Cryptocurrency Voting Games

Sanjay Bhattacherjee and Palash Sarkar
Applied Statistics Unit
Indian Statistical Institute
203, B.T. Road
Kolkata
India 700108
email:{sanjay.bhattacherjee@gmail.com, palash@isical.ac.in}

November 30, 2017

Abstract

This work shows that weighted majority voting games occur in cryptocurrencies. In particular, two such games are highlighted. The first game, which we call the Rule Game, pertains to the scenario where the entities in the system engage in a voting procedure to accept or reject a change of rules. The second game, which we call the Attack Game, refers to the scenario where a group of entities in a cryptocurrency system can form a coalition to engage in double spending. For the Rule Game we provide analysis to argue that the Coleman's preventive power measure is the appropriate tool for measuring a player's influence in the game while for the Attack Game, we define a notion of stability based on the notion of minimal winning coalitions. For both the Rule Game and the Attack Game, we show how to analyse the games based on a snapshot of real world data for Bitcoin which is presently the most popular of all the cryptocurrencies.

1 Introduction

A cryptocurrency is a form of digital currency which is powered by tools from modern cryptography. The first and to date the most successful cryptocurrency called Bitcoin was proposed by the eponymous Satoshi Nakamoto (Nakamoto, 2008). Since then many cryptocurrencies have been proposed[1]. Bitcoin, though remains the most popular and valuable cryptocurrency. In 2017, the exchange value of 1 Bitcoin rose from about 1000 US Dollars to about 10000 US Dollars[2].

The most remarkable feature of Bitcoin and other similar cryptocurrencies is that it is not backed by any central authority. This makes such currencies stand apart from all fiat currencies that have been proposed till date. It is perhaps due to this unique feature that cryptocurrencies are sometimes said to be a disruptive technology with the potential to revolutionise the socio-economic framework of the future.

From an academic point of view, the success and continuously growing popularity of cryptocurrencies make them a very interesting object of research. There are multiple dimensions to such research. Apart from the questions arising from the underlying cryptography, there are many questions which arise from the standpoint of economics. In this work, we will look at two such questions arising from the viewpoint of cooperative game theory, or more particularly voting games.

[1] `https://en.wikipedia.org/wiki/List_of_cryptocurrencies`
[2] `https://en.wikipedia.org/wiki/History_of_bitcoin#2017`

Voting Games

Formation of coalitions of people or entities with the intention of achieving some common goal occur in various socio-economic settings. A typical example is the process of decision making in committees. For a resolution to be passed, it must receive a designated amount of support. A coalition of members of the committee can provide the necessary support for the resolution to be adopted, or, it may also block the resolution from being adopted. Such decision making procedures have been studied in the literature as voting games. At an abstract level, a voting game consists of a set of players and the game is given by a function which assigns a value of 0 or 1 to any coalition of players, where 0 denotes that the coalition is losing while 1 denotes that the coalition is winning.

Of practical interest are weighted majority voting games. In such games, each player has a pre-assigned weight. A coalition 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 sum total of the weights of all the players. A typical example of a weighted majority voting game is a company boardroom where the weights represent the amount of shares that board members own. In the domain of public policy, two well known examples of weighted majority voting games are those arising in the decision making procedures of the International Monetory fund and the European Union.

There is an extensive literature on voting games in general and also weighted majority voting games in particular. One of the goals of the theory is to be able to capture the power or influence that a player has in the game. Two central concepts that arise are the notion of a swing and that of a minimal winning coalition. A player is a swing in a coalition if the coalition is winning, it contains the player and becomes losing if the player is dropped from the coalition. The number of coalitions in which a player is a swing can be taken as a measure of the power of the player in the game. A minimal winning coalition is a winning coalition such that if any player is dropped from the coalition, then it becomes a losing coalition. In other words, every player in a minimal winning coalition is a swing player. The notion of minimal winning coalition has also been used to provide alternative ways of capturing the notion of power of a player.

(Shapley, 1953; Shapley and Shubik, 1954) introduced the idea of measuring power in a voting game. (Banzhaf, 1965) suggested the use of swings to measure voting power of players. Later work (Coleman, 1971) 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 (Holler, 1982; Holler and Packel, 1983; Deegan and Packel, 1978). (Felsenthal and Machover, 1998) provides an extensive overview of voting games and an introduction to the topic can be found in (Chakravarty et al., 2015).

Voting Games in Cryptocurrencies

In most cryptocurrencies, generation of new currency requires computational power. The process of creating new currency is called mining. There are commercial entities called miners who are engaged in the task of mining. For Bitcoin, there is a sophisticated ecosystem for mining new coins[3]. At any point of time, there is a total amount of computational power that is engaged in the process of mining. This is usually specified in terms of the so-called "hash rate" which we explain later. Different miners have different proportions of the total hash rate of the entire system. For certain aspects of a cryptocurrency, the say of a miner is proportional to the fraction of the total hash rate that it possesses. This feature indicates the presence of an implicit voting game in cryptocurrencies.

In this work, we show how weighted majority voting games arise in cryptocurrencies. The players in the game are the miners and the weight of a miner is proportional to the fraction of the total hash

[3] https://en.bitcoin.it/wiki/Mining

rate that it controls. To completely specify the game, it is required to specify the winning threshold. The value of this threshold depends upon the particular situation. We identify two such scenarios.

The rules governing cryptocurrencies are described in terms of protocols which are initially specified at the time of inception. At later points of time, these rules (or protocols) can be modified. There is a well defined modification procedure. In the context of Bitcoin, this is called a Bitcoin Improvement Protocol (BIP). We show that a BIP can be viewed as a weighted majority voting game among the miners where the winning threshold q is 0.95. More generally, we use the term Rule Game to denote the voting games arising in the context of change of rule of any cryptocurrency. The purpose of having a high winning threshold in a Rule Game is to achieve near unanimity for a change of rule to take place. From the viewpoint of voting games, having a high winning threshold results in several miners becoming blockers. We discuss in details the appropriate measure of voting power that is applicable in this context. Our analysis reveal that the Coleman preventive power measure (Coleman, 1971) and the Holler public good measure (Holler, 1982) satisfy some basic intuition for measuring the influence of a miner. Further, we prove that the Coleman preventive power measure increases monotonically with the weights while the Holler public good index does not necessarily do so. It is perhaps an intuitive requirement that the powers of voters increase monotonically with their weights. Based on this intuition, we propose the use of Coleman preventive power measure as the appropriate tool for measuring a miner's influence in a Rule Game.

Cryptographic considerations show that 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 double spending of the currency. So, it is of interest to consider games where the winning threshold q is set to 0.51. We call such a game to be an Attack Game. From the viewpoint of voting games, several questions can be formulated to analyse the Attack Game. These are based upon 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 formulate several such questions and finally come up with a notion of stability of a cryptocurrency against the Attack Game.

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. Performing similar snapshot analysis at regular time intervals will provide a good understanding of the socio-economic dynamics of Bitcoin. More generally, such analysis can also be performed for other cryptocurrencies.

Related works: There is already a fairly large and growing literature on cryptocurrencies[4]. Some game theoretic aspects of cryptocurrencies have already been studied (Kiayias et al., 2016; Lewenberg et al., 2015; Fisch et al., 2017; Kroll et al., 2013). To the best of our knowledge, the application of weighted majority voting games to cryptocurrencies has not been considered earlier.

[4]https://github.com/decrypto-org/blockchain-papers

2 An Overview of Cryptocurrencies

In this section we provide a high level overview of cryptocurrencies. The purpose of this overview is to provide a sufficient background for understanding how voting games arise in cryptocurrencies. For further details, the reader may consult Narayanan et al. (2016).

2.1 Cryptographic Tools

Modern cryptography has many tools to enable various tasks in the present digital era. Two such cryptographic components are used in implementing cryptocurrencies. A brief overview of these two primitives are provided below.

Hash functions: A hash function maps a domain $\mathcal{D}$ to a range $\mathcal{R}$ where $\mathcal{D}$ is the set of all binary strings having some maximum possible length and $\mathcal{R}$ is the set of all binary strings of some fixed length. For example, $\mathcal{D}$ could be the set of all strings of length less than 2^{64} whereas $\mathcal{R}$ could be the set of all strings of length equal to 256 bits. In this case, the cardinality of $\mathcal{D}$ is $2^{2^{64}-1} - 1$ and the cardinality of $\mathcal{R}$ is 2^{256}. So, a hash function maps a very large domain to a comparatively smaller range. In absolute terms, however, the range itself is quite large. Since the domain of a hash function is larger than its range, there will be two (or more) distinct strings in the domain which are mapped to the same string in the range. Such a pair of strings is called a collision. For cryptocurrency applications, there are three requirements on a hash function.

Efficient to evaluate: Given a string in the domain, it should be possible to compute the output of the hash function very fast. The speed of evaluation of the hash function can often be improved by using very efficient software implementation or customized hardware implementation.

Collision resistance: As mentioned above, a hash function will necessarily have collisions. The requirement of collision resistance is met by a hash function if it is *computationally difficult* to actually find a collision.

One-way: Given a string in the range, it should be *computationally difficult* to find a pre-image for it. In other words, while the hash function should be very efficient to evaluate in the forward direction, it should be computationally difficult to invert the function.

Examples of practical hash functions which are efficient to evaluate and which are considered to be collision resistant and one-way are SHA-256, RIPEMD160 and SHA-3 among others. The first two are used in Bitcoin.

Digital signature scheme: This is the counterpart of usual handwritten signatures and have many applications in enabling e-commerce. While a digital signature is not a complete analogue of hand-written signatures, certain facets of hand-written signatures are indeed captured by digital signatures. The two main aspects of a digital signature scheme are secret signing and public verifiability. Secret signing means that signing of a digital message can be done only by the relevant person while public verification means that the process of verfiying the signature on the message can be done by anybody.

In a little more details, a digital signature scheme consists of three algorithms, namely, a key generation algorithm, a signing algorithm and a verification algorithm. A person (or an entity) invokes the key generation algorithm to generate a key pair $(\mathsf{sk}, \mathsf{pk})$, where sk is the signing key and is kept secret by the concerned entity; pk is the public (or, verification) key which is published and is used to verify the signature. Given a message M, the signer uses the signing algorithm with its signing key sk

to produce a signature σ on the message. Given a message-signature pair (M, σ), anybody can run the verification algorithm with the corresponding verification key pk to verify the validity of the signature on the message.

There is a large literature on digital signature algorithms addressing several aspects of its security and efficiency. Here we only mention the well known elliptic curve digital signature algorithm (ECDSA) which is used in the implementation of Bitcoin.

2.2 Cryptocurrency Basics

As mentioned above, cryptocurrencies are built using a hash function $\mathcal{H}$ and a digital signature algorithm. Whichever entity wishes to participate in the cryptocurrency uses the digital signature algorithm to first generate a signing-verification key pair. In fact, a single entity can generate several such key pairs. The cryptocurrency setup only recognises the key pairs and is not concerned about who created these.

A cryptocurrency is designated in a particular unit. Different cryptocurrencies have different units with different names. Examples are Bitcoin and Ether. The units are divisible and the extent to which a unit is divisible is governed by the rules of the particular cryptocurrency. Most present day cryptocurrencies are essentially based on Bitcoin[5]. Accordingly, our description below is primarily based on Bitcoin though the ideas are general enough and apply to other cryptocurrencies as well.

Public ledger: Public keys are owners of the cryptocurrency. There is a public ledger which records which public key owns what amount of the cryptocurrency. We later describe how the public ledger is maintained and updated and also how new units of the currency are created. For the moment, assume that the public ledger is available and consider the issue of spending.

Transactions: Spending is the transfer of a certain amount of the currency from one public key to one or more public keys. This is done via a transaction which specifies the spender public key and the set of recipient public keys, i.e., the amount of currency owned by the spender public key is to be assigned to one or more recipient public keys. The entire transaction is considered as a message which is digitally signed with the signing key corresponding to the spender public key. This signifies the spender's commitment to the transaction. Since the ledger is publicly available, it is easy to verify from the recorded history in the ledger if the spender public key indeed owned the stated amount of the currency. Also, since the signature can be publicly verified, it is possible to verify that the spender indeed signed the transaction. Once such verification is confirmed and the transaction enters the public ledger, the owners of the recipient public keys may provide the goods and/or services for which the spending has been made. After a transaction becomes a permanent part of the public ledger, the amount of currency associated to the spender public key is considered to be spent and this public key cannot spend this currency any further.

Blockchain: Let us now consider the public ledger. This is structured as a chain of blocks giving rise to the term *blockchain*. Each block consists of a link to the previous block, a set of transactions and a solution to a puzzle. Suppose $B_0, B_1, B_2, \ldots, B_r$ is the current chain of blocks. The block B_0 is called the genesis block. It is a special block and does not contain any link to a previous block. For $i \geq 1$, a block B_i is a triplet (h, R, η), where h is $\mathcal{H}(B_{i-1})$, R is the (ordered) set of transactions in block B_i and

[5] Wikipedia `https://en.wikipedia.org/wiki/Cryptocurrency` mentions: "As of September 2017, over a thousand cryptocurrency specifications exist; most are similar to and derived from the first fully implemented decentralized cryptocurrency, bitcoin."

η is a solution to the puzzle. The value h in B_i is said to point to the previous block B_{i-1}. The last block B_r in the chain is called the header block of the chain.

Difficulty: The puzzle can be described in the following manner. The output of the hash function $\mathcal{H}$ is a binary string of some fixed length n and this string can be considered to be the binary representation of an integer in the range $[0, \ldots, 2^n - 1]$. The puzzle specifies an integer d called the difficulty value. The quantity η in the block B_i is a solution to the puzzle if $\mathcal{H}(B_i) = \mathcal{H}(h, R, \eta) < d$. In this case, the block B_i is said to have been solved with difficulty d. Since $\mathcal{H}$ is one-way, it is computationally difficult to invert $\mathcal{H}$. So, the only way to solve the puzzle is to repeatedly apply $\mathcal{H}$ to (h, R, η) with different values of η until a solution to the puzzle is obtained. Heuristically assuming that the outputs of $\mathcal{H}$ are uniformly distributed (which is a very common assumption), the probability that a single invocation of $\mathcal{H}$ returns a value less than d is $d/2^n$. So, for example, if $d = 2^{n-32}$, then 2^{32} invocations of $\mathcal{H}$ will be required on an average to solve the puzzle specified by d.

The value of d is not constant. The cryptocurrency rules (or protocols) specify how it changes over time. It is determined so that the time required to solve a puzzle does not vary much with time. For example, in the case of Bitcoin, the difficulty is adjusted so that a puzzle is solved in about 10 minutes and 2016 blocks are mined in about two weeks. Starting from the genesis block, segments of every 2016 blocks constitute a window. The difficulty remains constant in each window and based on the time required to complete the current window, the difficulty is determined for the next window. The details of the difficulty adjustment procedure are not important for our purposes and so we do not describe these details.

Distributed public ledger: We have mentioned that the ledger is publicly available. Let us consider how this is done. There is no central user in the system which maintains the public ledger. Instead the system consists of a number of nodes which are globally distributed. A node is essentially a computer running the particular cryptocurrency software. There is no restriction on who can or cannot be a node. The cryptocurrency protocol is publicly available and anybody can implement the protocol on a computer and thus become a node in the system. The chain of blocks which forms the public ledger is maintained by all the nodes in the system. Each node can communicate with its nearby nodes and exchange information regarding transactions and blocks. Information initiated by any node in the system propagates very fast (within seconds) to all the nodes in the system. So, instead of having a central authority maintaining and updating the public ledger, it is maintained and updated in a distributed manner using a network of nodes. Such a system is called a peer-to-peer network.

Creating a new block: We next turn to the issue of how the public ledger is updated and new currency is created. The updation of the ledger consists of appending a new block to the end of the already existing chain of blocks. A block packs a number of transactions. These transactions enter the system through the nodes and are quickly propagated to all the nodes in the system. Any node can select a set R of valid transactions from those which have not been currently assigned to blocks, use the link h which is the output of $\mathcal{H}$ on the header block of the chain that it maintains and then obtain the solution η to the puzzle determined by d. Once a node obtains a solution to the puzzle, it appends the block (h, R, η) to the end of the chain that it locally maintains thus making the new block the current header of the chain and then the node propagates the new block to the other nodes. Any node which receives the block (h, R, η) performs some verification checks on the block and then appends to the end of the chain locally maintained by it thereby also making the new block the current header of its chain.

Updation of the blockchain: Several conflicting scenarios may arise. Suppose a node receives two blocks both of which point to the block which is the header block of the chain that the node presently maintains. Clearly, the node cannot insert both the blocks into the chain. The cryptocurrency protocol specifies which block it should insert as the new header. For Bitcoin, the node selects the block it obtained earlier as the new header of its chain.

More generally, consider the following scenario. Suppose the local chain maintained by a node is $\ldots, B_i, B_{i+1}, \ldots, B_r$. Due to possible delay in the network, suppose it receives a chain of blocks $B'_{i+1}, \ldots, B'_{r'}$ where B'_{i+1} points to B_i. This means that from block B_i the chain has forked, i.e., one segment of the network has created the chain extension $B_{i+1}, \ldots, B_r$ of length $r - 1$ while another segment of the network has created the chain extension $B'_{i+1}, \ldots, B'_{r'}$ of length $r' - 1$. At this point, the node needs to determine which chain it will follow. This is again specified by the rules of the cryptocurrency system. For Bitcoin, the node keeps the extension which is more difficult, i.e., for which the sum of difficulties of the blocks in the extension is greater. If the difficulty did not change during the time interval in which the fork has formed, then greater difficulty amounts to longer chain, i.e., the node keeps the longer of the two chains.

Confirmation of transactions: To diminish the chances of the above kind of transient behaviour, one usually waits until the block containing the transaction gets embedded sufficiently deeply in the chain. For Bitcoin, the recommendation is that one should wait until the transaction is embedded six blocks deep in the chain. Then the chances of the block containing the transaction getting dropped from the chain becomes negligible. Since the difficulty level of Bitcoin is determined so that a block is mined every ten minutes, confirmation of a transaction takes about an hour.

Immutability of the public ledger: As described above, the public ledger is constructed as a chain of blocks. The links in the chain are created using the one-way hash function. As a result, the chain is considered to be immutable in the following sense. Suppose that all the nodes in the system have the same copy of the chain. Let us consider the possibility of going back into the chain and modifying one of the blocks to create a new chain. Since the hash function is one-way, it is not possible to invert the links. So, the only way of creating the new chain is to work forward from the modified block. This requires solving the puzzles for each of the blocks appearing after the modified block. A sufficient number of blocks will have to be added so that the difficulty of the modified chain becomes more than the difficulty of the publicly available chain. Unless one entity acquires majority of the computational power, this will not be possible and the modified chain will be rejected by the nodes of the system.

Mining of blocks: Currency is created through successfully solving the puzzle. This procedure is also called mining a block. When a block is mined, the rules of the cryptocurrency permits assigning a certain amount of the currency to a public key. This amount of currency was not previously assigned to any public key and hence the mining procedure creates new currency. The amount of currency that is created by successful mining of a block is specified by the rules of the cryptocurrency and is called the block reward. It is through the block reward that new currency is created. The block reward is not constant and for Bitcoin it decreases at a constant rate over time.

The presence of block rewards incentivises nodes to invest computer effort into mining a block. The activity is profitable for a node if the cost of mining is less than the block reward. Commercial entities which primarily aim to mine blocks are called miners.

There is competition among miners to perform successful block mining. For Bitcoin, this competition is very intense and has led to a huge amount of dedicated hardware consuming enormous amount of energy for block mining. Block mining essentially boils down to computing the hash function $\mathcal{H}$ a large number of times. So, an entity which can compute $\mathcal{H}$ faster is more likely to solve a puzzle sooner. The hash rate of an entity is the number of times it can invoke $\mathcal{H}$ in one second.

Mining pools: While in principle, any node can be a miner, due to the huge resource requirement, it may not be possible for nodes with limited resources to successfully mine a block in a reasonable amount of time. So, individual nodes join what are called mining pools. Such pools integrate the efforts of many miners and rewards the individual miners by apportioning the obtained block reward among the members of the mining pool.

Double spending: A basic concern for any currency is that there should not be any double spending, i.e. a particular amount of currency should be spent at most once. In the context of cryptocurrency, double spending means that there are two or more transactions which spends the amount of currency associated to a single public key. This should normally not be possible. Once a transaction enters the public ledger through a block, any other node can verify that the amount associated to the spender's public key has been spent and so will reject any transaction which attempts to double spend from this public key. This possibility, however, arises in the following manner. Suppose a miner (or a coalition of miners) spends the amount of currency associated to a public key, creates a block consisting of the transaction and propagates this block. After some time, the block enters the chain of all the nodes and the miner receives the goods and/or services arising out of the spending. Now the miner initiates another transaction where currency is spent from the same public key. Suppose that the miner has sufficient computational power to quickly create a chain of blocks which is more difficult than the chain of blocks available with the other nodes. The miner now creates such a chain and propagates the new chain. Since the rules entail that a node will reject a less difficult chain in favour of a more difficult chain, the nodes in the system reject their present less difficult chains and accept the new more difficult chain. So, the earlier transaction consisting of the spend from the public key in question gets purged from the network and the new transaction enters the system. Correspondingly, the miner is also able to reap the benefits of the spend recorded in the new transaction. This creates a double spending scenario.

The 51% attack: Suppose a miner, or, a mining pool acquires more than 50% of the proportional hash rate of system. Such a powerful entity may gain control over the blockchain. It has more than 50% chance of mining every new block. Correspondingly, it has a high probability of mounting a double spending attack on the system. In the cryptocurrency community this scenario is called a 51% attack. In the case of Bitcoin, between 25th and 27th September 2013, the mining pool named Ghash.io was in possession of 51% of the total hash rate in the Bitcoin network and it launched a double-spending attack on the online gambling website named BetCoin Dice[6].

Cryptocurrency community: There are several kinds of users and entities that make up the community of any particular cryptocurrency.

Miners: These are individual entities who possess a substantial amount of computational ability to have a significant chance of successfully mining a new block.

Mining pools: These are coalitions of individual miners who choose to work together with the goal of successfully mining blocks. Mining pools have declared mechanisms for sharing the block rewards among the members of the mining pools.

---

[6] `https://bitcointalk.org/index.php?topic=321630.0`,`http://arstechnica.com/security/2014/06/bitcoin-security-guarantee-shattered-by-anonymous-miner-with-51-network-power/`

Users: These are persons and entities constituting the end users of the currency. They use the currency for purchasing various kinds of goods and services.

Sellers: These are entities which accept the currency for the goods and services that they provide.

Exchanges: These are organisations which allow conversion between cryptocurrencies and more conventional fiat currencies.

All the entities in the cryptocurrency community are bound by the rules governing the currency which are specified by the protocols of the system. These rules are not enforced by any central agency. Rather, the backing of the rules arise from their being accepted by all the entities in the system. Needless to say, a set of rules determined at a single point of time may not be the best option at all future time points. Thus arises the need to change the rules. A change of rule can affect different entities in different manners. So, not all entities may be agreeable to a proposed rule change. Since a rule is considered valid only if it is accepted by all the entities, it is problematic to change any rule. If one set of entities accept a change while another set does not, then blocks mined by the former set may not be accepted by the later set and vice versa. This can lead to a fork in the system where the blockchain is common up to a certain block and from that point on bifurcates into two distinct chains. Such a fork is called a hard fork.

Entities in a cryptocurrency community have their own particular requirements. We explain this with an example in the context of Bitcoin. A block in the Bitcoin blockchain can accommodate a certain number of transactions. As more and more people begin to use Bitcoins, a lot of transactions are generated. Not all of these can get into the next block and would have to wait quite some time before getting into a block. Also, after it gets into a block there is the recommended one hour waiting time for confirmation as explained earlier. While the one hour waiting time is a security feature, it is indeed possible to change the rules of Bitcoin so that each block can accommodate more transactions. This will reduce the delay time before a transaction can get into a block. Increasing the size of a block constitute a change of rules of the Bitcoin cryptocurrency. For this change to be effected it has to be accepted by all the entities in the community.

Change of rules: There is a procedure for proposing a change of rules. For the Bitcoin cryptocurrency, changes are defined and proposed through a Bitcoin Improvement Proposal (BIP) (Taaki, 2011). For the Ethereum network, it is done similarly through an Ethereum Improvement Proposal (EIP)[7].

Let us consider the Bitcoin cryptocurrency in more details. In addition to the information contained in a block as described above, each block also has a header. This contains meta information such as the time when it was mined. It also contains a provision for indicating support to a proposed BIP. Corresponding to the currently proposed BIPs, there is a set of bits in the header of a block. If the value of a bit corresponding to a BIP is 1, then it indicates support to the BIP, while if the value is 0, then it indicates opposition to the BIP. Upon successful mining of a new block, a miner can specify the values of the bits corresponding to the currently proposed BIPs. These indicate the miner's support to these BIPs.

Evolution of a BIP: Let us consider how a BIP evolves in some more details. A BIP is initially proposed by a handful of developers. For any BIP, one month after it is defined it moves into a phase called the *started* phase. When miners mine blocks, they show their support or disagreement towards a BIP by appropriately setting the value of the corresponding bit in the header of the mined block. A BIP is said to time-out or fail if it is not activated within one year of its being started.

[7] `https://github.com/ethereum/EIPs`

A *target, or, difficulty period* is a window of 2016 blocks during which the difficulty of block mining remains the same. Since the difficulty is adjusted so as to ensure that a target period is two weeks, there are 26 target periods covering the time period of a BIP from when it is started to when it times out. For a BIP, consider one such target period. If in this target period, at least a threshold number of blocks recorded their support for the BIP, then in the following difficulty period all entities who wish to upgrade can do so. After that, the BIP is activated and transactions generated by the new rules are accepted by Bitcoin miners. The required threshold is 1916 blocks ($\approx 95\%$ of 2016). As mentioned above, there are 26 target periods from the proposal of a BIP to its timing out. If in any of these target periods, the BIP receives the required threshold of support, it gets activated.

3 Voting Games Arising from Cryptocurrencies

In this section, we consider the voting games that arise in the context of cryptocurrencies. First we introduce the notion of voting games and then show how such voting games arise in the operation of cryptocurrencies.

3.1 Background on Voting Games

We provide a brief account of voting games. More details can be found in (Felsenthal and Machover, 1998; Chakravarty et al., 2015).

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 $\widehat{G} : 2^N \rightarrow \{0, 1\}$ 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 $L(G)$.

Below we recall some basic notions about voting games. For a finite set S, $\#S$ will denote the cardinality of S.

Swing: For any $S \subseteq N$, $A_i \in N$ is called swing in S if $A_i \in S$, $\widehat{G}(S) = 1$ but $\widehat{G}(S \setminus \{A_i\}) = 0$. For any $S \subseteq N$, $A_i \in N$ is called swing outside S if $\widehat{G}(S) = 0$ but $\widehat{G}(S \cup \{A_i\}) = 1$. For any $A_i \in N$, the number of winning coalitions in which A_i is a swing is the same as the number of losing coalitions outside which A_i is a swing. The number of subsets $S \subset N$ such that A_i is a swing in S will be denoted by $m_G(A_i)$.

Dummy player: A player $A_i \in N$ is called a dummy player if A_i is not a swing in any coalition, i.e., if $m_G(A_i) = 0$.

Minimal winning coalition: A coalition $S \subseteq N$ is called a minimal winning coalition if $\widehat{G}(S) = 1$ and there is no $T \subset S$ for which $\widehat{G}(T) = 1$. The set of all minimal winning coalitions in G will be denoted by $\mathsf{MW}(G)$ and the set of minimal winning coalitions containing the player A_i will be denoted as $\mathsf{MW}_G(A_i)$.

Dictator: A player $A_i \in N$ is called a dictator if $\{A_i\}$ is the only minimal winning coalition.

Blocking coalition: A coalition $S \subseteq N$ is called a blocking coalition in G if $\widehat{G}(N \setminus S) = 0$. A player A_i is called a blocker if $\{A_i\}$ is a blocking coalition. Equivalently, A_i is a blocker if and only if A_i is present in every winning 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 $\omega = \sum_{i=1}^{n} w_i$. The triplet $(N, \mathbf{w}, q)$ defines a weighted majority voting game G given by its characteristic function $\widehat{G} : 2^N \to \{0, 1\}$ in the following manner. 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*

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

We will write $G = (N, \mathbf{w}, q)$ to denote the weighted majority voting game G arising from the triplet $(N, \mathbf{w}, q)$.

3.2 Cryptocurrency Voting Games

Cryptocurrencies give rise to at least two weighted majority voting games. One such game pertains to the 51% attack, while the other pertains to change of rules. To define a weighted majority voting game, it is required to identify the set of players, the weights of the players and the winning threshold.

The set of players: The miners and the mining pools are the players in the game. For the purposes of voting game analysis, the distinction between miners and mining pools is not important. We will simply write miner to mean either an individual miner or a mining pool.

The weights of the players: Intuitively, the weight of a player is its ability to mine a new block. As mentioned earlier, mining a new block is to solve a puzzle and the solution of the puzzle is made possible by repeated application of the hash function $\mathcal{H}$. So, a player which has a higher hash rate is more likely to solve the puzzle. Thus, the hash rate of a miner is a measure of its ability to mine new blocks.

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 hash rate of a miner is not directly available. Instead, it has to be estimated from the activity of the miner. The proportion of the total hash rate that the i-th miner has is h_i/h. It might turn out that the proportional hash rates $h_1/h, \ldots, h_k/h$ are easier to estimate than the absolute hash rates $h_1, \ldots, h_k$. In this case, instead of using $h_1, \ldots, h_k$ as the weights one may use the proportions $h_1/h, \ldots, h_k/h$ as the weights.

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 theoretical aspects of our work is not dependent on the method employed to obtain estimates of the hash rates of the miners. From the point of view of applications, it is of course necessary to know the hash rates so as to be able to apply the theory to the cryptocurrency of interest. Later we consider in more details the application to Bitcoin where the hash rates are assumed to be proportional to the number of blocks mined by a miner in a given time period. We note though that our theory could be equally well applied to hash rates obtained using some other method.

The winning threshold: This depends on the particular game. Below, we identify the thresholds in two such games.

3.3 The Rule Game

The procedure for change of rules has been described above. As explained above, for Bitcoin, this is done through a BIP. Based on this, more generally we identify the Rule Game for change of rules in a cryptocurrency system.

The Rule Game arising from a BIP has the following feature. Once started and before time-out, each of the 26 target periods creates a new voting game for the BIP. As mentioned above, the winning threshold for a BIP is 95%, i.e., at least 95% of the 2016 blocks in a target period must indicate support for the BIP for it to become 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 target period. This, however, 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.

Limitations of the model: Our modelling of rule change has some limitations. These are mentioned below.

1. We assume the weight of a miner to be its hash rate. In a voting game, a player is always able to cast a vote with its weight. In the case of the Rule Game, the casting of votes is implicit and takes place by indicating support or opposition to the proposal in every block mined in a certain interval of time. So, the actual weight of miner's vote is the number of blocks that it is able to mine in the requisite 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, they are

unable to vote in that interval 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. This would lead to the casting of votes by the miners with high hash rates to be more or less equal to their weight.

2. 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 rule 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.

3.4 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 sum total 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. A particular player may choose a coalition depending on the pay-off that it would obtain by participating in that particular coalition. This is a typical scenario of weighted majority voting games.

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. (Eyal and Sirer, 2014) suggested that the Bitcoin system can be attacked with even lower threshold. 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 Analysis of the Rule Game

The BIP game can be analysed from the viewpoint of voting power. Before getting into the details of the analysis, we provide a brief background on the very rich topic of measuring voting powers.

4.1 Measurement of Voting Power

The notion of power is an important concept in a voting system. A *power measure* captures the capability of a player to influence the outcome of a vote.

Given a game G and a player A_i in G, a power measure $\mathcal{P}$ associates a non-negative real number $v_i = \mathcal{P}_G(A_i)$ to the player A_i. The number v_i captures the power that A_i has in the game G. If

$\sum_{A \in G} \mathcal{P}_G(A) = 1$ for all games G, then $\mathcal{P}$ is called a *power index*. In other words, for a power index the powers of the individual players sum to 1.

A widely studied index of voting power is the Shapley-Shubik index (Shapley, 1953; Shapley and Shubik, 1954). This index, however, is defined for a voting game where the order in which the players cast their votes is important. In our application of voting power to the voting games arising from cryptocurrencies, the order of casting votes is not important. So, we do not consider the Shapley-Shubik index in this work. Below we provide the definitions of some of the previously proposed power measures. See (Felsenthal and Machover, 1998; Chakravarty et al., 2015) for further details.

Banzhaf Power Measures. (Banzhaf, 1965) put forward the notion of using the number of swings of a player as a measure of the voting power of the player.

The *raw Banzhaf power measure* $\mathsf{BR}_G(A_i)$ for an entity A_i in the game G is defined as the number of distinct coalitions in which A_i is a swing. Hence,

$$\mathsf{BR}_G(A_i) = m_G(A_i).$$

The *non-normalized Banzhaf power measure* $\mathsf{BZN}_G(A_i)$ is defined as follows.

$$\mathsf{BZN}_G(A_i) = \frac{\mathsf{BR}_G(A_i)}{2^{n-1}} = \frac{m_G(A_i)}{2^{n-1}}.$$

The *Banzhaf normalized power index* $\mathsf{BZ}_G(A_i)$ is defined as follows.

$$\mathsf{BZ}_G(A_i) = \frac{\mathsf{BR}_G(A_i)}{\sum_{j=1}^{n} \mathsf{BR}_G(A_j)} = \frac{m_G(A_i)}{\sum_{j=1}^{n} m_G(A_j)}.$$

Coleman Power Measures. (Coleman, 1971) argued that to interpret 'power' as the 'influence' over the outcome of a coalition, the number of coalitions in which a player $A_i \in N$ is a 'swing in' with respect to the winning coalitions or 'swing outside' with respect to the losing coalitions, determines its influence or voting power.

The *Coleman preventive power measure* $\mathsf{CP}_G(A_i)$ for a player A_i in the game G is a measure of its ability to stop a coalition S from achieving $w_S \geq q$. It is defined as follows.

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

The *Coleman initiative power measure* $\mathsf{CI}_G(A_i)$ for a player A_i in the game G is a measure of its ability to turn an otherwise losing coalition S with $w_S < q$ into a winning coalition with $w_{S \cup \{A_i\}} \geq q$. It is defined as follows.

$$\mathsf{CI}_G(A_i) = \frac{m_G(A_i)}{\#L(G)}.$$

Holler Public Good Index. A public good is the undivided value of the coalition that each player in the system (and not just the winning coalition) will enjoy as a common benefit without rivalry in consumption and with access to all players in the coalition. (Holler, 1982) proposed the public good index $\mathsf{PGI}_{A_i}(G)$ as follows.

$$\mathsf{PGI}_{A_i}(G) = \frac{\#\mathsf{MW}_G(A_i)}{\sum_{A_j \in N} \#\mathsf{MW}_G(A_j)}.$$

The *public good index* for a player A_i differs from the Banzhaf normalized power index in only considering the minimal winning coalitions containing A_i instead of the number of swings. The rationale behind choosing the minimal winning coalitions is as follows. The fact that the undivided value of the coalition gets consumed without rivalry or non-excludability in access, creates scope for free riders. We pointed out before that all players of a minimal winning coalition have a swing position, but it is not necessary that every swing in G is for a minimal winning coalition. Winning coalitions will tend to be minimal because the supply of the public good will anyway be shared with those outside that coalition.

The non-normalised version of $\mathsf{PGI}_{A_i}(G)$ is called the *absolute public good measure*. It is defined as

$$\mathsf{PGM}_{A_i}(G) = \frac{\#\mathsf{MW}_G(A_i)}{\#\mathsf{MW}(G)}.$$

Deegan-Packel Power Measure. (Deegan and Packel, 1978) argued that when the prize for a win is to be split only among the players in the winning coalition, only minimal winning coalitions should be looked at while determining a player's power. They assumed that all minimal winning coalition are equally likely and players in a minimal winning coalition will share the prize of the victory in the winning camp. This implies that any two players belonging to the same minimal winning coalition will have the same power. Based on this idea, the following is defined. The *Deegan-Packel power measure* $\mathsf{DP}_G(A_i)$ for a player A_i in the game G is defined to be

$$\mathsf{DP}_G(A_i) = \frac{1}{\#\mathsf{MW}(G)} \sum_{S \in \mathsf{MW}_G(A_i)} \frac{1}{\#S}.$$

Power Profile. Suppose $\mathcal{P}$ is a measure of voting power. Then $\mathcal{P}$ assigns a non-negative real number to each of the n players in the game. So, $\mathcal{P}$ is given by a vector of non-negative real numbers. We will call this vector to be the $\mathcal{P}$-power profile of the game.

Computing Voting Powers. Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game. Given inputs $\mathbf{w}$ and q, there are known algorithms to compute the various power profiles. We refer to (Chakravarty et al., 2015) for a brief account of the relevant algorithms.

Consider a weighted majority voting game $G = (N, \mathbf{w}, q)$ having n players and weight vector $\mathbf{w} = (w_1, \ldots, w_n)$ with $\omega = \sum_{i=1}^{n} w_i$. The time complexity for computing the above indices is $O(n^2\omega)$. The space complexity of computing the above indices other than the Deegan-Packel index is $O(\omega)$ while for the Deegan-Packel index it is $O(n\omega)$.

4.2 Power in the Rule Game

Among the various proposals for measurement of power of individual players, which is the one suitable in the context of voting games arising from cryptocurrencies? This is an important question whose answer depends on the expectations of the participants in the game. For the BIP game, the winning threshold is 95%. The objective of such a high threshold is to ensure near unanimity. What would be an appropriate power measure of an individual player for such a scenario?

The basic Banzhaf index (or, the non-normalised Banzhaf measure) indicates a player's ability to influence the outcome. This index certainly captures one aspect of a player's voting power. On the other hand, this index does not capture a crucial notion which becomes relevant in the context of the BIP game. Consider a weighted majority voting game $G = (N, \mathbf{w}, q)$. Any player whose weight is greater than $(1 - q)\omega$ (where as before $\omega = \sum_{i=1}^{n} w_i$), is necessarily a blocker. The influence of a blocker in a game should be seen in its role in preventing winning. So, a blocker should be considered from the viewpoint of preventive power.

Theorem 1 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game and suppose a player A is a blocker in the game G. Then*

1. $\mathsf{CP}_G(A) = 1$.

2. $\mathsf{PGM}_G(A) = 1$.

Proof: Consider any winning coalition S. Since A is a blocker, it is present in every winning coalition and hence in S. Further, the removal of A from S results in a losing coalition, as otherwise, A would not be a blocker. So, A is a swing in S. In other words, A is a swing in every winning coalition showing that $m_G(A) = W_G$ and so $\mathsf{CP}_G(A) = 1$.

Any minimal winning coalition T in G is also a winning coalition and since A is a blocker, T must contain A. So, the number of minimal winning coalitions in G is equal to the number of minimal winning coalitions in G containing A. This shows that $\mathsf{PGM}_G(A) = 1$. $\square$

From Proposition 1 we get that the Coleman preventive power measure and the Holler's public good measure assign the maximum power to a blocker. This captures the intuition that a blocker has absolute power in the game. For a game where the threshold q is high, there are many blockers. For example, in the BIP game, the threshold is 95% and so any player whose weight is more than 5% of the total weight is a blocker. For such games, it is perhaps more meaningful to consider the power of a player to prevent winning with a blocker being assigned the maximum possible power. This leads to either of Coleman's preventive power measure or Holler's public good measure as natural choices.

We take a closer look at the scenario. Since the blockers have absolute powers in game G, to consider the powers of the non-blockers, it is helpful to consider the game arising by removing the blockers from G. Motivated by this, we make the following definition.

Definition 2 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game with n players $N = \{A_1, \ldots, A_n\}$ and $\omega = \sum_{i=1}^{n} w_i$. Let B be the set of all blockers in G. Suppose $\#B = n_1$ and w_B be the sum of the weights of all the players in B. Assume that $w_B < q \cdot \omega$. Let $\overline{B} = N \setminus B$ and $w_{\overline{B}} = \omega - w_B$. Define a new game $G_{\overline{B}} = (\overline{B}, \mathbf{w}_{\overline{B}}, q_{\overline{B}})$ with $\mathbf{w}_{\overline{B}}$ to be the weight vector $\mathbf{w}$ restricted to $\overline{B}$ and $q_{\overline{B}} = (q \cdot \omega - w_B)/w_{\overline{B}}$.*

Remark: The condition $w_B < q \cdot \omega$ is required to ensure that $q_{\overline{B}} > 0$. If, on the other hand, $w_B \geq q \cdot \omega$, then the set B of blockers form a minimal winning coalition which is the only minimal winning coalition in the game. Further, none of the non-blockers are swings in any coalition and hence they are dummies. Such a game is relatively simple to analyse and Definition 2 does not cover such games.

Essentially game $G_{\overline{B}}$ is formed by removing the blockers from game G and appropriately recalibrating the winning threshold as is given by the following result.

Proposition 1 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game and B its set of blockers. Let $S \subseteq \overline{B}$.*

1. *S is a winning coalition in $G_{\overline{B}}$ if and only if $S \cup B$ is a winning coalition in G. Consequently, $\#W(G) = \#W(G_{\overline{B}})$.*

2. *S is a minimal winning coalition in $G_{\overline{B}}$ if and only if $S \cup B$ is a minimal winning coalition in G. Consequently, $\#\mathsf{MW}(G) = \#\mathsf{MW}(G_{\overline{B}})$.*

Proof: Let w_S be the sum of weights of all players in S. Then $w_S + w_B$ is the sum of weights of all players in $S \cup B$. The coalition S is winning in $G_{\overline{B}}$ if and only if $w_s \geq q_{\overline{B}} \cdot w_{\overline{B}}$ if and only if $w_s \geq q \cdot \omega - w_B$ if and only if $w_s + w_B \geq q \cdot \omega$ if and only if the coalition $S \cup B$ is winning in G. This shows the first point.

Suppose S is a minimal winning coalition in $G_{\overline{B}}$. Since every element of B is a blocker in G, we have that $S \cup B$ is a minimal winning coalition in G. Conversely, any minimal winning coalition in G is of the form $S \cup B$ where S is a subset of the non-blockers. Since $w_B < q \cdot \omega$, S cannot be empty. Suppose $A \in S$ and has weight w. Since $S \cup B$ is a minimal winning coalition in G, $w_S + w_B \geq q \cdot \omega$ and $w_S + w_B - w < q \cdot \omega$. Recalling that $q_{\overline{B}} = (q \cdot \omega - w_B)/w_{\overline{B}}$ we have $w_S \geq q_{\overline{B}} \cdot w_{\overline{B}}$ and $w_S - w < q_{\overline{B}} \cdot w_{\overline{B}}$. So, dropping any player from S results in a losing coalition in $G_{\overline{B}}$. Therefore S is a minimal winning coalition in $G_{\overline{B}}$. This shows the second point. □

The next result shows that the Coleman preventive power measure and the Holler's public good measure remains unchanged for the non-blockers in moving from game G to game $G_{\overline{B}}$.

Theorem 2 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game and B its set of blockers. Let A be any non-blocker in G. Then*

1. $\mathsf{CP}_G(A) = \mathsf{CP}_{G_{\overline{B}}}(A)$.
2. $\mathsf{PGM}_G(A) = \mathsf{PGM}_{G_{\overline{B}}}(A)$.

Proof: From Proposition 1, we have $\#W(G) = \#W(G_{\overline{B}})$. Suppose $S \subset \overline{B}$. The player A is a swing in S in the game $G_{\overline{B}}$ if and only if "$A \in S$, S is a winning coalition in $G_{\overline{B}}$ and $S \setminus \{A\}$ is a losing coalition in $G_{\overline{B}}$" if and only if "$A \in S$, $S \cup B$ is a winning coalition in G and $(S \cup B) \setminus \{A\}$ is a losing coalition in G" if and only if A is a swing in $S \cup B$ in the game G. So, $m_{G_{\overline{B}}}(A) = m_G(A)$. This fact together with $\#W(G) = \#W(\overline{B})$ shows that $\mathsf{CP}_G(A) = \mathsf{CP}_{G_{\overline{B}}}(A)$.

An almost identical argument shows that $\#\mathsf{MW}_G(A) = \#\mathsf{MW}_{G_{\overline{B}}}(A)$ which combined with $\#\mathsf{MW}(G) = \#\mathsf{MW}(G_{\overline{B}})$ shows that $\mathsf{PGM}_G(A) = \mathsf{PGM}_{G_{\overline{B}}}(A)$. □

There is a consequence of Theorem 2 to the computation of the CP and the PGM indices. This is stated in the following result.

Theorem 3 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game on n players and B be its set of blockers, with $n_1 = \#B$. Let ω be the sum of weights of all the players in G and w_B be the sum of weights of all the blockers. Then the values of $\mathsf{CP}_G(\cdot)$ and $\mathsf{PGM}_G(\cdot)$ for all the non-blockers in G can be computed in time $O(n_2 w_{\overline{B}})$ and requiring space $O(w_{\overline{B}})$ where $n_2 = n - n_1$ and $w_{\overline{B}} = \omega - w_B$.*

Proof: From Theorem 2, for any non-blocker A in G, we have $\mathsf{CP}_G(A) = \mathsf{CP}_{G_{\overline{B}}}(A)$ and $\mathsf{PGM}_G(A) = \mathsf{PGM}_{G_{\overline{B}}}(A)$. So, it is sufficient to compute the values of CP and PGM of all the players in the game $G_{\overline{B}}$. The number of players in $G_{\overline{B}}$ is clearly n_2. The weight of all the players in $G_{\overline{B}}$ is the weight of all the non-blockers in G and this value is equal to ω_2. So, the computation of CP and PGM for all the players in $G_{\overline{B}}$ can be done in $O(n_2\omega_2)$ time and requires $O(\omega_2)$ space. □

In the case where ω is large and there are a number of blockers in G, the savings in time and space guaranteed by Theorem 3 becomes significant.

From Theorem 1 and Theorem 2, we have two power measures which seem to be appropriate to quantify power in the setting where there are a number of blockers and a player's ability to prevent a resolution from being adopted is required to be quantified. The Coleman preventive power measure is based on the number of swings for a player while the Holler public good measure is based on the number of minimal winning coalitions containing a player. Compared to the number of minimal winning coalitions, the number of swings better reflects the variation in the weights. The next result shows that the number of swings increases monotonically with the increase in the weights of the players.

Theorem 4 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game with the set of players $N = \{A_1, \ldots, A_n\}$ and player A_i having weight w_i. If $w_i \le w_j$, then $m_G(A_i) \le m_G(A_j)$ and consequently $\mathsf{CP}_G(A_i) \le \mathsf{CP}_G(A_j)$.*

Proof: Let $\omega = \sum_{i=1}^{n} w_i$. Consider a coalition $S \subseteq N$ of weight w_S in which A_i is a swing. So, S is winning in G but, $S \setminus \{A_i\}$ is losing in G. This means, $w_S/\omega \ge q$ but $(w_S - w_i)/\omega < q$. There are two cases to consider.

- Case $A_j \in S$: Since $w_i \le w_j$, $(w_S - w_j)/\omega \le (w_S - w_i)/\omega < q$. Hence, A_j is also a swing in S.
- Case $A_j \notin S$: Consider the coalition $S' = (S \setminus \{A_i\}) \cup \{A_j\}$ and let its weight be $w_{S'}$. We have $w_{S'} = w_S - w_i + w_j$ and $w_{S'} \setminus \{A_j\} = w_S - w_i$. Then $w_{S'}/\omega = (w_S - w_i + w_j)/\omega \ge w_S/\omega \ge q$. Also, $(w_{S'} - w_j)/\omega = (w_S - w_i)/\omega < q$ which means that A_j is a swing in S'.

The above two points show that for every set S in which A_i is a swing there is a corresponding set S' in which A_j is a swing. From this the theorem follows. □

We next provide an example to show that the number of minimal winning coalitions is not necessarily monotone increasing with the weights of the players.

Example 1: Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game on 6 players $N = \{A_1, \ldots, A_6\}$ with the weight of A_i being w_i. Let $w_1 = w_2 = 8$, $w_3 = 5$, $w_4 = 3$, $w_5 = 2$ and $w_6 = 1$. The sum of weights is $\omega = 27$. Set $q = 20/27$. Then A_1 and A_2 are blockers. The only minimal winning coalition containing A_3 is $\{A_1, A_2, A_3\}$ while $\{A_1, A_2, A_4, A_5\}$ and $\{A_1, A_2, A_4, A_6\}$ are the two minimal winning coalitions containing A_4. So, we have $w_3 > w_4$ but $\mathsf{MW}_G(A_3) < \mathsf{MW}_G(A_4)$.

Given a measure of voting power, it is perhaps not very intuitive that the powers of the players do not increase monotonically with their weights. Such a phenomenon can be difficult to explain to the general public. The two possibilities CP and PGM have been identified earlier as appropriate for capturing the power to prevent action. Among the two, CP is based on swings whereas PGM is based on minimal winning coalitions. Based on Theorem 4 and Example 1, we may conclude that it is more meaningful to choose a voting power measure which is defined based on swings rather than on minimal winning coalitions. So, we suggest that CP, i.e., the Coleman preventive power measure is an appropriate measure of voting power arising in the context of the Rule Game arising in cryptocurrency systems.

5 Analysis of the Attack Game

The features of Attack Game are different from that of the Rule Game, or, other usual voting games. In this game, the goal is not to decide upon some resolution, i.e., there is no proposal which is to be accepted or rejected by the players. Rather, the idea is to secretly form a coalition of miners which possesses 51% or more of the hash rate of the system so that double spending becomes a possibility. In any such coalition, one can still talk about a miner becoming a swing, i.e., the presence of the miner in the coalition makes the hash rate at least 51% while its absence drops the hash rate below 51%. It may be of interest for a miner to know in how many coalitions it plays a swing role. This is captured by the (raw or non-normalised) Banzhaf index.

More generally, the various voting power indices and measures have been proposed with some intuition. For example, the Coleman measures are intended to capture a player's ability to prevent or initiate an action while the Holler indices are intended to capture the notion of power when voting takes place on an issue of public good. The relevance of the background motivation of these indices to the context of the Attack Game is not clear. So, a simple computation of the various indices for this game will not provide much useful information.

On the other hand, certain aspects of the formal analysis of voting games are very relevant to the Attack Game. These are all built around the central concept of minimal winning coalitions. We identify these questions below.

Considering success in the Attack Game to be compromising the security of the system, it is of interest to know the number of different ways in which the system can be compromised. This is formulated as follows.

Question 1 *What is the number of minimal winning coalitions in the game?*

It is unlikely that all possible minimal winning coalitions can actually form. More granular information provides better understanding of the system. The relevant question is the following.

Question 2 *Given a positive integer c, how many minimal winning coalitions of cardinality c are present in the game?*

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 stability is the maximum value of c such that $\mathsf{mw}_c = 0$. This would ensure that the system is secure against a coalition of c or less number of miners. This leads to the following definition.

Definition 3 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game. The game G is said to be $\mathfrak{c}$-stable if $\mathfrak{c} = \max\{c : \mathsf{mw}_c = 0\}$. Equivalently, G is said to be $\mathfrak{c}$-stable if $\mathsf{mw}_c = 0$ for all $c \leq \mathfrak{c}$ and $\mathsf{mw}_{\mathfrak{c}+1} \neq 0$.*

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, we pose the following question.

Question 3 *For a subset S of players and for any positive integer c, how many minimal winning coalitions of cardinality c containing all elements of S are present in the 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 stability of the system 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 4 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game and let $S \subset N$. The game G is said to be $\mathfrak{c}$-stable with respect to S if $\mathfrak{c} = \max\{c : \mathsf{mw}_c(S) = 0\}$. Equivalently, G is said to be $\mathfrak{c}$-stable 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 G be $\mathfrak{c}$-stable with respect to the player A. In this case, we write $\mathsf{mw}_c(A)$ instead of $\mathsf{mw}_c(\{A\})$.

Proposition 2 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game such that G is $\mathfrak{c}$-stable. Then $\mathfrak{c} = \min_{A \in N} \max\{c : \mathsf{mw}_c(A) = 0\}$.*

Proof: Since G is $\mathfrak{c}$-stable, there are no minimal winning coalitions of size c with $c \leq \mathfrak{c}$ and there is at least one minimal winning coalition S of size $\mathfrak{c}+1$. Let A be a player such that G is $\mathfrak{d}$-stable with respect to A and $\mathfrak{d} = \min_{A \in N} \max\{c : \mathsf{mw}_c(A) = 0\}$. Then G does not have any minimal winning coalition of size $\leq \mathfrak{d}$ and it has at least one minimal winning coalition of size $\mathfrak{d}+1$. Then by definition, $\mathfrak{d} = \mathfrak{c}$. □

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. This leads to the following question. More generally, we consider the following question.

Question 4 *Given disjoint subsets S_1 and S_2 of players A and a positive integer c, how many minimal winning coalitions of cardinality c containing all elements of S_1 but not containing any element of S_2 are present in the game?*

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 5 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game and S_1 and S_2 be two subsets of N. G is $\mathfrak{c}$-stable with respect to the pair (S_1, S_2) if $\mathfrak{c} = \max\{c : \mathsf{mw}(S_1, S_2, c) = 0\}$.*

Remarks:

1. Consider $\mathsf{mw}(\emptyset, \emptyset, c)$. This is simply the number of minimal winning coalitions in G as postulated in Question 2.

2. For any player A, $\mathsf{mw}(\{A\}, \emptyset, c)$ is the number of minimal winning coalitions in G containing A and having cardinalities equal to c. So, we get back to the scenario of Question 3. Consequently, G is $\mathfrak{c}$-stable with respect to A if and only if it is $\mathfrak{c}$-stable with respect to the pair $(\{A\}, \emptyset)$.

3. For any subset S of players, $\mathsf{mw}(\emptyset, S, c)$ is the number of minimal winning coalitions in G not containing any element of S and having cardinalities equal to c. Consequently, G is $\mathfrak{c}$-stable 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 players, we ask for the possiblity of the system being compromised by some coalition of the other players. The maximum value of c such that G is $\mathfrak{c}$-stable with respect to the pair $(\emptyset, S)$ provides a measure of stability of the system against coalitions of miners who are not in S.

Definition 6 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game. We say that G is $(\mathfrak{c}_1, \mathfrak{c}_2, \mathfrak{c})$-stable, if*

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

Remarks:

1. If $S_1 = S_2 = \emptyset$, then there are no constraints and in this case $\mathsf{mw}(S_1, S_2, c)$ is the number of minimal winning coalitions of cardinality c in G.
   G is $(0, 0, \mathfrak{c})$-stable 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.
   G is $(1, 0, \mathfrak{c})$-stable if for any player A_i, the size of any minimal winning coalition containing A_i is at least $\mathfrak{c}$.

3. G is $(0, 0, \mathfrak{c})$-stable if and only if the cardinality of any minimal winning coalition in G is at least $\mathfrak{C}+1$. On the other hand, G is $(1, 0, \mathfrak{c})$-stable if and only if the cardinality of any minimal winning coalition in G containing at least one player is at least $\mathfrak{c} + 1$. Since a minimal winning coalition must contain at least one player, it follows that G is $(0, 0, \mathfrak{c})$-stable if and only if G is $(1, 0, \mathfrak{c})$-stable.

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 in G, but, not containing any element of S_2.
   G is $(1, 1, \mathfrak{c})$-stable if for any two players 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.
   G is $(0, \mathfrak{c}_2, \mathfrak{c})$-stable 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 $\mathfrak{c}$.

5.1 Stability with Respect to "Large" Miners

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 who need to form a coalition with the miners in S_1 to win the Attack Game.

Let G be a weighted majority voting game on a set N of players 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 and disjoint from $L \setminus S$ and having cardinalities equal to c.

Definition 7 *Let $G = (N, \mathbf{w}, q)$ be a weighted majority voting game and L be a subset of N. We say that G is $(L, \mathfrak{c}_1, \mathfrak{c})$-stable if*

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

For an $(L, \mathfrak{c}_1, \mathfrak{c})$-stable game G, 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 player in $L \setminus S$. Then to win the Attack Game the coalition S must collude with at least $\mathfrak{c} - \#S$ players from $N \setminus L$.

6 A Snapshot Analysis of Bitcoin

To compute the relevant parameters of the Rule Game and the Attack Game, it is required to identify the actual players and their weights. The players are the miners and the weight of a miner is its hash rate. The hash rate of a miner is not directly available and has to be estimated. As explained earlier, under the assumption that the number of blocks mined by a miner is proportional to the hash rate of the miner, one may use the number of blocks as the weight of the miner. In this section, we proceed under the assumption that the hash rate of a miner is a constant multiple of the number of blocks mined by a miner. So, the weights of the players are taken to be the numbers of blocks mined by these players over some interval of time.

For Bitcoin, several online websites provide information regarding the miners and the numbers of blocks that were mined by the different miners. We have used data from the following website:

`https://www.blocktrail.com/BTC`. From this website, the number of mined blocks can be obtained in a specified previous interval of time. This interval can be the last 24 hours, the last week, the last month, the last six months and the last year. By fixing a particular date and an interval of time, it is possible to obtain a snapshot of the number of blocks generated in the given time interval prior to the given date. The ensuing analysis following from this data provides a snapshot analysis of the Rule and the Attack Games. To illustrate how such a snapshot analysis can be meaningfully carried out, we have done the analysis with data for the last six months from the date 7th November, 2017. This data is shown in Figure 1. Before proceeding with the actual analysis, we note the following point.

1. There is nothing special about the particular date that we have used and the analysis can be applied to data corresponding to any date; also, the analysis can be applied to data obtained for other time intervals.

2. We proceed under the assumption that the hash rate of a miner is proportional to the number of blocks mined by it based on which we have taken the weights of miners to be the numbers of blocks that they mined in a given time period. There is nothing really particular about using the number of blocks as estimates of the hash rates. Our analysis can be applied equally well if the hash rates are estimated using some other methods.

3. While we work with data for Bitcoin, similar analysis can also be applied to data obtained from other cryptocurrencies.

4. The snapshot analysis that we carry out can be performed on a cryptocurrency system at regular time intervals. This will provide valuable insights into the nature of evolution of the socio-economics dynamics of the cryptocurrency system.

The data in Figure 1 attributes a rather high number of blocks to “Unknown”. This means that these many blocks were mined by miners whose identities 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”. It is reasonable to assume that the miners in U are those with limited computational resources, i.e., any one of them would have mined a small number of blocks. So, the weights of the miners in U are small, though the sum total of all these weights is quite significant. We handle the miners in “Unknown” in the following manner.

Suppose the total weight of the components $\mathbf{w}$ is ω out of which the 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 8 *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 $G^{(p)} = (N \setminus \{U\}, \mathbf{w}_{\overline{U}}, q^{(p)})$ where $q^{(p)} = (q \cdot \omega - p \cdot w)/(\omega - w)$ and ω 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 $\omega - w$. To win, a coalition in the original game G needed to have weight at least $q \cdot \omega$. So, in $G^{(p)}$, to win a coalition needs to have weight at least $q \cdot \omega - p \cdot w$.

Block Size Vote	Blocks	%
segwit	10,032	35.56%
default	9,782	34.67%
BU	8,300	29.42%
BIP109	87	0.31%
BIP148	9	0.03%
8MB	4	0.01%
5MB	1	0.00%

Block Versions	Blocks	%
v4	28,215	100.00%

Pool	Blocks	%
AntPool	5,094	18.05%
BTC.TOP	3,572	12.66%
BTC.com	2,740	9.71%
unknown	2,684	9.51%
DiscusFish / F2Pool	2,454	8.70%
BTCC	2,100	7.44%
ViaBTC	1,997	7.08%
Slush	1,528	5.42%
Bitfury	1,378	4.88%
BitClub Network	1,029	3.65%
BW Pool	1,017	3.60%
GBMiners	644	2.28%
Bitcoin.com	538	1.91%
1Hash	530	1.88%
Kano CKPool	331	1.17%
BATPOOL	260	0.92%
Canoe Pool	193	0.68%
ConnectBTC	76	0.27%
P2Pool.org	28	0.10%
Solo CKPool	9	0.03%
BitMinter	8	0.03%
MMPool	2	0.01%
NiceHashSolo	2	0.01%
Eligius	1	0.00%

Figure 1: Number of blocks mined by various miners in the last one six months before 7th November, 2017. Data available from `https://www.blocktrail.com/BTC`.

Remark: Note that setting $p = 1$, we obtain the case where all miners in "Unknown" try to win while setting $p = 0$, we obtain the case where all players in "Unknown" try to block the proposal.

In the game $G = (N, \mathbf{w}, q)$ obtained from Figure 1, there are a total of $n = 24$ miners with the weight vector $\mathbf{w}$ as given in Figure 1 and $q = 0.95$. The value of ω is 28215 and "Unknown" miners have total weight of $w = 2684$. 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. The numbers of the players in $G^{(p)}$ and their weights are shown in Table 1.

6.1 Computation of Voting Powers in the Rule Game

As discussed earlier, in the Rule Game, whether a miner is a blocker is of interest.

Proposition 3 *Let $G = (N, \mathbf{w}, q)$ where the sum of the weights of all the players in G is ω. 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 $w_i > (1 - q)\omega + (p - 1)w$.*

Proof: In $G^{(p)}$, we have $q^{(p)} = (q \cdot \omega - p \cdot w)/(\omega - w)$ and the total weight of all the players in $G^{(p)}$ is $\omega - w$. A_i is a blocker in $G^{(p)}$ if and only if $w_i/(\omega - w) > 1 - q^{(p)}$. Simplifying, we obtain the stated condition. □

Table 1: Numbers and weights of miners other than "Unknown" as obtained from Figure 1.

player	1	2	3	4	5	6	7	8
weight	5094	3572	2740	2454	2100	1997	1528	1378

player	9	10	11	12	13	14	15	16
weight	1029	1017	644	538	530	331	260	193

player	17	18	19	20	21	22	23
weight	76	28	9	8	2	2	1

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 $B^{(p)}$ in $G^{(p)}$ is fixed. From Proposition 1, both the CP and the PGM indices assign a value of 1 to a blocker. So, it is the power of the non-blockers which need to be computed. From Theorem 3, the computation of these powers become more efficient by considering the modified game $G^{(p)}$.

We consider the game $G^{(p)}$ for various values of p. The players of $G^{(p)}$ and their weights are given in Table 1. The power profiles for the CP and PGM for $G^{(p)}$ for various values of p are shown in Tables 2 and 3. In both Tables 2 and 3, a value of 1 in the (i, p) cell indicates that player number i is a blocker in $G^{(p)}$. Based on these tables, we make the following observations.

1. From Table 2 we observe that for a fixed value of p, i.e., in the game $G^{(p)}$, the values of CP decreases monotonically with decrease in the weights. This confirms the behaviour predicted in Theorem 4.

2. From Table 3 we observe that for a fixed value of p, i.e., in the game $G^{(p)}$, the values of PGM do not decrease monotonically with decrease in the weights. For example, from Table 1 the weight of player number 11 is 644 and the weight of player number 12 is 538, yet for $p = 0, 0.1, 0.2$ and 0.3, the PGM value of player number 12 is more than that of player number 11. The possibility of such behaviour was given in Example 1.

3. As p increases, the number of blockers decreases. In Table 2, the numbers of blockers are 8, 6, 6, 5, 4, 3, 2, 2, 2, 1 and 1 corresponding to the values of $p = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ and 1. Player 1 is a blocker in all the games, but as the fraction of miners in "Unknown" who support the rule 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. This, however, is not true for the PGM measure as indicated by the values in Table 3.

6.2 Computation of Swings and Minimal Winning Coalitions 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.

Swings: As mentioned earlier, it is of interest to a miner to know the number of swings that it has in the game. Table 4 provides the values of the non-normalised Banzhaf index of the different players for various values of p. It is of interest to note that as p increases, the power of the player with the highest

Table 2: Values of the Coleman preventive power index of the different players in the Rule Game for various values of p. The entries are shown up to three decimal places.

	p										
player	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
02	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.997	0.992
03	1.000	1.000	1.000	1.000	1.000	1.000	0.993	0.982	0.965	0.947	0.926
04	1.000	1.000	1.000	1.000	1.000	0.991	0.975	0.953	0.930	0.904	0.880
05	1.000	1.000	1.000	1.000	0.982	0.960	0.927	0.894	0.858	0.826	0.792
06	1.000	1.000	1.000	0.990	0.964	0.935	0.900	0.863	0.829	0.793	0.759
07	1.000	0.993	0.952	0.906	0.849	0.793	0.743	0.700	0.662	0.630	0.602
08	1.000	0.962	0.906	0.839	0.780	0.718	0.677	0.635	0.602	0.573	0.548
09	0.912	0.829	0.726	0.641	0.581	0.544	0.517	0.494	0.472	0.445	0.421
10	0.911	0.827	0.720	0.634	0.575	0.539	0.512	0.489	0.466	0.439	0.415
11	0.545	0.461	0.418	0.401	0.377	0.354	0.328	0.313	0.295	0.280	0.265
12	0.475	0.419	0.378	0.358	0.328	0.302	0.282	0.265	0.249	0.236	0.223
13	0.473	0.413	0.372	0.353	0.324	0.297	0.278	0.262	0.246	0.233	0.220
14	0.278	0.252	0.238	0.212	0.204	0.185	0.176	0.163	0.155	0.147	0.139
15	0.266	0.175	0.218	0.151	0.173	0.140	0.142	0.128	0.121	0.117	0.105
16	0.174	0.156	0.139	0.137	0.118	0.114	0.099	0.099	0.090	0.087	0.081
17	0.069	0.052	0.056	0.051	0.042	0.046	0.036	0.037	0.034	0.032	0.032
18	0.027	0.024	0.019	0.019	0.019	0.016	0.014	0.012	0.011	0.010	0.009
19	0.007	0.007	0.005	0.007	0.005	0.005	0.004	0.004	0.004	0.004	0.004
20	0.007	0.006	0.005	0.006	0.004	0.005	0.004	0.004	0.004	0.003	0.003
21	0.001	0.001	0.001	0.002	0.001	0.001	0.001	0.001	0.001	0.001	0.001
22	0.001	0.001	0.001	0.002	0.001	0.001	0.001	0.001	0.001	0.001	0.001
23	0.001	0.001	0.001	0.001	0.001	0.001	0.000	0.001	0.001	0.000	0.000

Table 3: Values of the Holler Public Good Measure of the different players in the Rule Game for various values of p. The entries are shown up to three decimal places.

	p										
player	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
02	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.985	0.993
03	1.000	1.000	1.000	1.000	1.000	1.000	0.994	0.974	0.950	0.924	0.901
04	1.000	1.000	1.000	1.000	1.000	0.994	0.978	0.965	0.960	0.919	0.920
05	1.000	1.000	1.000	1.000	0.983	0.975	0.944	0.896	0.850	0.817	0.806
06	1.000	1.000	1.000	0.960	0.950	0.917	0.866	0.857	0.864	0.855	0.839
07	1.000	0.938	0.913	0.880	0.824	0.790	0.765	0.749	0.761	0.773	0.742
08	1.000	0.979	0.928	0.900	0.882	0.809	0.816	0.745	0.738	0.730	0.730
09	0.935	0.917	0.870	0.820	0.731	0.732	0.737	0.762	0.728	0.695	0.693
10	0.903	0.854	0.812	0.740	0.706	0.675	0.648	0.649	0.621	0.628	0.622
11	0.484	0.438	0.522	0.560	0.689	0.650	0.670	0.658	0.615	0.596	0.577
12	0.613	0.750	0.681	0.640	0.605	0.618	0.615	0.584	0.638	0.613	0.558
13	0.742	0.667	0.609	0.620	0.605	0.561	0.642	0.584	0.611	0.625	0.565
14	0.613	0.438	0.377	0.410	0.420	0.446	0.447	0.455	0.522	0.552	0.551
15	0.677	0.562	0.565	0.550	0.429	0.605	0.413	0.632	0.432	0.578	0.499
16	0.581	0.562	0.449	0.490	0.454	0.490	0.492	0.532	0.465	0.523	0.475
17	0.355	0.479	0.391	0.380	0.395	0.363	0.391	0.398	0.372	0.378	0.395
18	0.355	0.250	0.232	0.280	0.311	0.287	0.302	0.294	0.282	0.297	0.272
19	0.161	0.271	0.203	0.290	0.235	0.236	0.218	0.255	0.239	0.265	0.234
20	0.161	0.292	0.246	0.290	0.227	0.217	0.229	0.264	0.229	0.230	0.229
21	0.097	0.125	0.130	0.190	0.168	0.146	0.134	0.177	0.166	0.163	0.156
22	0.097	0.125	0.130	0.190	0.168	0.146	0.134	0.177	0.166	0.163	0.156
23	0.097	0.104	0.087	0.120	0.109	0.096	0.078	0.117	0.113	0.099	0.099

Table 4: Values of the non-normalised Banzhaf index of the different players in the Attack Game for various values of p. The entries are shown up to three decimal places.

	p										
player	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
01	0.548	0.549	0.548	0.544	0.537	0.528	0.516	0.502	0.486	0.468	0.449
02	0.337	0.337	0.337	0.336	0.333	0.330	0.325	0.320	0.313	0.306	0.298
03	0.256	0.256	0.255	0.254	0.252	0.249	0.245	0.240	0.235	0.229	0.222
04	0.227	0.227	0.227	0.226	0.224	0.221	0.218	0.214	0.210	0.204	0.198
05	0.193	0.193	0.193	0.192	0.190	0.188	0.185	0.182	0.178	0.174	0.169
06	0.183	0.184	0.183	0.183	0.181	0.179	0.176	0.173	0.169	0.165	0.161
07	0.139	0.139	0.139	0.138	0.137	0.136	0.134	0.132	0.129	0.126	0.122
08	0.126	0.126	0.126	0.125	0.124	0.122	0.120	0.118	0.116	0.113	0.110
09	0.093	0.094	0.093	0.093	0.092	0.091	0.090	0.088	0.086	0.084	0.082
10	0.092	0.092	0.092	0.092	0.091	0.090	0.089	0.087	0.085	0.083	0.081
11	0.058	0.058	0.058	0.058	0.058	0.057	0.056	0.055	0.054	0.053	0.051
12	0.049	0.049	0.049	0.048	0.048	0.047	0.047	0.046	0.045	0.044	0.043
13	0.048	0.048	0.048	0.048	0.047	0.047	0.046	0.045	0.044	0.043	0.042
14	0.030	0.030	0.030	0.029	0.029	0.029	0.029	0.028	0.028	0.027	0.026
15	0.024	0.023	0.024	0.023	0.023	0.023	0.023	0.022	0.022	0.021	0.021
16	0.017	0.018	0.017	0.017	0.017	0.017	0.017	0.017	0.016	0.016	0.015
17	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.006	0.006	0.006
18	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.002	0.002	0.002	0.002
19	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
20	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
21	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

weight decreases much more steeply compared to the powers of players with lower weights. Also, more generally, with increase in p, the decrease in power of a player with a larger weight is more significant compared to the decrease in power of a player with a lower weight. This is due to the fact that the total weight of the "Unknown" miners is quite significant and so as p increases, a significant portion of this weight contributes to the attack. This in turn leads to a dimunition of the influence of players with larger weights.

Number of minimal winning coalitions: The cardinality wise number of minimal winning coalitions in $G^{(p)}$ for different values of p are shown in Table 5. 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 $\mathfrak{c} \leq 4$, the value 0 denotes that there is actually no winning coalition in the game while for $\mathfrak{c} \geq 17$, 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 5.

1. There is no winning coalition of cardinality 3 or less.
2. If 20% of the weight of the 'Unknown' miners can be roped in then there is a minimal winning

Table 5: Cardinality wise number of minimal winning coalitions in $G^{(p)}$.

	p										
$\mathfrak{c}$	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	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	1	3	5	7	9	15	21	29
5	23	33	33	54	52	59	79	100	97	110	137
6	118	135	170	185	240	217	230	275	298	315	286
7	331	358	424	403	415	499	490	503	523	562	587
8	584	566	592	695	686	661	709	738	741	668	684
9	700	693	735	682	700	744	719	680	687	681	733
10	654	626	668	625	652	614	607	658	533	558	507
11	493	545	489	518	549	471	484	447	437	476	364
12	407	397	354	420	382	333	342	252	317	253	269
13	305	220	247	251	204	227	190	195	190	141	161
14	120	124	139	128	104	106	100	105	67	77	53
15	49	63	59	46	60	35	35	38	26	29	24
16	20	23	32	25	16	23	12	11	9	5	5
17	3	7	3	5	3	2	1	1	0	0	1
18	0	3	1	0	0	0	0	0	0	0	0
19	0	1	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

coalition of the other 23 miners of cardinality 4.

3. There are several minimal winning coalitions of cardinalities 5 (or more).

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.

In Table 6, we provide the cardinality wise number of minimal winning coalitions containing the largest miner. It is possible to compute similar data for all the players. Table 6 shows that if 20% of the miners in 'Unknown' can be roped in, then the largest miner can form a coalition consisting of itself and 3 of the other 22 miners to win the Attack Game. On the other hand, if coalitions of size 5 or more are considered, then the largest miner can form several winning coalitions in the Attack Game without involving any of the miners in 'Unknown'.

$(L, \mathfrak{c}_1, \mathfrak{c})$-stability: For a set L of large miners, we consider $(L, \mathfrak{c}_1, \mathfrak{c})$-stability 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

Table 6: Cardinality wise number of minimal winning coalitions containing the largest miner in $G^{(p)}$.

	p										
$\mathfrak{c}$	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	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	1	3	5	7	9	15	21	29
5	23	33	33	54	52	59	78	99	93	105	126
6	117	133	166	179	228	196	203	230	250	237	205
7	310	331	383	338	316	384	377	345	344	311	298
8	497	434	424	500	483	418	419	392	348	322	294
9	476	482	501	406	379	379	334	309	276	214	249
10	431	380	371	324	320	273	253	233	169	199	140
11	277	301	258	265	250	179	173	162	154	150	80
12	248	218	178	179	151	131	137	78	96	57	59
13	170	104	99	102	70	71	52	51	32	26	34
14	42	49	54	37	31	34	15	23	7	12	1
15	18	23	20	14	10	8	6	2	1	4	0
16	6	9	4	8	2	2	1	1	2	0	0
17	3	0	1	0	0	1	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
22	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0

each case, we have computed the corresponding value of $\mathfrak{c}$. Table 7 provides values of $\mathfrak{d}$ such that $G^{(p)}$ is $(L, \mathfrak{c}_1, \mathfrak{c}_1 + \mathfrak{d} - 1)$-stable for different values of p and $\mathfrak{c}_1$. 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 7, we make the following observations.

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. Similar observation holds for $\#L = 5$ and $\#L = 6$. Put another way, any attack on the system certainly involves one of the six largest miners.

2. Case $\#L = 5$ and $\mathfrak{c}_1 = 5$. 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 entry for $p = 0$ is '–'. This means that if none of the miners in 'Unknown' are involved in the attack, then the attack cannot be successful. On the other hand, the entry for $p = 0.1$ is 18. This means that if 10% of the miners in 'Unknown' are involved in the attack, then leaving out the three largest miners, a coalition of 18 of the other 20 miners in $N \setminus L$ 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 in $N \setminus L$, 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 5. 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 18 other miners in $N \setminus L$, it is necessary and sufficient to have only 5 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 3. 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, a larger number of miners is required from the remaining $N \setminus L$ miners.

Remark: For the analysis of the Attack Game, we have considered only the cardinalities of the minimal winning coalitions under different settings. From a practical point of view, it would be of interest to obtain the actual minimal winning coalitions. The algorithms for computing the cardinalities of the minimal winning coalitions can be extended to compute the actual sets of players who form the relevant minimal winning coalitions. We refer to (Chakravarty et al., 2015) for such details.

7 Conclusion

In this work, we have shown that weighted majority voting games arise naturally in the context of cryptocurrencies. Two such games, namely the Attack Game and the Rule Game have been identified and appropriately analysed. We hope that this will stimulate interest in the connection between cryptocurrencies and voting games leading to further interesting work on the intersection of these two topics.

Table 7: The entries in the table are $\mathfrak{d}$ such that $G^{(p)}$ is $(L, \mathfrak{c}_1, \mathfrak{c}_1 + \mathfrak{d} - 1)$-stable 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	6	6	6	6	6	5	5	5	5	5
	$\mathfrak{c}_1 = 1$	4	4	3	3	3	3	3	3	3	3	3
$\#L = 2$	$\mathfrak{c}_1 = 0$	9	8	8	8	8	7	7	7	7	6	6
	$\mathfrak{c}_1 = 1$	5	4	4	4	4	4	4	4	3	3	3
	$\mathfrak{c}_1 = 2$	3	3	2	2	2	2	2	2	2	2	2
$\#L = 3$	$\mathfrak{c}_1 = 0$	–	18	13	12	11	10	10	9	9	8	8
	$\mathfrak{c}_1 = 1$	5	5	5	5	5	4	4	4	4	4	4
	$\mathfrak{c}_1 = 2$	3	3	3	3	3	2	2	2	2	2	2
	$\mathfrak{c}_1 = 3$	2	2	1	1	1	1	1	1	1	1	1
$\#L = 4$	$\mathfrak{c}_1 = 0$	–	–	–	–	–	–	–	–	–	–	–
	$\mathfrak{c}_1 = 1$	7	6	6	6	6	5	5	5	5	4	4
	$\mathfrak{c}_1 = 2$	4	3	3	3	3	3	3	2	2	2	2
	$\mathfrak{c}_1 = 3$	2	2	2	2	1	1	1	1	1	1	1
	$\mathfrak{c}_1 = 4$	1	1	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
$\#L = 5$	$\mathfrak{c}_1 = 0$	–	–	–	–	–	–	–	–	–	–	–
	$\mathfrak{c}_1 = 1$	11	10	9	8	8	7	7	6	6	5	5
	$\mathfrak{c}_1 = 2$	4	4	4	4	3	3	3	3	3	2	2
	$\mathfrak{c}_1 = 3$	2	2	2	2	1	1	1	1	1	1	1
	$\mathfrak{c}_1 = 4$	1	1	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
	$\mathfrak{c}_1 = 5$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
$\#L = 6$	$\mathfrak{c}_1 = 0$	–	–	–	–	–	–	–	–	–	–	–
	$\mathfrak{c}_1 = 1$	–	–	–	–	–	–	–	10	9	8	7
	$\mathfrak{c}_1 = 2$	6	5	5	4	4	4	4	3	3	3	3
	$\mathfrak{c}_1 = 3$	3	2	2	2	2	2	1	1	1	1	1
	$\mathfrak{c}_1 = 4$	1	1	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
	$\mathfrak{c}_1 = 5$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
	$\mathfrak{c}_1 = 6$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$

References

Banzhaf, J. F. (1965). Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19:317–343.

Chakravarty, S. R., Mitra, M., and Sarkar, P. (2015). *A Course on Cooperative Game Theory*. Cambridge University Press.

Coleman, J. S. (1971). Control of collectives and the power of a collectivity to act. In Lieberma, B., editor, *Social Choice*, pages 269–298. Gordon and Breach, New York.

Deegan, J. and Packel, E. W. (1978). A new index of power for simple n-person games. *International Journal of Game Theory*, 7(2):113–123.

Eyal, I. and Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. In Christin, N. and Safavi-Naini, R., editors, *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, volume 8437 of *Lecture Notes in Computer Science*, pages 436–454. Springer.

Felsenthal, D. S. and Machover, M. (1998). *The Measurement of Voting Power*. Edward Elgar, Cheltenham.

Fisch, B. A., Pass, R., and A.Shelat (2017). Socially optimal mining pools. In Devanur, N. R. and Lu, P., editors, *Proceedings of the 2017 International Conference on Web and Internet Economics, WINE, 2017, Bengaluru, India, December 17–20, 2017*, volume 10660 of *Lecture Notes in Computer Science*, pages 205–218. Springer.

Holler, M. J. (1982). Forming coalitions and measuring voting power. *Political Studies*, 30(2):262–271.

Holler, M. J. and Packel, E. W. (1983). Power, luck and the right index. *Journal of Economics*, 43(1):21–29.

Kiayias, A., Koutsoupias, E., Kyropoulou, M., and Tselekounis, Y. (2016). Blockchain mining games. In Conitzer, V., Bergemann, D., and Chen, Y., editors, *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 365–382. ACM.

Kroll, J., Davey, I., and Felten, E. W. (2013). The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. In *Workshop on the Economics of Information Security*.

Lewenberg, Y., Bachrach, Y., Sompolinsky, Y., Zohar, A., and Rosenschein, J. S. (2015). Bitcoin mining pools: A cooperative game theoretic analysis. In Weiss, G., Yolum, P., Bordini, R. H., and Elkind, E., editors, *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 919–927. ACM.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`.

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

Shapley, L. S. (1953). A value for n-person games. In Kuhn, H. W. and Tucker, A. W., editors, *Contributions to the Theory of Games II (Annals of Mathematics Studies)*, pages 307–317. Princeton University Press.

Shapley, L. S. and Shubik, M. J. (1954). A method for evaluating the distribution of power in a committee system. *American Political Science Review*, 48:787–792.

Taaki, A. (2011). Bitcoin Improvement Proposal 1. `https://github.com/bitcoin/bips/blob/master/bip-0001.mediawiki`, created on 19-08-2011.