1	0	0	0	0	0	0

5.2 Computation of Security in the Attack Game

As in the Rule Game, the role of the miners in the group marked "Unknown" is tackled by considering the game $G^{(p)}$ for various values of p. This indicates that a fraction p of the total weight of the miners in "Unknown" are trying to attack the system while a fraction $1-p$ of the total weight of the miners in "Unknown" do not form part of any such attack coalition.

The cardinality wise number of minimal winning coalitions in $G^{(p)}$ for different values of p are shown in Table 3. The value of 0 means that there is no minimal winning coalition for the particular values of $\mathfrak{c}$ and p. There is, however, a nuance in the interpretation of this condition. For $\mathfrak{c} \leq 3$, the value 0 denotes that there is actually no winning coalition in the game while for $\mathfrak{c} \geq 16$, the value 0 denotes that the winning coalitions are not minimal, i.e., dropping any miner from the coalition does not convert it into a losing coalition. We have the following observations from Table 3.

1. There is no winning coalition of cardinality 1 of known miners. So a coalition of at least 2 known miners along with more than 80% of the unknown miners' weight is required to win the Attack Game.
2. If 50% of the weight of the "Unknown" miners can be roped in then there are 3 minimal winning coalitions of the other 22 miners of cardinality 3.
3. There are several minimal winning coalitions of cardinalities 4 (or more) that do not require any unknown miner to win the Attack Game. So the system is vulnerable if 4 miners collude. In fact, there are 22 different ways of forming a set of 7 miners (without the "Unknown" miners) which can compromise the system. Given that the Bitcoin market is worth billions of dollars, the thought that there are multiple ways to form a malicious coalition of only 7 miners (all mining pools) is disconcerting.
4. In general, as p increases, the number of minimal coalitions initially increases and then decreases. The increase indicates that the number of winning coalitions itself goes up while the decrease indicates that some of the winning coalitions fail to remain minimal.

Table 3. Cardinality wise number of minimal winning coalitions in $G^{(p)}$ for the time period July 2021 to June 2022 of Table 1.

$\mathfrak{c}$	p										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1	3
3	0	0	0	0	0	3	6	11	18	19	24
4	1	3	6	11	25	15	17	12	20	10	25
5	8	8	15	13	17	13	26	17	53	44	48
6	3	0	23	19	47	30	38	47	44	29	36
7	22	0	18	5	45	15	59	53	31	58	33
8	41	17	50	50	44	37	48	50	64	65	7
9	63	64	40	61	55	63	43	80	50	88	18
10	76	72	74	101	107	110	77	102	43	99	17
11	85	97	120	108	146	126	63	97	35	65	4
12	95	84	94	108	117	96	70	58	26	14	1
13	73	51	40	48	45	88	41	45	9	13	1
14	26	64	3	31	22	52	11	31	1	18	7
15	14	34	5	36	8	10	9	5	1	5	1
16	4	20	1	9	0	9	1	0	0	1	0
17	1	8	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0
Total	512	522	489	600	678	667	509	608	395	529	225

In Table 4, we provide the cardinality wise number of minimal winning coalitions containing the largest miner AntPool. It is possible to compute similar data for all the individual players and their subsets. From the totals of the first columns of Tables 3 and 4 we see that there are $512-362 = 150$ minimal winning coalitions of miners other than the largest miner AntPool and the "Unknown" miners in the Attack Game. Table 4 shows that if 50% of the "Unknown" miners can be roped in, then AntPool can form possible coalitions consisting of itself and just 2 of the other 22 miners to win the Attack Game. On the other hand, if coalitions of size 4 or more are considered, then AntPool can form several winning coalitions in the Attack Game without involving any of the miners in "Unknown". Again, this is not a very comfortable scenario. Note that for $p = 0$, $\mathsf{mw}_5(\{Antpool\}) = 0$ although $\mathsf{mw}_4(\{Antpool\}) > 0$. This means that the winning coalitions of cardinality 5 are *not minimal*.

$(L, \mathfrak{c}_1, \mathfrak{c})$-Security: For a set L of large miners, we consider $(L, \mathfrak{c}_1, \mathfrak{c})$-security in $G^{(p)}$ for different values of p. We have considered several options for L, namely, L consists of the miners with i of the largest weights where we have taken $i = 1, 2, 3, 4, 5$ and 6. The value of $\mathfrak{c}_1$ is in the set $\{0, 1, \ldots, i\}$. In each case, we have computed the corresponding value of $\mathfrak{c}$. Table 5 provides values of $\mathfrak{d}$ such that $G^{(p)}$ is $(L, \mathfrak{c}_1, \mathfrak{c}_1 + \mathfrak{d} - 1)$-secure for different values of p and $\mathfrak{c}_1$. This means that $\mathfrak{c}_1$ largest miners in L need to collude with at least $\mathfrak{d}$ miners outside of L to win the Attack Game. In the table, a '–' denotes that there is no winning coalition for the corresponding condition whereas a '$*$' denotes that any coalition of size $\mathfrak{c}_1$ of L is already a winning coalition in $G^{(p)}$. Based on Table 5, we make the following observations.

Table 4. Cardinality wise number of minimal winning coalitions containing the largest miner AntPool for Table 1 in $G^{(p)}$.

$\mathfrak{c}$	p										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	1	3
2	0	0	0	0	0	3	5	8	12	9	3
3	1	3	6	10	22	8	9	6	8	0	3
4	8	6	12	10	14	12	26	17	35	25	20
5	0	0	19	19	32	30	36	47	34	27	15
6	2	0	2	4	22	11	33	43	20	35	25
7	26	10	16	21	2	3	15	21	30	44	2
8	35	36	17	28	11	17	9	18	31	53	0
9	41	46	16	40	18	34	21	22	18	64	8
10	57	64	25	43	7	26	4	19	7	33	3
11	75	49	25	43	2	11	0	18	3	4	1
12	72	24	18	24	0	17	1	10	0	6	0

(*continued*)

Table 4. (*continued*)

$\mathfrak{c}$	p										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
13	26	45	1	17	9	12	0	0	0	1	0
14	14	29	5	21	8	2	0	0	0	0	0
15	4	13	1	9	0	4	0	0	0	0	0
16	1	2	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0
Total	362	327	163	289	147	190	159	229	198	302	83

1. Case $\#L = 4$ and $\mathfrak{c}_1 = 0$. All corresponding entries in the table are '–'. This means that if the largest four miners are left out, then there is no way to win the Attack Game. In other words, the set of four largest miners form a minimal blocking coalition. So any attack on the system certainly involves one of the four largest miners. As mentioned earlier, this fact contains useful information. If an attack is detected in the future, then one can be sure that at least one of the four largest miners was certainly part of the attack.
2. Case $\#L = 5$ and $\mathfrak{c}_1 \geq 5$. All corresponding entries in the table are marked by '$*$'. Similarly, for $\#L = 6$. This means that if five (or more) of the largest miners collude, then the Attack Game is immediately won.
3. Case $\#L = 3$ and $\mathfrak{c}_1 = 0$, i.e., the three largest miners are left out. The entries for $p \leq 0.6$ are '–'. This means that if less than 60% of the hash rate of "Unknown" miners are involved in the attack, then the attack cannot be successful. On the other hand, the entry for $p = 0.7$ is 9. This means that if 70% of the hash rate of the "Unknown" miners are involved in the attack, then leaving out the three largest miners, a coalition of 9 of the other $23 - 1 - 3 = 19$ miners is both necessary and sufficient to win the Attack Game.
4. Case $\#L = 4$ and $\mathfrak{c}_1 = 3$ and $p = 0.4$. The corresponding entry in the table is 1. The condition $\#L = 4$ and $\mathfrak{c}_1 = 3$ means that out of the four largest miners, one is left out. If 40% of the miners in "Unknown" can be roped in, then out of the 19 other miners, it is necessary and sufficient to have only 1 miner to win the Attack Game.
5. Case $\#L = 5$, $\mathfrak{c}_1 = 1$ and $p = 0.9$. The corresponding entry in the table is 4. The condition $\#L = 5$ and $\mathfrak{c}_1 = 1$ means that out of the five largest miners, four are left out. If 90% of the miners in "Unknown" can be roped in, then out of the 7 other miners, it is necessary and sufficient to have only 4 miners to win the Attack Game.

6. Consider the cases ($\#L = 5$, $\mathfrak{c}_1 = 3$, $p = 0$) and ($\#L = 6$, $\mathfrak{c}_1 = 3$, $p = 0$). The corresponding entries in the table are 2 and '–'. This may appear to be surprising, since in both cases $\mathfrak{c}_1 = 3$. The explanation is that in the first case, out of the five largest miners, two are left out, while in the second case, out of the six largest miners, three are left out. Since in the second case, more miners are left out, that leads to the absence of any (minimal) winning coalition.

Table 5. The entries in the table are $\mathfrak{d}$ such that $G^{(p)}$ is $(L, \mathfrak{c}_1, \mathfrak{c}_1+\mathfrak{d}-1)$-secure where L consists of the miners with the i largest weights as given in Table 1 for $i = 1, 2, 3, 4, 5, 6$.

		p										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\#L = 1$	$\mathfrak{c}_1 = 0$	6	5	5	4	4	4	3	3	3	3	3
	$\mathfrak{c}_1 = 1$	3	3	3	3	3	2	2	2	2	1	1
$\#L = 2$	$\mathfrak{c}_1 = 0$	-	-	-	7	5	5	4	4	3	3	3
	$\mathfrak{c}_1 = 1$	4	4	3	3	3	3	2	2	2	2	1
	$\mathfrak{c}_1 = 2$	2	2	2	2	2	1	1	1	1	$*$	$*$
$\#L = 3$	$\mathfrak{c}_1 = 0$	-	-	-	-	-	-	-	9	4	4	3
	$\mathfrak{c}_1 = 1$	-	9	4	4	3	3	3	2	2	2	1
	$\mathfrak{c}_1 = 2$	3	2	2	2	2	1	1	1	1	$*$	$*$
	$\mathfrak{c}_1 = 3$	1	1	1	1	1	$*$	$*$	$*$	$*$	$*$	$*$
$\#L = 4$	$\mathfrak{c}_1 = 0$	-	-	-	-	-	-	-	-	-	-	-
	$\mathfrak{c}_1 = 1$	-	-	-	-	-	9	3	3	3	2	2
	$\mathfrak{c}_1 = 2$	6	3	3	2	2	1	1	1	1	$*$	$*$
	$\mathfrak{c}_1 = 3$	2	1	1	1	1	$*$	$*$	$*$	$*$	$*$	$*$
	$\mathfrak{c}_1 = 4$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
$\#L = 5$	$\mathfrak{c}_1 = 0$	-	-	-	-	-	-	-	-	-	-	-
	$\mathfrak{c}_1 = 1$	-	-	-	-	-	-	-	-	-	4	2
	$\mathfrak{c}_1 = 2$	-	-	-	9	3	2	2	1	1	$*$	$*$
	$\mathfrak{c}_1 = 3$	2	2	1	1	1	$*$	$*$	$*$	$*$	$*$	$*$
	$\mathfrak{c}_1 = 4$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
	$\mathfrak{c}_1 = 5$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
$\#L = 6$	$\mathfrak{c}_1 = 0$	-	-	-	-	-	-	-	-	-	-	-
	$\mathfrak{c}_1 = 1$	-	-	-	-	-	-	-	-	-	-	-
	$\mathfrak{c}_1 = 2$	-	-	-	-	-	-	3	1	1	$*$	$*$
	$\mathfrak{c}_1 = 3$	-	7	1	1	1	$*$	$*$	$*$	$*$	$*$	$*$
	$\mathfrak{c}_1 = 4$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
	$\mathfrak{c}_1 = 5$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
	$\mathfrak{c}_1 = 6$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$

6 Conclusion

Protocol stability and security play extremely important roles in the socio-economic dynamics of any proof-of-work cryptocurrency. In this work, we have modelled these two key functional aspects using weighted majority voting games. Our modelling immediately allows the rich tools from the theory of voting games to be used for analysis of cryptocurrency systems. As a practical contribution, we have shown how to perform concrete snapshot analysis on the games using such tools. We suggest that such analysis be performed at regular intervals to build a good understanding of the socio-economic dynamics of proof-of-work cryptocurrencies like Bitcoin. Wide dissemination of the results of such periodic analysis will help the general public to understand and appreciate the risks involved in using and investing in cryptocurrencies. This will also place the usually small number of parties who can compromise the system under intense public scrutiny and hopefully prevent any malicious behaviour. We also hope that this work will stimulate interest in the connection between cryptocurrencies and voting games leading to further interesting work on the intersection of these two topics.

Acknowledgements. We would like to thank the anonymous reviewers for their comments and suggestions. Majority of this work was done while Sanjay Bhattacherjee was visiting the Turing Laboratory, Applied Statistics Unit, Indian Statistical Institute.

References

1. 51% Attacks (Digital Currency Initiative, MIT Media Lab). https://dci.mit.edu/51-attacks. Accessed 8 Sept 2022
2. Banzhaf, J.F.: Weighted voting doesn't work: a mathematical analysis. Rutgers Law Rev. **19**, 317–343 (1965)
3. Bhattacherjee, S., Sarkar, P.: Cryptocurrency voting games. Cryptology ePrint Archive, Paper 2017/1167 (2017). https://eprint.iacr.org/2017/1167
4. Bhattacherjee, S., Sarkar, P.: Weighted voting procedure having a unique blocker. Int. J. Game Theory **50**(1), 279–295 (2021)
5. BIP 141: Segregated Witness (Consensus layer). https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki. Accessed 8 Sept 2022
6. BIP 91: Reduced threshold Segwit MASF. https://github.com/bitcoin/bips/blob/master/bip-0091.mediawiki. Accessed 8 Sept 2022
7. BIP Github page. https://github.com/bitcoin/bips. Accessed 8 Sept 2022
8. Bitcoin Magazine article on a 51% attack on Bitcoin SV (2021). https://bitcoinmagazine.com/markets/bitcoin-sv-sees-51-attack. Accessed 8 Sept 2022
9. Bitcoin.com article on a 51% attack on Bitcoin Gold (2020). https://news.bitcoin.com/bitcoin-gold-51-attacked-network-loses-70000-in-double-spends/. Accessed 8 Sept 2022
10. Bitcoin.com article on a 51% attack on Verge (2021). https://news.bitcoin.com/privacy-coin-verge-third-51-attack-200-days-xvg-transactions-erased/. Accessed 8 Sept 2022
11. Chakravarty, S.R., Mitra, M., Sarkar, P.: A Course on Cooperative Game Theory. Cambridge University Press, Cambridge (2015)

12. Chakravarty, S., Sarkar, P.: An Introduction to Algorithmic Finance, Algorithmic Trading and Blockchain. Emerald Group Publishing, Bingley (2020)
13. CoinDesk article on a 51% attack on Ethereum Classic (2020). https://www.coindesk.com/markets/2020/08/29/ethereum-classic-hit-by-third-51-attack-in-a-month/. Accessed 8 Sept 2022
14. Coleman, J.S.: Control of collectives and the power of a collectivity to act. In: Lieberma, B. (ed.) Social Choice, pp. 269–298. Gordon and Breach, New York (1971)
15. Deegan, J., Packel, E.W.: A new index of power for simple n-person games. Int. J. Game Theory **7**(2), 113–123 (1978)
16. EIP Github page. https://github.com/ethereum/EIPs. Accessed 8 Sept 2022
17. Eyal, I., Sirer, E.G.: Majority is not enough: bitcoin mining is vulnerable. In: Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, 3–7 March 2014, pp. 436–454 (2014)
18. Felsenthal, D.S., Machover, M.: The Measurement of Voting Power. Edward Elgar, Cheltenham (1998)
19. Fisch, B., Pass, R., Shelat, A.: Socially optimal mining pools. In: Devanur, N.R., Lu, P. (eds.) WINE 2017. LNCS, vol. 10660, pp. 205–218. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71924-5_15
20. Holler, M.J.: Forming coalitions and measuring voting power. Political Stud. **30**(2), 262–271 (1982)
21. Holler, M.J., Packel, E.W.: Power, luck and the right index. J. Econ. **43**(1), 21–29 (1983)
22. Kiayias, A., Koutsoupias, E., Kyropoulou, M., Tselekounis, Y.: Blockchain mining games. In: Proceedings of the 2016 ACM Conference on Economics and Computation, EC 2016, Maastricht, The Netherlands, 24–28 July 2016, pp. 365–382 (2016)
23. Kroll, J., Davey, I., Felten, E.W.: The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. In: Workshop on the Economics of Information Security (2013)
24. Kurz, S., Maaser, N., Napel, S., Weber, M.: Mostly sunny: a forecast of tommorow's power index research. Homo Oecon. **32**, 133–146 (2015)
25. Lewenberg, Y., Bachrach, Y., Sompolinsky, Y., Zohar, A., Rosenschein, J.S.: Bitcoin mining pools: a cooperative game theoretic analysis. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, 4–8 May 2015, pp. 919–927 (2015)
26. LIP Github page. https://github.com/litecoin-project/lips. Accessed 8 Sept 2022
27. Livebitcoinnews.com article on a 51% attack on Litecoin Cash (2018). https://www.livebitcoinnews.com/litecoin-cash-51-attack-highlights-insecurity-of-smaller-pow-coins/. Accessed 8 Sept 2022
28. Medium.com article on a 51% attack on Vertcoin (2018). https://medium.com/coinmonks/vertcoin-vtc-is-currently-being-51-attacked-53ab633c08a4. Accessed 8 Sept 2022
29. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
30. Namecoin (IFA) Proposals Github page. https://github.com/namecoin/proposals. Accessed 8 Sept 2022
31. Napel, S.: Voting power. In: Congleton, R., Grofman, B., Voigt, S. (eds.) Oxford Handbook of Public Choice. Oxford University Press, Oxford (2016)
32. Narayanan, A., Bonneau, J., Felten, E.W., Miller, A., Goldfeder, S.: Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press, Princeton (2016)

33. Penrose, L.S.: The elementary statistics of majority voting. J. Roy. Stat. Soc. **109**(1), 53–57 (1946)
34. Shapley, L.S.: A value for n-person games. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games II (Annals of Mathematics Studies), pp. 307–317. Princeton University Press, Princeton (1953)
35. Shapley, L.S., Shubik, M.J.: A method for evaluating the distribution of power in a committee system. Am. Polit. Sci. Rev. **48**, 787–792 (1954)
36. The Guardian article on a 51% attack on Bitcoin (2014). https://www.theguardian.com/technology/2014/jun/16/bitcoin-currency-destroyed-51-attack-ghash-io. Accessed 8 Sept 2022
37. Why Ethereum Classic? (2022). https://ethereumclassic.org/why-classic. Accessed 8 Sept 2022