Just Enough Security: Reducing Proof-of-Work Ecological Footprint

Itay Tsabary
Technion, IC3
sitay@campus.technion.ac.il

Alexander Spiegelman
VMware Research
spiegelmans@vmware.com

Ittay Eyal
Technion, VMware Research, IC3
ittay@technion.ac.il

***Abstract*—*Proof-of-work* (*PoW*) mechanisms secure about 80% of the $250B cryptocurrency market. PoW requires system participants to expend computational resources, and protects the system from attackers who cannot expend resources at an equivalent rate. These systems operate in the *permissionless* setting and compensate their users with cryptocurrency, having a monetary value. As cryptocurrency prices sore so do the invested resources, and Bitcoin expenditures alone are 0.24% of the global electricity consumption. Arguably, this is superfluous, and lowering the ecological footprint justifies settling for a lower attack threshold.**

We present novel protocols that allow the system designer to accurately trade off security for expenditure reduction. To the best of our knowledge, this is the first work to do so without adding qualitatively stronger model assumptions. Moreover, our protocols reduce PoW resource expenditure significantly, but with only limited security degradation.

To analyze these protocols We refine the common blockchain model to take into account the cryptocurrency value in real terms, expenditure, and security metrics, distinguishing common revenue-seeking attacks from sabotage. Our analysis of game-theoretic and economical properties of the protocols can be used to tune blockchain security to its required level and limit its ecological damage.

I. Introduction

Recent years, since Nakamoto's introduction of the *Nakamoto's blockchain* and Bitcoin [1], have seen a renaissance of *cryptocurrencies*, with a market cap estimated at $250B [2], [3], [4]. Cryptocurrencies enable minting and transacting of tokens among the system clients. Unlike pre-Bitcoin systems, they are *decentralized*, with no privileged entities, allowing agents, named miners, to join the system without permission from existing participants.

Over 80% [2] of the cryptocurrency market cap comprises systems such as Bitcoin [1] and Ethereum [5] utilizing *PoW* [6], [7] for security. PoW protocols require computational resources expenditure for participation, and reward participants with cryptocurrency for their efforts. Thus, higher cryptocurrency prices imply higher rewards for miners, leading to more mining power joining the system, and better security.

However, with the rise of cryptocurrency prices, resource consumption has become excessive [8], [9], [10], [11], [12], [13]. The amount of resources spent on PoW mining has been exponentially growing [14], [15], with Bitcoin consuming 0.24% of the global electricity usage [16], [17], surpassing countries like Austria and Colombia. The required computational resources [18], [19], [20] imply a significant, arguably superfluous, ecological impact.

Previous work (§II) explored less environmentally-damaging alternatives, including wasting different resources [21], owning, but not spending, the cryptocurrency tokens instead of external resources [22], [23], and advanced versions of classical centralized solutions [24]. However, these approaches require making qualitatively stronger assumptions.

In this work we present two alternative PoW cryptocurrency protocols that reduce external expenditure compared to existing protocols, but with only a limited security reduction. We start by modeling a cryptocurrency system as a game (§III), similar to previous work [25], [26], [27], [28], [29]. The participants strive to maximize their profits from mining. Unlike previous work, we explicitly define the relation between the cryptocurrency and external expenses, and how minting affects that relation: We assume that the monetary value of the entire cryptocurrency (market cap) is determined by factors outside the scope of this work (adoption, regulation, etc.), and the external monetary value of each internal coin is its fraction out of the entire supply. Nakamoto [1] naturally fits in our model (§IV).

Note that the trivial solution of reducing the

Some of the work was done during the first author's internship at VMware Research Group, Israel.

rewards, e.g., by half, is ineffective. Suppose, for example, the Bitcoin system would have minted coins at half the rate. Everything else being equal, the value of each token in this system would be twice that of the actual system. Miners would receive half the nominal cryptocurrency amount compared to Bitcoin, but spend the same resources as in Bitcoin.

To evaluate our protocols we consider six metrics (§V). Our goal is to reduce the amount spent by the miner on PoW. We also consider commonly used metrics such as coalition resistance and the system's tendency to encourage coalitions (and centralization). We define a metric of *permissiveness*, which describes the resources necessary for a new participant to join. Finally, we refine security metrics to consider the threshold cost of safety-violating attacks in general and free attacks in particular.

Both of the protocols we present mint rewards at the same rate as a pure PoW algorithm, implying the same coin value and inflation rate, and allowing a direct comparison. Both of them are parametrized generalizations of the Nakamoto blockchain.

The first protocol, (ℓ, α) *-PR*, awards a miner who generates a block only α of the block reward, and distributes the remaining $(1 - \alpha)$ among the other participants of the system (§VI). The protocol thus fractioning the miners' incentive for external expenditure by α, achieving our primary goal. However, all attacks on the system security now become cheaper by α as well. This protocol therefore provides an additional knob for the system designer, to trade off security for environmental friendliness.

The second protocol, (ℓ, α, F) *-IIE*, aims to improve on (ℓ, α) *-PR* by reducing its security degradation compared to pure PoW. (ℓ, α, F) *-IIE* allows miners to give away an amount of cryptocurrency in advance, in order to increase their block rewards (§VII). This incentivizes miners to allocate and spend a portion of their resources internally, keeping the miner expenses at the same level as pure PoW, but reducing the external expenditure. This mechanism introduces a new trade off: Joining miners who do not yet own cryptocurrency cannot obtain it without the permission of the existing miners, and participating as such will result in lower revenues. Therefore, this protocol is less permissive, monetary wise, than the pure PoW one, or (ℓ, α) *-PR*. Its advantage is that it deters adversaries who seek to increase their revenue rather than to sabotage the system. Such revenue-seeking adversaries are, arguably, the main threat to cryptocurrencies, whereas sabotage is less common due to the requirement for sustained effort and lack of direct motive.

Implementing these protocols raise several practical considerations (§VIII). The shift to reduced external expenditure should follow a ramp-up period where pure PoW is used, until sufficient mining power secures the system. To perform the cryptocurrency allocation of (ℓ, α, F) *-IIE*, miners can pay to a dedicated address, and the system can disperse this amount homogeneously among the participants, or to a random subset.

In summary, we make the following contributions:

1) A model for PoW blockchains that relates internal and external currencies
2) Refined metrics for cryptocurrency security
3) Protocol with tunable expenditure and security
4) A more centralized protocol with tunable expenditure, and only somewhat reduced security

II. Related Work

Many previous work focus on alternatives to the wastefulness of PoW. One type of work assumes a known and static set of participants, and lets them run a byzantine fault tolerant [30], [31], [32], [33], [34] algorithm to determine system state. These systems are *permissioned*, as participants require authorization to join them. Such solutions are common in the enterprise market [35], [24] but are impertinent for permissionless cryptocurrency systems, where the participant set is inherently unknown and dynamic. Contrarily, we operate in the *permissionless* setting, allowing a dynamic participation set.

An alternative solution is in the form of *proof-of-stake*, instantiating systems such as Algorand [22] and Ouroboros [23], [36]. On the intuitive level, these system implement a reduction from permissionless to permissioned networks — out of a possibly infinite set of potential participants, only those who have stake in the system (i.e., own tokens) can participate. As in with the previous approach, the participants run an agreement protocol to determine the system state. However, proof-of-stake systems have an inherent drawback. To participate in the agreement protocol, one must obtain tokens first, requiring the permission of current system participants. Essentially, new willing users can be prevented from joining the system. Variously, one of our suggested protocols is completely free of such predicament, while in another new users lacking the permission of others will be able to participate but their expected rewards will be lower compared to if they had permission. Our third protocol suffer the same impasse.

Furthermore, these systems prove security under various assumptions. Namely, Algorand [23] does not consider participation incentives and assumes the stake majority is held by altruistic parties. Contrarily, Ouroboros [22] considers stake-holder rewards and shows incentive-compatibility for certain scenarios. However, to combat *long-range* attacks [37], [38] it has to either trust its users to willingly delete their private keys once used, or assume network synchrony and user availability since its inauguration.

A different suggestion for PoW replacement is *proof-of-burn* [39], [40], in which miners prove the depletion of another cryptocurrency, typically PoW-based, to create blocks. This scheme hence outsources external expenditures, maintaining the negative environmental impact. Alternatively, our protocols focus on the reduction of expended computational resources, without effectively outsourcing the minting issue.

Another line of previous work focuses on incentive compatibility analysis of classical Nakamoto systems [25], [26], [41], [27], [28], [42], [43], [44], [45], [29]. These work typically consider a set of specific protocols, which they analyze in the following manner. First, they introduce a model capturing the main properties of said PoW systems. Then, they formulate the system as a game, where the system participants are the players, their strategies are the way they interact with the system and among each other, and the system state implies player utilities. They follow up with analysis of said strategies utilizing both analytical game-theory methods as well as heuristic and numeric search methods. Namely, Sapirshtein et al. [26], Gervais et al. [29] and Zhang et al. [45] all utilize Markov-Decision-Process to derive plausible attack vectors. By the analysis they draw their results, bettering the understanding of these system incentive mechanisms. Similarly, we also define a cryptocurrency model with which we analyze our protocols. We also utilize game theoretic analysis and MDP, both to reason about plausible user strategies. Differently, we also engage in mechanism design, presenting new protocols. We also define evaluation metrics, with which we compare and contrast our proposed protocols with NC.

Chen et al. [46] defines properties of reward allocation rules in a PoW system. It focuses on theoretical proofs for a family of reward allocation rules for a single block, and does not consider environmental impact nor plausible attack vectors. Contrarily, we present system-level evaluation metrics, aiming to quantify monetary costs of attacks and electricity consumption. We assert that our proposed protocols all match their presented axioms.

III. Model

We now present a model for an abstract blockchain system, instantiated with a cryptocurrency mechanism. We clearly distinguish the blockchain from the cryptocurrency, allowing us to mull over several designed cryptocurrency mechanism.

We denote the cryptocurrency protocol as Π, and begin by modeling how the value of cryptocurrency is determined by external participants (§III-A). We then detail the system state (§III-B), participating entities (§III-C), and the execution (§III-D). We follow with formalizing the cryptocurrency system as a game, to be used in our game-theoretic analysis (§III-E). We emphasize generality of the model, and that participating player plausible strategies and utilities derive from the particular cryptocurrency protocol instance.

A. Internal and External Currencies

The cryptocurrency system facilitates an internal currency *ic* (e.g., Ether [5]). Let there be some external currency *ec* (e.g., EUR, USD, RMB). We assume the external currency has a market cap order of magnitudes larger than that of the cryptocurrency [47], and possible minting of new *ec* is negligible.

We make the following observation — cryptocurrencies derive their value from their sparsity, and purchasing a cryptocurrency token is essentially owning a portion of a scarce resource. As with every economy, an increase in inflation means a decrease in the purchasing power of a single token, That is, the value of a single token derives from the total market cap of the cryptocurrency, divided by number of present tokens. We assume such market cap is a function of the currency's adoption, usability, legislation and other considerations out of the scope of this work [48], [49], [50], [51], [52], [53].

Throughout this work when we compare two different cryptocurrency systems we assume their market caps are equal, regardless of their coin granularities. It immediately follows that if two such systems mints the same number of coins at the same period of time, then the monetary value in *ec* of each of these coins is equal.

We assume there is an instantaneous, frictionless, and unlimited exchange service of *ec* and *ic* currency, where the exchange rate is in accordance with the

coin granularity and market cap. For ease of writing we set that rate to be exactly 1.

B. System State

The system comprises a shared *global storage* S^g and a dynamic set of users.

The global storage is an append-only set containing elements called *blocks*. Each block includes a reference to another block, with the only a so-called *genesis block* that does not reference any block. The global storage S^g initially contains only the genesis block. Hence, blocks define a directed tree data structure.

We refer to paths in the data structure starting with the genesis block and ending at a leaf block as *chains*. We denote for any chain C its last block and length as $\mathit{last}(C)$ and $\mathit{length}(C)$, respectively. We say a block's *height* is $h \in \mathbb{N}$ if it is exactly h blocks apart on a path from the genesis block.

Let there be the function $\mathit{LC}(S^g)$, returning an arbitrarily-yet-deterministically ordered set of the longest chains in S^g. We refer to the j'th element of $\mathit{LC}(S^g)$ as $\mathit{LC}_j(S^g)$. Also let there be the function $\mathit{CP}(S^g)$, returning the longest common chain prefix of the longest chains in S^g. Note that if there is a single longest chain ($|\mathit{LC}(S^g)| = 1$) then that chain is also the longest common prefix of the longest chains, meaning $\mathit{LC}_1(S^g) = \mathit{CP}(S^g)$, and we often refer to it as the *main chain*.

The system implements a state machine, and users derive the current system state by parsing the content of S^g. The cryptocurrency protocol Π is defined by a parameter ℓ and a function $\mathit{Bal}(\mathit{CP}(S^g))$. Specifically, for any user i the function $\mathit{Bal}_i(\mathit{CP}(S^g))$ returns a value in $\mathbb{R}_{>0}$, denoted as the cryptocurrency *possession* in *ic* of that user. We omit the user index i when referring to the entirety of users.

Function $\mathit{Bal}(\mathit{CP}(S^g))$ also implements the minting of new currency. It divides $\mathit{CP}(S^g)$ to *epochs* of ℓ blocks. For any $k \in \mathbb{N}$, epoch k includes the series of blocks $[\ell k + 1, \ell(k+1)]$ in $\mathit{CP}(S^g)$. We denote by $N_i^k(C)$ the number of blocks in epoch k created by agent i on chain C.

The protocol distributes the newly minted currency at the concluding block of every epoch. That is, the overall cryptocurrency possessions grow at epoch conclusion by the minted amount, and $\mathit{Bal}(\mathit{CP}(S^g))$ details how these coins were shared among the system users. The number of generated coins in epoch k is ℓr_k, averaging r_k coins per block. The protocol may set different r_k for every epoch.

Note. *One can consider an implementation of* $Bal()$ *taking as input one of the longest chains* $\mathit{LC}(S^g)$*, however different users may consider different chains, resulting with inconsistent local state.*

To avoid that protocols often make the new coins available only after sufficiently many other blocks are created. For example, Bitcoin [1] makes available coins minted in block x *only after block* $x + 100$ *is created. We could have incorporated such a mechanism as well, but opted not to for ease of writing.*

C. System Users

For every epoch k we distinguish two complementary user sets. First, we consider *agents*, users actively participating in the system maintaining the system. Contrarily, we denote passively-present users that do not support system sustenance as *clients*. We denote the set of all agents and clients of epoch k as $\mathcal{A}(k)$ and $\mathcal{C}(k)$, respectively. Agents are the main system actors, while clients are of no interest to us except their aggregated cryptocurrency holdings. We assume that $\mathcal{A}(k)$ and $\mathcal{C}(k)$ are fixed throughout the epoch, and denote $n(k) \triangleq |\mathcal{A}(k)|$. We often omit the epoch index when clear from context.

Similarly to their *ic* possessions, agents also own *ec*. We use the terms *external* and *internal* to distinguish the different currency holdings, and *budget* to describe the overall currency value owned by agents. Agent budgets are set exogenously (in another game theory phrasing – by *nature*) at every epoch beginning. This includes possibly exchanging any previous *ic* holdings for *ec*, and adding or removing *ec* to fit the set budget.

For every epoch k we denote $B_i^{ec}(k)$ and $B_i^{ic}(k)$ as the external and internal currency value of each agent $i \in \mathcal{A}(k)$, both measured in *ec*, respectively. Ergo, the budget of each agent i is $B_i(k) \triangleq B_i^{ec}(k) + B_i^{ic}(k)$, and her relative budget is $b_i(k) \triangleq \frac{B_i(k)}{\sum_{j=1}^{n} B_j(k)}$. We denote $B_{\mathcal{A}}^{ic}(k) \triangleq \sum_{j=1}^{n} B_j^{ic}(k)$ and $B_{\mathcal{A}}^{ec}(k) \triangleq \sum_{j=1}^{n} B_j^{ec}(k)$ the accumulated value of internal and external currency all agents own, respectively. Similarly, we denote the external and internal relative budgets of agent i as $b_i^{ec}(k) \triangleq \frac{B_i^{ec}(k)}{B_{\mathcal{A}}^{ec}(k)}$ and $b_i^{ic}(k) \triangleq \frac{B_i^{ic}(k)}{B_{\mathcal{A}}^{ic}(k)}$, respectively. We also denote $B_{\mathcal{A}}(k) \triangleq B_{\mathcal{A}}^{ic}(k) + B_{\mathcal{A}}^{ec}(k)$. We denote by $B_{\mathcal{C}}^{ic}(k)$ the cryptocurrency value of all system clients.

Agents can exchange their owned internal and external currency with the exchange service. We use the term *apportion* to describe agents balancing their internal and external currency holdings to a specific ratio. Agent i apportions her budget $B_i(k)$ with

the invocation of function $\mathsf{Apportion}_i\left(S^g, B_i\left(k\right)\right)$, returning a tuple of her set internal and external budgets $\left\langle B_i^{ic}\left(k\right), B_i^{ec}\left(k\right)\right\rangle$.

Agents maintain the cryptocurrency system through the expenditure of their budgets. The cryptocurrency protocol Π may require agents to explicitly spend resources either internally, externally or a combination of both. Hence, the way an agent apportions her budget affects the way she can act within the protocol. We consider agents that expend the entirety of their budgets per epoch.

Each agent i has a local storage S_i^l accessible only to her. Like the global storage, the local storage is also an append-only block set. Agent i creates a block locally when invoked with the function $\mathsf{Generate}_i^{\Pi}\left(S^g, S_i^l\right)$, returning a newly generated block. The cryptocurrency protocol Π states validity rules of which blocks must abide, and invalid blocks do not affect the system state. Creating invalid blocks is futile and we consider agents who avoid doing so.

Agent i may add any of her previously-private local blocks to S^g when invoked with function $\mathsf{Publish}_i\left(S^g, S_i^l\right)$, returning a set of local blocks for publication. The cryptocurrency protocol Π states prescribed implementations for any agent i of $\mathsf{Apportion}_i\left(S^g, B_i\left(k\right)\right)$, $\mathsf{Generate}_i^{\Pi}\left(S^g, S_i^l\right)$, and $\mathsf{Publish}_i\left(S^g, S_i^l\right)$, which we refer to as *prescribed equilibrium strategy* and denote as $\sigma_{desired}$. Note that Π cannot oblige agents to follow $\sigma_{desired}$, and each agent may choose her own function implementations.

We assume that monetary-wise the total maintenance expenses of the system for a single epoch k are negligible compared to the monetary value it serves, that is $B_{\mathcal{A}}\left(k\right) \ll B_{\mathcal{C}}^{ic}\left(k\right)$.

Note. *In the time of writing these lines creating a Bitcoin block costs (and rewards) the miner $100K, while Bitcoin has a total market of $150B (USD). Ethereum market cap is of $20B while the block reward is roughly $10k (USD).*

D. Execution

Initially the global storage S^g contains only the genesis block, and each agent i has an empty local storage $S_i^l = \emptyset$.

The system progresses in epochs as detailed in Algorithm 1. Each epoch k with its agent set $\mathcal{A}\left(k\right)$ begins when $\mathit{length}\left(\mathit{CP}\left(S^g\right)\right) = \ell k$.

First, the *nature* exogenously sets the budget for each agent i in $\mathcal{A}\left(k\right)$, upholding $B_{\mathcal{A}}\left(k\right) \ll B_{\mathcal{C}}^{ic}\left(k\right)$ (line 1). Then, the scheduler invokes $\mathsf{Apportion}_i\left(S^g, B_i\left(k\right)\right)$ for each agent i, apportioning their budgets (lines 3 – 4).

Algorithm 1: Scheduler in epoch k

```
   /* Initial storage state */
   input : S^g such that length(CP(S^g)) = ℓk
   /* Budget distribution */
1  for i ← 1 to n do
2      B_i(k) ← v ∈ ℝ_{>0}, chosen by nature such that
           B_A(k) ≪ B_C^{ic}(k)
   /* Apportion */
3  for i ← 1 to n do
4      ⟨B_i^{ic}(k), B_i^{ec}(k)⟩ ← Apportion_i(S^g, B_i(k))
   /* Extention */
5  while length(CP(S^g)) < ℓ(k+1) do
       // Generation
6      i ← agent index chosen at random,
           ∀ j ∈ 1,...,n : Pr(i = j) = b_j^{ec}(k)
7      S_i^l ← S_i^l ∪ {Generate_i^Π(S^g, S_i^l)}
       // Publication
8      blocks_to_publish ← ∅
9      do
10         S^g ← S^g ∪ blocks_to_publish
11         blocks_to_publish ← ∅
12         for i ← 1 to n do
13             blocks_to_publish ←
                   blocks_to_publish ∪ Publish_i(S^g, S_i^l)
14     while blocks_to_publish ≠ ∅
```

The rest of the epoch execution progresses in steps, until longest common prefix chain is extended by ℓ blocks (lines 5 – 14). Each step begins with the scheduler probabilistically selecting a single agent i according her relative external expenditure, that is $\forall\ j \in\ 1,...,n : \Pr\left(i = j\right) = b_j^{ec}\left(k\right)$ (line 6). It then invokes the agent i's $\mathsf{Generate}_i^{\Pi}\left(S^g, S_i^l\right)$ function and adds the returned block to her local storage S_i^l (line 7). Then, the scheduler lets all agents publish their local blocks via invoking $\mathsf{Publish}\left(S^g, S^l\right)$, until all agents do not wish to publish any more blocks (lines 7 – 14).

Note. *Similarly to the work by Eyal and Sirer [25] and Arnosti and Weinberg [54], the model rounds represent logical state changes, contrary to time-based rounds [55], [22]. Block publication includes the publication loop (lines 9 – 14) to facilitate strategic-block-release behaviors [25], [26], [41]. As in Eyal and Sirer [25], Sapirshtein et al. [26] and Nayak et al. [41], our model does not include spontaneous forks.*

E. Block Creation as a Game

The model gives rise to a game, played for the duration of a single epoch k. The players are the agents, with their available budgets as inputs.

We define the *expected income* of player i in epoch ℓ as her expected cryptocurrency holdings with the conclusion of said epoch (i.e., when

$\mathit{length}\left(\mathit{CP}\left(S^g\right)\right) = \ell\left(k+1\right)$):

$$\mathit{ExpInc}_i\left(k\right) = \mathbb{E}\left[\mathit{Bal}_i\left(\mathit{CP}\left(S^g\right)\right)\right]. \tag{1}$$

We model the expected income as the utility of player i, which she tries to maximize.

We analyze the system in equilibrium, where all players participate and the total profit is zero [25], [27], [56], [28], [57]. Accordingly, the sum of all agent expected incomes is as the total agent expenses:

$$B_{\mathcal{A}}\left(k\right) = \sum_{j=1}^{n} \mathit{ExpInc}_j\left(k\right), \tag{2}$$

and we normalize such that

$$\ell r_k = 1. \tag{3}$$

The player strategy space includes choosing the budget apportion ratio, what blocks to generate, and when to publish them. Strategy implementations are instantiations of $\mathsf{Apportion}\left(S^g, B\left(k\right)\right)$, $\mathsf{Generate}^{\Pi}\left(S^g, S^l\right)$ and $\mathsf{Publish}\left(S^g, S^l\right)$.

IV. Nakamoto Protocol

To make those definitions concrete, we now instantiate a classical Nakamoto blockchain protocol like Bitcoin [1], denoted ℓ-*NB*, in our model. Note that specifically Bitcoin utilizes a single-block epochs, that is 1-*NB*.

This definition both exemplifies our model and used throughout the work for comparison with our proposed protocols. The balance function of ℓ-*NB* awards each agent i with a relative reward equal to her relative contributed blocks in the epoch k. Hence, the balance of each agent $i \in 1, ..., n$ with the conclusion of epoch k is $\mathit{Bal}_i^{\ell\text{-}\mathit{NB}}\left(\mathit{CP}\left(S^g\right)\right) = \frac{N_i^k\left(\mathit{CP}\left(S^g\right)\right)}{\sum_{j=1}^{n} N_j^k\left(\mathit{CP}\left(S^g\right)\right)} \ell r_k$ coins, and the total number of minted coins in the epoch is exactly ℓr_k.

The prescribed equilibrium strategy $\sigma_{\mathit{desired}}^{\ell\text{-}\mathit{NB}}$ (Algorithm 2) is such that each agent i apportions her budget $B_i^{ec}\left(k\right) = B_i\left(k\right)$ and $B_i^{ic}\left(k\right) = 0$, points created blocks to $\mathit{last}\left(\mathit{CP}\left(S^g\right)\right)$, and publish them immediately. In case of conflicting longest chains, $\sigma_{\mathit{desired}}^{\ell\text{-}\mathit{NB}}$ states that the agent points her next block to either of them picked uniformly-at-random.

Note. *Bitcoin [1] defines a different tie-breaking rule — pick the first chain that the agent became aware of. That variation assures different security guarantees based on the underlying network assumptions. As in previous work [58], [59] we avoid such assumptions by using the uniformly-at-random variation.*

Algorithm 2: $\sigma_{\mathit{desired}}^{\ell\text{-}\mathit{NB}}$ of agent i

```
Function Apportion(S^g, B_i(k)):
    return ⟨0, B_i(k)⟩

Function Generate(S^g, S_i^l):
    C ← uniformly at random from LC(S^g)
    pointer ← last(C)
    return NewBlock(pointer)

Function Publish(S^g, S_i^l):
    return All previously unpublished blocks
```

Assume all agents follow $\sigma_{\mathit{desired}}^{\ell\text{-}\mathit{NB}}$. It follows that there is a single longest chain ($|\mathit{LC}\left(S^g\right)| = 1$) that is also its longest common prefix, which we denote as $C = \mathit{CP}\left(S^g\right)$. By the design of the scheduler the number of blocks an agent i creates in an epoch k follows the binomial distribution parameterized with the epoch length and her relative external budget, that is $N_i^k\left(C\right) \sim \mathrm{Bin}\left(\ell, b_i\left(k\right)\right)$. Consequently, it holds that $\mathbb{E}\left[N_i^k\left(C\right)\right] = \ell b_i^{ec}\left(k\right)$, and based on Eq. 3 and 1 we get

$$\mathit{ExpInc}_i^{\ell\text{-}\mathit{NB}}\left(k\right) = b_i^{ec}\left(k\right), \tag{4}$$

and by Eq. 2 we get $B_{\mathcal{A}}\left(k\right) = B_{\mathcal{A}}^{ec}\left(k\right) = 1$.

V. Cryptocurrency Evaluation Metrics

We now define metrics for evaluating cryptocurrency systems. We then use these metrics to reason about our proposed protocols, namely with contrast to ℓ-*NB*. Typically, we assume that for a protocol Π all agents follow the prescribed strategy $\sigma_{\mathit{desired}}^{\Pi}$, and examine different system aspects derived from that. We can classify our metrics under the following categories.

First, we consider the *incentive compatibility* of a system. We evaluate profit from coalescing (§V-A), and assess profitable deviations from $\sigma_{\mathit{desired}}^{\Pi}$ (§V-B). Both of these properties are in the center of a major research effort [25], [26], [41], [28], [42], [43], [44], [45], [29], [46], [60], [61].

We move to quantify the *required attack costs* for succeeding in such operation. Namely, we focus on *double-spend* attacks [62], [63], [64], [65], [66], where the attacker requires (§V-C) and forfeits (§V-D) her block reward in case of a successful attack.

We conclude by evaluating the permissiveness (§V-E) and the *environmental impact* (§V-F) of the system.

A. size-indifference

Cryptocurrency systems rely their security on the assumption that there are multiple distinctive agents

whom none has a substantial control over the system. For that, these systems strive to distribute their rewards in a way that is size-indifferent, meaning that agents do not increase their relative gainings disproportionately by coalescing.

The metric *size-indifference* measures how a protocol satisfies this desideratum. Formally, assume each agent $i \in 1, ..., n$ with relative budget $b_i(k)$ follows $\sigma^{\Pi}_{desired}$. The expected income of such player is $ExpInc^{\Pi}_i(k)$, and we denote $\delta_i \triangleq \left| b_i(k) - \frac{ExpInc^{\Pi}_i(k)}{\sum_{j=1}^{n} ExpInc^{\Pi}_i(k)} \right|$. Then we define

$$\textit{size-indifference} = \max_{i \in 1,...,n} \delta_i \ . \tag{5}$$

We want *size-indifference* to minimal, for as it grows agents get less than their proportional share. Preferably, *size-indifference* $= 0$ indicates all agents get reward proportional to their budget, suggesting that coalescing does not yield an increase in the expected reward.

Example. *For ℓ-NB it holds that size-indifference $= 0$ as shown in [1]. Eq. 4 also yields this result.*

B. coalition-resistance

Recall protocol Π provides a prescribed strategy $\sigma^{\Pi}_{desired}$ that agents individually choose whether to follow or not. The metric *coalition-resistance* bounds from above the required relative budget $b_i(k)$ of an agent i such that $\sigma^{\Pi}_{desired}$ is her best-response strategy.

Formally, let σ^{Π}_{br} denote the best-response strategy of agent i with relative budget $b_i(k)$ when all other agents follow $\sigma^{\Pi}_{desired}$. *coalition-resistance* is the maximal value $b_i(k)$ such that $\sigma^{\Pi}_{br} = \sigma^{\Pi}_{desired}$. It follows that $\sigma^{\Pi}_{desired}$ is a Nash-equilibrium strategy if all agent relative budgets are not greater than *coalition-resistance*.

Example. *Sapirshtein et al. [26] showed that this bound relies on network assumptions, and in the comparable case of uniform tie-breaking values as coalition-resistance $= 0.232$.*

C. free-safety-violation-threshold

In a *double-spend* attack [62], [63], [64], [65], [66] the attacker tries to create an alternative longest chain, nullifying a payment on the original chain, presumably after she already received the goods for that payment. To mount this attack in a *ℓ-NB* cryptocurrency the attacker is required to spend resources to create the alternative chain, yet she is compensated for these expense with her earned block rewards if the attack is successful. As such, there is a required resources threshold to mount this attack, but once met, the attack fully compensates the attacker for the said expended resources.

The metric *free-safety-violation-threshold* measures the minimal required resources for an external party in *ec* to deploy such a double-spend attack on the system, assuming all current agents follow the prescribed strategy. Note the attacker may rent vast computational resources for a short period of time [64] or a moderate amount for longer periods. We instead measure the cost to create a single block.

Formally, assume all agents follow $\sigma^{\Pi}_{desired}$ and let d be the number of blocks an external attacker needs to create to surpass the original chain. *free-safety-violation-threshold* is the minimal cost in *ec* to create d blocks that guarantee the agent with block reward of at least dr_k coins.

Example. *In ℓ-NB the reward for each block is $r_k = \frac{1}{\ell}$ in ec, and in equilibrium that is also the cost to create a block. To create d blocks the attacker is required to spend that amount d times, hence free-safety-violation-threshold $= \frac{d}{\ell}$.*

D. safety-violation-threshold

The metric *safety-violation-threshold* highly resembles *free-safety-violation-threshold*, but differs regarding the requirement for compensation via the block reward. That is, *safety-violation-threshold* is the required cost to mount a double-spend attack on the system where the attacker does not expect to be fully rewarded for her blocks. Note such attacks are less likely with the absence of an explicit motive, especially compared to the self-sustaining alternative.

Formally, assume all agents follow $\sigma^{\Pi}_{desired}$ and let d be the number of blocks the attacker needs to create to surpass the original chain. *safety-violation-threshold* is the minimal cost in *ec* to create d blocks, regardless of the expected block reward the attacker might receive due to these blocks.

Example. *In ℓ-NB all blocks produce the same reward, hence an agent cannot reduce the cost for a double-spend attack by opting for creating less-rewarding blocks, yielding, similarly, safety-violation-threshold $= \frac{d}{\ell}$.*

E. permissiveness

Cryptocurrency systems may require agents to own *ic* as a condition to create blocks. That is, for instance, the core principal of proof-of-stake systems [22], [23], [36].

As these systems operate in the permissionless setting, supposedly any agent should be able to join the system without the consent of other agents.

However, obtaining *ic* requires updating the new coin ownership in the system state, which requires the authorization of already-present system agents. Therefore, the existing agents can prevent new agents from obtaining *ic*, depriving the permissionless stipulation. Note that different systems that require *ic* possessions may opt in a middle ground where agents that do not own *ic* get relatively lower reward.

Therefore, *permissiveness* measures the permissiveness of the system through its reward ratios when agents fail to obtain *ic*.

Formally, assume an agent i with budget $B_i(k)$, and that all other agents follow $\sigma^{\Pi}_{desired}$. Denote $\sigma^{\Pi}_{no\text{-}ic}$ as $\sigma^{\Pi}_{desired}$ with the exception that her chosen implementation of $\mathsf{Apportion}_i(S^g, B_i(k))$ returns $\langle 0, B_i(k)\rangle$. Denote $ExpInc_i^{no\text{-}ic}(k)$ and $ExpInc_i^{desired}(k)$ the expected income of agent i should she follow $\sigma^{\Pi}_{no\text{-}ic}$ and $\sigma^{\Pi}_{desired}$, respectively. We define *permissiveness* as

$$permissiveness = \frac{ExpInc_i^{no\text{-}ic}(k)}{ExpInc_i^{desired}(k)} \ . \qquad (6)$$

When $permissiveness = 1$ it follows that an agent's expected income is not effected by her inability to obtain *ic*, thus Π achieves its permissionless objective. However, $permissiveness = 0$ indicates that agents unable to obtain *ic* are completely prevented from participation.

Example. *In ℓ-NB the strategies $\sigma^{\Pi}_{no\text{-}ic}$ and $\sigma^{\Pi}_{desired}$ are the same, hence $ExpInc_i^{no\text{-}ic}(k) = ExpInc_i^{desired}(k)$ and $permissiveness = 1$.*

F. external-expenses

external-expenses evaluates the environmental impact of a protocol Π. Lower values of *external-expenses* indicate a desired lower impact.

Formally, *external-expenses* measures in *ec* the external expenses of all agents in an epoch when all of them follow $\sigma_{desired}$. That is, $external\text{-}expenses = B^{ec}_{\mathcal{A}}(k)$.

Example. *For ℓ-NB it holds that $B^{ec}_{\mathcal{A}}(k) = 1$ and as such $external\text{-}expenses = 1$.*

VI. (ℓ, α)-PR Protocol

We now present (ℓ, α)-*PR*, standing for *Partial Reward*, a cryptocurrency protocol achieving lower environmental impact of ℓ-*NB*. In a nutshell, it mints coins as ℓ-*NB*, but distributes some of these coins among all system users, and not just the block-creating miners. As the coin minting rate is identical so is the monetary value of each, resulting in lower monetary reward for the participating agents. Consequently, in an equilibrium the external expenditure is desirably lower as well. Defectively, it renders safety violations less requiring.

The rest of this section is organized as follows. We start by presenting the (ℓ, α)-*PR* specifics (§VI-A), and analyzing it assuming all agents follow the prescribed strategy (§VI-B). We then use our metrics to evaluate (ℓ, α)-*PR*, contrasting it with ℓ-*NB* (§VI-C).

A. (ℓ, α)-PR Description

(ℓ, α)-*PR* mints ℓr_k new coins with each epoch conclusion. Set with a parameter $\alpha \in [0, 1)$, it distributes only $(1-\alpha)\,\ell r_k$ of the newly-minted coins to agents, while the rest of $\alpha \ell r_k$ are distributed among all system participants (i.e., including the passive clients) based on their *ic* holdings at the epoch beginning. The prescribed equilibrium strategy $\sigma^{(\ell,\alpha)\text{-}PR}_{desired}$ is as of ℓ-*NB* (Algorithm 2).

The balance of each agent $i \in 1, ..., n$ with epoch k conclusion is

$$Bal_i^{(\ell,\alpha)\text{-}PR}(CP(S^g)) = (1-\alpha)\frac{N_i^k(CP(S^g))}{\sum_{j=1}^{n} N_j^k(CP(S^g))}\ell r_k + \alpha\frac{B_i^{ic}(k)}{B_{\mathcal{A}}^{ic}(k) + B_{\mathcal{C}}^{ic}(k)}\ell r_k \ . \qquad (7)$$

Note that in (ℓ, α)-*PR* the passive clients also increase their internal currency possessions by $\alpha\frac{B_{\mathcal{C}}^{ic}(k)}{B_{\mathcal{A}}^{ic}(k)+B_{\mathcal{C}}^{ic}(k)}\ell r_k$ coins, and as stated, the number of minted coins in the epoch is exactly ℓr_k. Note that $(\ell, 0)$-*PR* is exactly ℓ-*NB*, as agents expend all their budgets externaly and receive the entirety of the reward with the epoch conclusion.

B. (ℓ, α)-PR prescribed strategy analysis

Assume all agents follow $\sigma^{(\ell,\alpha)\text{-}PR}_{desired}$. Hence agents extend only the longest chain, which is also the longest common prefix. We denote this chain $C = CP(S^g)$.

By $\sigma^{(\ell,\alpha)\text{-}PR}_{desired}$ each agent $i \in 1, ..., n$ apportions such that $B_i^{ic}(k) = 0$, hence her expected income is $ExpInc_i^{(\ell,\alpha)\text{-}PR}(k) = (1-\alpha)\frac{\mathbb{E}[N_i^k(C)]}{\sum_{j=1}^{n}\mathbb{E}[N_j^k(C)]}\ell r_k + \alpha\frac{B_i^{ic}(k)}{B_{\mathcal{A}}^{ic}(k)+B_{\mathcal{C}}^{ic}(k)}\ell r_k$. Summing for all agents and applying Eq. 2 and 3 yields $B_{\mathcal{A}}(k) = 1 - \alpha + \frac{\alpha B_{\mathcal{A}}(k)}{B_{\mathcal{A}}(k)+B_{\mathcal{C}}^{ic}(k)}$. However, recall that $B_{\mathcal{A}}(k) \ll B_{\mathcal{C}}^{ic}(k)$ and $\alpha < 1$, and as such $\frac{\alpha B_{\mathcal{A}}(k)}{B_{\mathcal{A}}(k)+B_{\mathcal{C}}^{ic}(k)}$ is negligible.

We conclude that $B_{\mathcal{A}}(k) = B^{ec}_{\mathcal{A}}(k) = 1 - \alpha$ and the cost to create each single block is $\frac{1-\alpha}{\ell}$.

C. (ℓ, α) -PR Evaluation

We now use our metrics (§V) to evaluate (ℓ, α) -*PR*, detailed throughout the following section. The decreased block rewards agent receive leads to the desired decrease in the system external expenses. However, it also decreases the required costs for performing both indicated types of safety violations. ℓ-*NB* and (ℓ, α) -*PR* equal at the other metrics.

1) size-indifference: For each agent $i \in 1, ..., n$ it holds that $\mathbb{E}\left[N_i^k(C)\right] = \ell b_i^{ec}(k)$ and therefore the expected income of each player is $ExpInc_i^{(\ell,\alpha)\text{-}PR}(k) = (1-\alpha)\, b_i^{ec}(k)$. It also holds that $b_i^{ec}(k) = b_i(k)$ and $\frac{ExpInc_i(k)}{\sum_{j=1}^{n} ExpInc_j(k)} = b_i^{ec}(k)$, resulting with *size-indifference* $= 0$.

2) coalition-resistance: An agent i may decide to apportion her budget such that $B_i^{ic}(k) > 0$ and $B_i^{ec}(k) < B_i(k)$. A reason for that would be to increase her expected income with the $\alpha\ell r_k$ component distributed among all system participants. However, recall that $B_i^{ic}(k) < B_{\mathcal{A}}(k)$ and $B_{\mathcal{A}}(k) \ll B_{\mathcal{C}}^{ic}(k)$, rendering any potential reward negligible. Moreover, by apportioning $B_i^{ec}(k) < B_i(k)$ agent i decreases her expected number of blocks $\mathbb{E}\left[N_i^k(C)\right]$, decreasing her expected income.

Therefore, any possible strategy deviations of (ℓ, α) -*PR* are as of (ℓ, α) -*PR*, resulting with *coalition-resistance* $= 0.232$ [26].

3) free-safety-violation-threshold: The attacker is required to create d blocks, and the cost to create each block is $\frac{1-\alpha}{\ell}$, and therefore *safety-violation-threshold* $= \frac{(1-\alpha)d}{\ell}$.

4) safety-violation-threshold: For the same considerations as *free-safety-violation-threshold* it holds that *safety-violation-threshold* $= \frac{(1-\alpha)d}{\ell}$.

5) permissiveness: In (ℓ, α) -*PR* new agents do not necessitate internal currency to participate, and expected income is practically unaffected by it. We conclude that $ExpInc_i^{no\text{-}ic}(k) = ExpInc_i^{desired}(k)$ and as such *permissiveness* $= 1$.

6) external-expenses: In (ℓ, α) -*PR*, agents are only rewarded $1-\alpha$ of the block rewards, and in equilibrium these are their expenses. All these expenses are external, thus *external-expenses* $= 1 - \alpha$.

VII. (ℓ, α, F) -IIE Protocol

We now present (ℓ, α, F) -*IIE*, standing for *Incentivized Internal Expenses*, our second protocol designed to achieve lower *external-expenses* compared to ℓ-*NB*. It is also designed to maintain the required resource threshold for free safety violations. However, its parameter values includes inherent trade-offs, where perfecting by one metric comes at the expense of another.

Recall (ℓ, α) -*PR* worsened its resiliency as it lowered the required resources to create a single block. Hence, to uphold the original level, a key principal of (ℓ, α, F) -*IIE* is to keep said resources at their original level. Considering the prerequisite for lower external expenses leads to the primary design concept of (ℓ, α, F) -*IIE* — get agents to internally, rather than externally, spend their budgets.

We begin by presenting a straw-man protocol, forcing agents to do just that (§VII-A). We briefly state its evident shortcomings, paving the way for detailing (ℓ, α, F) -*IIE* (§VII-B). We then analyze (ℓ, α, F) -*IIE* when all agents follow the desired strategy (§VII-C), and evaluate by the presented metrics (§VII-D). We conclude by summarizing (ℓ, α, F) -*IIE*.

A. Straw-man Protocol

We suggest the following straw-man protocol, *forcing* agents to internally spend. As in ℓ-*NB*, it mints ℓr_k coins with each epoch and distributes them solely among the agents, based on their relative block contributions to the main chain. Resembling (ℓ, α) -*PR*, the straw-man protocol also includes a parameter $\alpha \in [0, 1)$. However, it requires that for each block in epoch k an agent i creates on $CP(S^g)$ she must have expended αr_k of *ic* at the epoch beginning. The straw-man protocol distributes the expended *ic* among all system participants based on their *ic* holdings at the epoch beginning.

The balance of each agent i with the epoch conclusion is $Bal_i^{\text{straw-man}}(CP(S^g)) = \frac{N_i^k(C)}{\sum_{j=1}^{n} N_j^k(C)} \ell r_k + \frac{B_i^{ic}(k)}{B_{\mathcal{A}}^{ic}(k) + B_{\mathcal{C}}^{ic}(k)} B_{\mathcal{A}}^{ic}(k)$. The second summed element resembles that of (ℓ, α) -*PR*'s Balance function (Eq. 7), however, their sources are fundamentally different. For the straw-man protocol it originates from the internal resource redistribution of all agents, while in (ℓ, α) -*PR* it is the relevant portion of the newly-minted currency.

This protocol utilizes epochs to force agents to spend their *ic* in advance, and for long periods of time. If that was not the case, then agents could spend αr_k coins per block they were hoping to create. Aside any potential practical overhead of such excessive commits, this scheme also highly favors agents with greater relative budgets, as they commit lower percentile of their budget, increasing their chance to create blocks unevenly.

Algorithm 3: $\sigma^{\text{straw-man}}_{desired}$ of agent i

Function `Apportion`$(S^g,\ B_i(k))$**:**
  return $\langle \alpha B_i(k), (1-\alpha) B_i(k) \rangle$

Function `Generate`$(S^g,\ S_i^l)$**:**
  $C \leftarrow$ uniformly at random from $LC(S^g)$
  pointer $\leftarrow$ *last* (C)
  return NewBlock(pointer)

Function `Publish`$(S^g,\ S_i^l)$**:**
  return All previously unpublished blocks

The desired strategy $\sigma^{\text{straw-man}}_{desired}$ resembles that of $\sigma^{\ell\text{-NB}}_{desired}$, aside from suggesting an agent i should apportion her budget $\langle \alpha B_i(k), (1-\alpha) B_i(k) \rangle$. We present $\sigma^{\text{straw-man}}_{desired}$ in Algorithm 3.

However, this protocol suffer from two inherent, intolerable shortcomings. First, agents entail to obtain *ic* for block creation, rendering the protocol permissioned. Moreover, in equilibrium, agents are expected to commit *ic* sufficient to enable the creation of exactly the number of blocks they expect. Now, assume an agent becomes absent, either maliciously or unintentionally. Other agents cannot create new blocks exceeding their quota, and the system halts.

These problems arise as the protocol conditions block creation on early *ic* expenditure. Hence, we present (ℓ, α, F) *-IIE*, *incentivizing, but not forcing,* agents to internally spend.

B. (ℓ, α, F) *-IIE Description*

We now lay out the specifics of (ℓ, α, F) *-IIE*. It includes two types of blocks, *regular* and *factored*, and the creating agent determines the block's type at block generation. Block types are immutable, that is, to alter a block's type an agent needs to re-create the block. An agent i can always create a regular block when invoked with $\mathsf{Generate}_i^{\Pi}\left(S^g, S_i^l\right)$. Contrarily, an agent may create up to $\left\lfloor \frac{B_i^{ic}(k)}{\alpha r_k} \right\rfloor$ factored blocks in an epoch on chain C. (ℓ, α, F) *-IIE* assigns a *score* to each block, and factored and regular blocks have scores of $F \in \mathbb{R}_{>1}$ and 1, respectively.

a) Reward distribution: (ℓ, α, F) *-IIE* mints ℓr_k coins in epoch k, and shares them among solely among the agents.

Denote by $N_i^k(C)$ the number of factored blocks in the current epoch created by agent i on chain C. Denote by $S_i(C)$ the total score of blocks created in the current epoch by agent i on chain C.

(ℓ, α, F) *-IIE* distributes the newly-minted ℓr_k coins among all agents, based on their total block score in the epoch. (ℓ, α, F) *-IIE* distributes the internal expenses $B^{ic}_{\mathcal{A}}(k)$ among all system participants based on their *ic* possessions at the epoch beginning.

Algorithm 4: $\sigma^{(\ell,\alpha,F)\text{-IIE}}_{desired}$ of agent i

Function `Apportion`$(S^g,\ B_i(k))$**:**
  return $\langle \alpha B_i(k), (1-\alpha) B_i(k) \rangle$

Function `Generate`$(S^g,\ S_i^l)$**:**
  $C \leftarrow$ uniformly at random from $LC(S^g)$
  pointer $\leftarrow$ *last* (C)
  if $N_i^k(C) < \left\lfloor \frac{B_i^{ic}(k)}{\alpha r_k} \right\rfloor$ **then**
    factored $\leftarrow$ true
  else
    factored $\leftarrow$ false
  return NewBlock(pointer, factored)

Function `Publish`$(S^g,\ S_i^l)$**:**
  return All previously unpublished blocks

Therefore, the balance of each agent $i \in 1, ..., n$ with the epoch conclusion is

$$Bal_i^{(\ell,\alpha,F)\text{-IIE}}\left(CP\left(S^g\right)\right) = \frac{S_i\left(CP\left(S^g\right)\right)}{\sum_{j=1}^{n} S_j\left(CP\left(S^g\right)\right)} \ell r_k + \frac{B_i^{ic}(k)}{B^{ic}_{\mathcal{A}}(k) + B^{ic}_{\mathcal{C}}(k)} B^{ic}_{\mathcal{A}}(k) \quad . \quad (8)$$

Note. *When F=1 or α=0 there is no incentive to spend ic, and thus $(\ell, 0, 1)$ -IIE identifies with ℓ-NB.*

b) Prescribed strategy: The prescribed strategy $\sigma^{(\ell,\alpha,F)\text{-IIE}}_{desired}$ defines that an agent i apportions her budget such that $B_i^{ec}(k) = (1-\alpha) B_i(k)$ and $B_i^{ic}(k) = \alpha B_i(k)$, points her created blocks to a uniformly-at-random selected chain from $LC(S^g)$ and sets the first $\left\lfloor \frac{B_i^{ic}(k)}{\alpha r_k} \right\rfloor$ created blocks as factored. In addition, agent i publishes any block she creates immediately. Algorithm 4 presents $\sigma^{(\ell,\alpha,F)\text{-IIE}}_{desired}$.

C. (ℓ, α, F) *-IIE prescribed strategy analysis*

We now derive the expected score $\mathbb{E}[S_i]$ of agent i assuming all agents follow $\sigma^{(\ell,\alpha,F)\text{-IIE}}_{desired}$. As before, agents extend only the longest chain, resulting with it being the longest common prefix, denoted C.

By $\sigma^{(\ell,\alpha,F)\text{-IIE}}_{desired}$ it holds for all $i \in 1, ..., n$: $B_i^{ec}(k) = (1-\alpha) B_i(k)$ and therefore $b_i^{ec}(k) = \frac{B_i^{ec}(k)}{B^{ec}_{\mathcal{A}}(k)} = \frac{(1-\alpha)B_i(k)}{(1-\alpha)B_{\mathcal{A}}(k)} = b_i(k)$.

By definition of $N_i^k(C)$, we get that it is a random variable indicating the number of blocks agent i creates on chain C. As the scheduler picks an agent based on their relative external budgets, it follows that $N_i^k(C) \sim \text{Bin}(\ell, b_i(k))$. Therefore, $\Pr\left(N_i^k(C) = m\right) = \binom{\ell}{m} \cdot (b_i(k))^m \cdot (1 - b_i(k))^{\ell - m}$ and $\mathbb{E}\left[N_i^k(C)\right] = \ell b_i(k)$.

According to $\sigma^{(\ell,\alpha,F)\text{-IIE}}_{desired}$, agent i sets $B_i^{ic}(k) = \alpha B_i(k)$ and therefore can create at most $\left\lfloor \frac{B_i(k)}{r_k} \right\rfloor =$

$\lfloor \ell B_i(k) \rfloor$ factored blocks. For better readability, from this point onward we assume that $\ell b_i(k) \in \mathbb{N}$.

Agent i's score assuming $N_i^k(C) = m$ blocks is $S_i\left(C|N_i^k(C)=m\right) =$

$$\begin{cases} mF, & 0 \le m \le \ell B_i(k) \\ \ell B_i(k) F + m - \ell B_i(k) & \ell B_i(k) < m \le \ell \end{cases},$$

and hence her expected score is $\mathbb{E}[S_i(C)] = \sum_{m=0}^{\ell} \Pr\left(N_i^k(C)=m\right) S_i\left(C|N_i^k(C)=m\right)$, resulting with $ExpInc_i^{(\ell,\alpha,F)\text{-}IIE}(k) = \frac{\mathbb{E}[S_i(C)]}{\sum_{j=1}^{n}\mathbb{E}[S_j(C)]}\ell r_k + \frac{\alpha B_i(k)}{\alpha B_{\mathcal{A}}(k)+B_{\mathcal{C}}^{ic}(k)}\alpha B_{\mathcal{A}}(k)$.

Summing for all agents and applying Eq. 2 and 3 yields $B_{\mathcal{A}}(k) = 1 + \frac{(\alpha B_{\mathcal{A}}(k))^2}{\alpha B_{\mathcal{A}}(k)+B_{\mathcal{C}}^{ic}(k)}$, leading to $B_{\mathcal{A}}(k)\left(1 - \frac{\alpha^2 B_{\mathcal{A}}(k)}{\alpha B_{\mathcal{A}}(k)+B_{\mathcal{C}}^{ic}(k)}\right) = 1$. Recall that $\alpha < 1$ and $B_{\mathcal{A}}(k) \ll B_{\mathcal{C}}^{ic}(k)$, therefore $\alpha^2 B_{\mathcal{A}}(k) \ll B_{\mathcal{C}}^{ic}(k)$ and $\frac{\alpha^2 B_{\mathcal{A}}(k)}{\alpha B_{\mathcal{A}}(k)+B_{\mathcal{C}}^{ic}(k)}$ is negligible. We get that $B_{\mathcal{A}}(k) = 1$ and a single block's creation cost is $\frac{1}{\ell}$.

D. (ℓ, α, F) -IIE Evaluation

In this section we analyze (ℓ, α, F) *-IIE* with respect to the evaluation metrics. A summary is given below and a more detailed analysis follows.

We show how *size-indifference* improves with longer epochs ℓ and lower factored blocks score F. It evaluates the same for any α value, yet is affected by the budget distribution among agents.

We use MDP to analyze *coalition-resistance*, showing that sufficiently-large F values approximate the evaluation results of ℓ*-NB*. We also show analytically that $\frac{1-\alpha}{2-\alpha}$ upper-bounds *coalition-resistance*.

We show that the resources threshold for free safety violations, *free-safety-violation-threshold*, is as of (ℓ, α) *-PR*, successfully meeting a design goal of (ℓ, α, F) *-IIE*. However, *safety-violation-threshold* is as of (ℓ, α) *-PR*, which is sub-optimal, but tolerable.

The analysis also shows that *permissiveness* improves with lower F values, and *external-expenses* is desirably as of (ℓ, α) *-PR*.

1) size-indifference: We devote this section to evaluate *size-indifference* of (ℓ, α, F) *-IIE*. We begin by analytically analyzing how the system parameters, namely ℓ and F affect *size-indifference*. We note that α has negligible effect on the expected income of an agent (Eq. 8). We show that for sufficiently large values of ℓ the protocol achieves *size-indifference* $= 0$, and that lower values of F also result in lower *size-indifference* (§VII-D1a).

We then present the affect of agent budget distribution on *size-indifference*. This is a unique phenomena for (ℓ, α, F) *-IIE* that we discuss and quantify with numerical examples (§VII-D1b).

a) ℓ and F values: For ease of writing we define the normalized expected score of player i to be $nes_i = \frac{\mathbb{E}[S_i]}{\ell b_i(k) F}$. We begin with presenting two lemmas, stating necessary and sufficient conditions for *size-indifference* $= 0$:

Lemma 1. *Iff* $\forall i, j \in 1, ..., n : nes_i = nes_j \neq 0$ *then size-indifference* $= 0$.

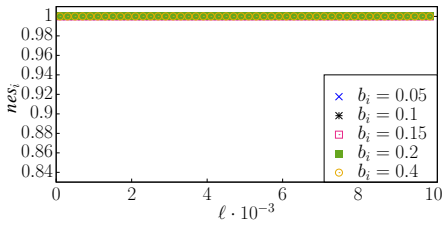
Lemma 2. *For each agent* $i : \lim_{\ell\to\infty} nes_i \xrightarrow{a.s.} 1$.

We bring both proofs in Appendix A. Both lemmas lead to the following corollary:

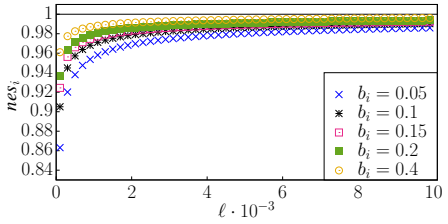
Corollary. *If* $\lim_{\ell\to\infty}$ *then size-indifference* $= 0$.

That is, for sufficiently large values of ℓ, (i.e., sufficiently long epochs) (ℓ, α, F) *-IIE* achieves *size-indifference* $= 0$.

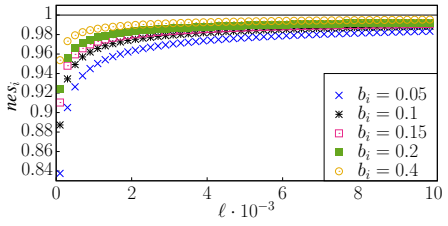
b) Budget Distribution: We now evaluate *size-indifference* for different values of ℓ, F and $b(k)$. For fixed F and $b(k)$ values, we numerically calculate and plot nes_i as a function of ℓ. We present our results in Fig. 1.



(a) $F = 1$.



(b) $F = 5$.



(c) $F = 20$.

Fig. 1: nes_i as a function of an epoch block count ℓ.

Fig. 1 confirms Lemma 2 — for all values of $F, b_i(k)$, nes_i approaches 1 as ℓ grows. It also shows

Agent Relative Budgets				δ_1	δ_2
$b_2(k)$	$b_3(k)$	$b_4(k)$	$b_5(k)$		
0.4	0.4	-	-	0.001	0.000
0.8	-	-	-	0.002	0.002
0.6	0.2	-	-	0.001	0.002
0.3	0.2	0.3	-	0.000	0.000
0.2	0.2	0.2	0.2	0.000	0.000

TABLE I: Budget distribution and *size-indifference* for $b_1(k) = 0.2$, $\ell = 2500$, $\alpha = 0.5$ and $F = 5$.

the effect of F value. As expected, when $F = 1$ then $\mathbb{E}[S_i] = \ell b_i(k)$ and $nes_i = 1$ for all ℓ values (Fig. 1a). However, increasing the values of F and $b_i(k)$ decrease nes_i (Figures 1b,1c).

However, Fig. 1 presents an undesired effect that (ℓ, α, F) *-IIE* bares — for fixed ℓ, F agents with different relative budgets have different *nes*, which by Lemma 1 leads to *size-indifference* > 0.

This phenomena does not occur in ℓ*-NB* nor (ℓ, α) *-PR*, and its origin is the possibly-different score each block has. Intuitively, agents with lower $b(k)$ suffer a greater decrease in their score when not creating exactly their expected number of blocks, resulting in a lower *nes*.

We dedicate the rest of this section to discuss how different relative budgets affect agent expected incomes and *size-indifference*. We consider following arbitrarily-constructed setting of $n = 5$ agents, with epochs of $\ell = 2500$ blocks and $F = 5$. We fix $b_1(k) = 0.2$ and consider different budget distributions of agents 2 to 5, detailed in Table I. For each such budget distribution we numerically calculate $\forall\ i \in 1, ..., n : \delta_i$, and present those values for $i = 1, 2$. We also present the maximal value indicating *size-indifference*.

Table I shows that higher variance in budget distribution results in higher *size-indifference*. For instance, consider the setting with only two players where $b_1(k) = 0.2$ and $b_2(k) = 0.8$. That is, agent 2 has a budget four times greater than that of player 1 This setting leads to the highest *size-indifference* $= 0.002$. Contrarily, the setting of five agents, all with equal relative budgets of $b(k) = 0.2$ results in *size-indifference* $= 0$.

Note. *The setting where* $b_1(k) = 0.2$ *and* $b_2(k) = 0.8$ *is unrealistic, presented only as an example for a highly-uneven budget distribution. Consider that even in that extreme scenario agent* 1 *has a relative expected income of* 0.198*, which is* 99% *of her relative budget.*

We summarize these results in an informal manner — unbalanced budget distribution increases deviation from optimal reward distribution, although such deviations are considerably minor even in the most extreme scenario examined. By increasing ℓ and decreasing F the system designer tune the system to nullify these deviations.

2) coalition-resistance: We search for sufficient conditions to assure that $\sigma_{br}^{(\ell,\alpha,F)\text{-}IIE}$ identifies with $\sigma_{desired}^{(\ell,\alpha,F)\text{-}IIE}$.

The strategy space of (ℓ, α, F) *-IIE* grows exceedingly with the introduction of block scores, and we utilize MDP [26], [29] to find $\sigma_{br}^{(\ell,\alpha,F)\text{-}IIE}$.

Our analysis shows that increasing F values or lowering α increases *coalition-resistance*. Specifically,$\ell = 10$, $F = 20$ and $\alpha = 0.5$ suffice to obtain *coalition-resistance* $= 0.2$. We differ the details to Appendix B.

We also present the close-form upper bound of $\frac{1-\alpha}{2-\alpha}$ for *coalition-resistance*. This is a loose bound, stating the susceptibility threshold to a *%51-attack* [1], [67]. We bring the analysis in Appendix B-A. Both of the analyses stand with the comparable cases of previous work [26].

3) free-safety-violation-threshold: In equilibria the total external expenses are $1-\alpha$ of total budgets, that is $B_{\mathcal{A}}^{ec}(k) = (1-\alpha) B_{\mathcal{A}}(k)$. As $B_{\mathcal{A}}(k) = \ell r_k = 1$ it follows that the required external expenses to (expectedly) create a single block is $\frac{1-\alpha}{\ell}$. An agent has to create factored blocks to be fully rewarded for them, and therefore each block requires spending additional $\frac{\alpha}{\ell}$ in *ic*. Hence, the cost to create a single block is $\frac{1}{\ell}$. The cost to create d blocks is just a multiplication of the cost of a single block, and therefore *free-safety-violation-threshold* $= \frac{d}{\ell}$.

4) safety-violation-threshold: We deduced the required external expenses to create a single block is $\frac{1-\alpha}{\ell}$. As an agent can choose not to create factored blocks, she bares no additional internal expenses for block creation. The cost to create d blocks is just a multiplication of the cost of a single block, and therefore *safety-violation-threshold* $= \frac{(1-\alpha)d}{\ell}$.

5) permissiveness: We now evaluate *permissiveness* of (ℓ, α, F) *-IIE*. The number of agents and their budget distribution heavily affects the expected income of players, and thus *permissiveness* as well. We exemplify an analysis for an arbitrarily-set system.

Assume there are two agents $1, 2$ with budgets $B_1(k), B_2(k)$, respectively. Assume agent 1 is cannot obtain *ic* due to lack of other agents' permission (and in this particular case, agent 2), and hence apportions her budget $B_1^{ic}(k) = 0, B_1^{ec}(k) = B_1(k)$.

Agent 2 apportions according to $\sigma_{desired}^{(\ell,\alpha,F)\text{-}IIE}$, that is $B_2^{ic}(k) = \alpha B_2(k), B_2^{ec}(k) = (1-\alpha) B_2(k)$.

It follows that $b_1^{ec}(k) = \frac{B_1(k)}{B_1(k)+(1-\alpha)B_2(k)}$ and $b_2^{ec}(k) = \frac{(1-\alpha)B_2(k)}{B_1(k)+(1-\alpha)B_2(k)}$. For simplicity we assume ℓ is sufficiently large and thus the agents create their expected number of blocks (see Lemma 2).

We now derive the expected score of both of the agents. Agent 1 can create normal blocks, and her expected number of blocks is $\frac{B_1(k)}{B_1(k)+(1-\alpha)B_2(k)}$, therefore $\mathbb{E}[S_1] = \ell\frac{B_1(k)}{B_1(k)+(1-\alpha)B_2(k)}$. For agent 2 it holds that $b_2^{ec}(k) < B_2^{ec}(k)$, and therefore she will create less factored blocks than her internal budget suffices, resulting in $\mathbb{E}[S_2] = \ell F\frac{(1-\alpha)B_2(k)}{B_1(k)+(1-\alpha)B_2(k)}$. By the expected income definition we get $ExpInc_1^{no\text{-}ic}(k) = \frac{B_1(k)}{B_1(k)+FB_2(k)}$.

Note. *This analysis does not consider a permission-withholding agent who alters her own apportion ratio, which we leave such analysis for future work.*

Following the same analysis for a system where agent 1 manages to obtain *ic* yields $ExpInc_1^{desired}(k) = \frac{B_1(k)}{B_1(k)+B_2(k)}$. Recall that $B_1(k)+B_2(k) = 1$ and $b_i(k) = B_i(k)$, leading to $permissiveness = \frac{1}{b_1(k)+F\left(1-b_1(k)\right)}$.

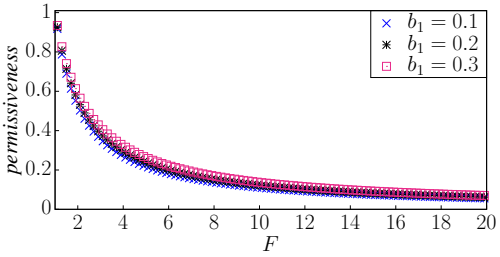


Fig. 2: *permissiveness* of (ℓ,α,F) *-IIE.*

We present *permissiveness* as a function of F in Fig. 2, for different values of $b_1(k)$. It shows that higher factor values F lead to lower values of *permissiveness*, tending the system towards being permissioned. It also shows that agents with higher relative budgets are less susceptible to these effects.

These results settle with intuition — higher values of F increase the reward of creating factored blocks, and failing to create such results in lower income. Higher relative budget enables creating more blocks, decreasing the overall score of the other agents, increasing her income despite the lack of *ic*.

6) external-expenses: As agents follow $\sigma_{desired}^{(\ell,\alpha,F)\text{-}IIE}$ then $B_{\mathcal{A}}^{ec}(k) = (1-\alpha) B_{\mathcal{A}}(k)$ and $external\text{-}expenses = 1-\alpha$.

VIII. Practical Implementation Considerations

We present a few implementation suggestions for (ℓ,α) *-PR* and (ℓ,α,F) *-IIE*. We suggest to include a *ramp up* period, in which the protocols perform as a classical Nakamoto protocol [1]. That allows enough currency to accumulate, justifying the negligible maintenance assumption.

We suggest using the transaction mechanism to make coins unavailable, by letting agents transact them to a null address, conceptually similar to proof-of-burn schemes [39], [40].

An implementation for the redistribution mechanism over a Turing-complete smart contracts blockchain [5] is more intuitive, for a simpler Bitcoin-like [1] blockchain we suggest utilizing the blockchain parsing mechanism, as follows. The agents derive the system state by parsing the main chain, updating their perceived balances of all system participants. When parsing an epoch concluding block, the system can include in the updated following state the redistributed coins as required. This resembles the way the Ethereum network nullified the DAO attack [68].

To avoid coin fragmentations due to precision errors we suggest the system enforcing a coin threshold, required for being included in the coin distribution. The system can also select a random subset of users to whom she distributes the coins. As in [1] we suggest using a difficulty mechanism, moderating block creation rate.

IX. Conclusion

We introduce a cryptocurrency model detailing internal and external currency relations, and formalize security, permissiveness, and environmental impact metrics. We generalize Nakamoto's blockchain design, allowing to accurately trade off security and ecological footprint reduction. We introduce two new protocols, both achieving reduced ecological footprint. The first reduces the monetary reward agents receive for block generation, while the other incentivizes agents to spend resources internally.

Reasoning about system security and ecological damage is an important step in widening the usability of cryptocurrency systems, and this work makes a step in that direction. These protocols may serve as the base for an update and design of both current and future cryptocurrency systems.

X. Acknowledgments

This research was supported by the Israel Science Foundation (grant No. 1641/18), Engima MPC Inc, and the Technion Hiroshi Fujiwara cyber-security research center.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," http://www.bitcoin.org/bitcoin.pdf, 2008.

[2] cryptoslate.com. (2019) Cryptocurrency market capitalizations. [Online]. Available: https://cryptoslate.com/coins/

[3] coinmarketcap.com. (2019) Cryptocurrency market capitalizations. [Online]. Available: https://coinmarketcap.com/

[4] blockchain.info, "Bitcoin market capitalization," http://blockchain.info/charts/market-cap, retrieved Oct. 2019.

[5] V. Buterin, "A next generation smart contract & decentralized application platform," https://www.ethereum.org/pdfs/EthereumWhitePaper.pdf/, 2013.

[6] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *Annual International Cryptology Conference*. Springer, 1992, pp. 139–147.

[7] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols," in *Secure Information Networks*, 1999.

[8] A. L. Goodkind, B. A. Jones, and R. P. Berrens, "Cryptodamages: Monetary value estimates of the air pollution and human health impacts of cryptocurrency mining," *Energy Research & Social Science*, 2020.

[9] Ethan Lou, "Bitcoin as big oil: the next big environmental fight?" https://www.theguardian.com/commentisfree/2019/jan/17/bitcoin-big-oil-environment-energy

[10] P. Giungato, R. Rana, A. Tarabella, and C. Tricase, "Current trends in sustainability of bitcoins and related blockchain technology," *Sustainability*, vol. 9, no. 12, 2017. [Online]. Available: https://www.mdpi.com/2071-1050/9/12/2214

[11] Alex Hern, "Bitcoins energy usage is huge we can't afford to ignore it," https://www.theguardian.com/technology/2018/jan/17/bitcoin-electricity-usage-huge-climate-cryptocurrency

[12] Maximilian Feige, "Bitcoin doesnt incentivize green energy," https://www.theblockcrypto.com/2019/01/30/bitcoin-doesnt-incentivize-green-energy

[13] Alan L. Johnson, "The negative environmental impact of bitcoin," https://sci-techmaven.io/superposition/society/the-negative-environmental-impact-of-bitcoin-y84bbL8VUi37CHHlsTw/, retrieved Jul. 2019.

[14] blockchain.info, "Bitcoin difficulty chart," https://www.blockchain.com/charts/difficulty?timespan=all, retrieved Jul. 2019.

[15] etherscan.info, "Ethereum difficulty chart," https://etherscan.io/chart/difficulty, retrieved Jul. 2019.

[16] C. C. for Alternative Finance. (2019) Cambridge bitcoin electricity consumption index. [Online]. Available: https://www.cbeci.org/comparisons/

[17] Digiconomist.net, "Bitcoin energy consumption index," https://digiconomist.net/bitcoin-energy-consumption, 2018.

[18] J. Tuwiner, "Bitcoin mining hardware," https://www.buybitcoinworldwide.com/mining/hardware/, 2017.

[19] wikipedia.com, "Electricity pricing," https://en.wikipedia.org/wiki/Electricity_pricing, retrieved Jul. 2019.

[20] digiconomist.net, "Bitcoin energy consumption index," https://digiconomist.net/bitcoin-energy-consumption.

[21] I. Bentov, P. Hubácek, T. Moran, and A. Nadler, "Tortoise and hares consensus: the meshcash framework for incentive-compatible, scalable cryptocurrencies." *IACR Cryptology ePrint Archive*, vol. 2017, p. 300, 2017.

[22] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual International Cryptology Conference*, 2017.

[23] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017.

[24] M. Baudet, A. Ching, A. Chursin, G. Danezis, F. Garillot, Z. Li, D. Malkhi, O. Naor, D. Perelman, and A. Sonnino, "State machine replication in the libra blockchain," 2018.

[25] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security*, 2014.

[26] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in Bitcoin," in *Financial Cryptography and Data Security*, 2016.

[27] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[28] I. Tsabary and I. Eyal, "The gap game," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018.

[29] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 3–16.

[30] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *Practical Byzantine fault tolerance*, 1999.

[31] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: speculative byzantine fault tolerance," in *ACM SIGOPS Operating Systems Review*, 2007.

[32] R. Kapitza, J. Behl, C. Cachin, T. Distler, S. Kuhnle, S. V. Mohammadi, W. Schröder-Preikschat, and K. Stengel, "Cheapbft: resource-efficient byzantine fault tolerance," in *Proceedings of the 7th ACM european conference on Computer Systems*. ACM, 2012, pp. 295–308.

[33] G. G. Gueta, I. Abraham, S. Grossman, D. Malkhi, B. Pinkas, M. K. Reiter, D.-A. Seredinschi, O. Tamir, and A. Tomescu, "Sbft: a scalable and decentralized trust infrastructure," *arXiv preprint arXiv:1804.01626*, 2018.

[34] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: Bft consensus with linearity and responsiveness," in *38th ACM symposium on Principles of Distributed Computing (PODC19)*, 2019.

[35] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Workshop on distributed cryptocurrencies and consensus ledgers*, vol. 310, 2016, p. 4.

[36] EOS. (2019) Eos.io technical white paper v2. [Online]. Available: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md

[37] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 2017, pp. 297–315.

[38] P. Gaži, A. Kiayias, and A. Russell, "Stake-bleeding attacks on proof-of-stake blockchains," in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, 2018.

[39] I. Stewart, "Proof of burn," *bitcoin.it*, 2012.

[40] P4Titan, "Slimcoin: A peer-to-peer crypto-currency with proof-of-burn,"

https://github.com/slimcoin-project/slimcoin-project.github.io/raw/master/Whitepaper_SLM.pdf, retrieved Oct. 2019, 2014.
[41] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016.
[42] K. Liao and J. Katz, "Incentivizing blockchain forks via whale transactions," in *International Conference on Financial Cryptography and Data Security*, 2017.
[43] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.
[44] I. Eyal, "The miner's dilemma," in *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 2015, pp. 89–103.
[45] R. Zhang and B. Preneel, "Lay down the common metrics: Evaluating proof-of-work consensus protocols security," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.
[46] X. Chen, C. Papadimitriou, and T. Roughgarden, "An axiomatic approach to block rewards," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, ser. AFT '19, 2019.
[47] fiatmarketcap.com. (2019) Top fiat currencies by market capitalization. [Online]. Available: https://fiatmarketcap.com/
[48] P. Ciaian, M. Rajcaniova, and d. Kancs, "The economics of bitcoin price formation," *Applied Economics*, 2016.
[49] L. Kristoufek, "What are the main drivers of the bitcoin price? evidence from wavelet coherence analysis," *PloS one*, 2015.
[50] J. Bouoiyour, R. Selmi, A. K. Tiwari, O. R. Olayeni *et al.*, "What drives bitcoin price," *Economics Bulletin*, 2016.
[51] /hackernoon.com. (2018) Factors influencing bitcoin price. [Online]. Available: https://hackernoon.com/factors-influencing-bitcoin-price-cfabdf634894
[52] wikipedia.com, "Cryptocurrency bubble," https://en.wikipedia.org/wiki/Cryptocurrency_bubble, 2018.
[53] A. Bloomenthal. (2019) What determines the price of 1 bitcoin? [Online]. Available: https://www.investopedia.com/tech/what-determines-value-1-bitcoin/
[54] N. Arnosti and S. M. Weinberg, "Bitcoin: A natural oligopoly," *arXiv preprint arXiv:1811.08572*, 2018.
[55] J. A. Garay, A. Kiayias, and N. Leonardos, "The Bitcoin backbone protocol: Analysis and applications," in *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2015.
[56] G. Huberman, J. Leshno, and C. Moallemi, "Monopoly without a monopolist: An economic analysis of the bitcoin payment system. ssrn scholarly paper id 3025604," *Social Science Research Network, Rochester, NY, URL https://papers. ssrn. com/abstract*, 2017.
[57] G. Goren and A. Spiegelman, "Mind the mining," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019.
[58] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol." in *NSDI*, 2016.
[59] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *25th USENIX Security Symposium (USENIX Security 16)*, 2016.
[60] G. Andresen, "Centralized mining," https://bitcoinfoundation.org/2014/06/centralized-mining/, 2014.
[61] M. Rosenfeld "Analysis of Bitcoin pooled mining reward systems," *arXiv preprint arXiv:1112.4980*, 2011.
[62] M. Rosenfeld, "Analysis of hashrate-based double spending," *arXiv preprint arXiv:1402.2009*, 2014.
[63] C. Karame, G. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, 2012.
[64] J. Bonneau, "Why buy when you can rent?" in *International Conference on Financial Cryptography and Data Security*, 2016.
[65] P. McCorry, A. Hicks, and S. Meiklejohn, "Smart contracts for bribing miners," in *International Conference on Financial Cryptography and Data Security*, 2018.
[66] A. Judmayer, N. Stifter, A. Zamyatin, I. Tsabary, I. Eyal, P. Gaži, S. Meiklejohn, and E. Weippl, "Pay-to-win: Incentive attacks on proof-of-work cryptocurrencies," 2019.
[67] andes, "Bitcoin's kryptonite: The 51% attack." https://bitcointalk.org/index.php?topic=12435, June 2011.
[68] V. Buterin, "Critical update re: Dao vulnerability," https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/, July 2014.
[69] R. Zhang and B. Preneel, "Publish or perish: A backward-compatible defense against selfish mining in bitcoin," in *Cryptographers Track at the RSA Conference*, 2017.

APPENDIX A
LEMMA PROOFS

We now bring the proofs for Lemma 1 and Lemma 2. We begin with Lemma 1:

Proof. Assume $\forall i,j \in 1,...,n : \textit{nes}_i = \textit{nes}_j$. It follows $\forall i \in 1,...,n : \mathbb{E}[S_i] = \textit{nes}_1 \cdot \ell b_i(k) F$. Therefore the expected income of agent i is $\textit{ExpInc}_i^{(\ell,\alpha,F)\text{-}\textit{IIE}}(k) = \frac{\textit{nes}_1 \cdot \ell b_i(k) F}{\sum_{j=1}^n \textit{nes}_1 \cdot \ell b_j(k) F} \ell r_k = b_i(k) \ell r_k$, leading to $\frac{\textit{ExpInc}_i^{(\ell,\alpha,F)\text{-}\textit{IIE}}(k)}{\sum_{j=1}^n \textit{ExpInc}_j^{(\ell,\alpha,F)\text{-}\textit{IIE}}(k)} = b_i(k)$, resulting in *size-indifference* $= 0$.

Now assume *size-indifference* $= 0$. By definition of *size-indifference* it holds that $\forall i \in 1,...,n : \frac{\textit{ExpInc}_i^{(\ell,\alpha,F)\text{-}\textit{IIE}}(k)}{\sum_{j=1}^n \textit{ExpInc}_j^{(\ell,\alpha,F)\text{-}\textit{IIE}}(k)} = b_i(k)$. Substituting in the utility $\mathbb{E}[S_i] = \textit{nes}_i \cdot \ell b_i(k) F$, that is $\forall i \in 1,...,n : \frac{\textit{nes}_i \cdot b_i(k)}{\sum_{j=1}^n \textit{nes}_j \cdot b_j(k)} = b_i(k)$ or $\forall i \in 1,...,n : \textit{nes}_i = \sum_{j=1}^n \textit{nes}_j \cdot b_j(k)$. As $\forall i \in 1,...,n : \mathbb{E}[S_i] > 0, b_i(k) > 0$ then it immediately follows that $\forall i,j \in 1,...,n : \textit{nes}_i = \textit{nes}_j \neq 0$. □

Lemma 2:

Proof. By the strong law of large numbers it holds that $N_i^k(C) \xrightarrow{a.s.} \mathbb{E}\left[N_i^k(C)\right]$ as $\ell \to \infty$. As $\mathbb{E}\left[N_i^k(C)\right] = \ell b_i(k)$ then $S_i \xrightarrow{a.s.} \ell b_i(k) F$ and $\textit{nes}_i \xrightarrow{a.s.} 1$. □

APPENDIX B
(ℓ,α,F) *-IIE - coalition-resistance* ANALYSIS

1) coalition-resistance: We devote this section to reason about $\sigma_{br}^{(\ell,\alpha,F)\text{-}\textit{IIE}}$ for various parameter values. Unlike presenting a strategy that triumphs over a what-to-be-believed a best-response strategy [25], [41], [27], [28], we face a different challenge —

finding the best-response strategy for a given protocol. The wide parameter range and strategy space complicate finding a close-form bound, and we take the following approach instead.

We follow the path of previous work [26], [29] and analyze $\sigma_{br}^{(\ell,\alpha,F)\text{-}IIE}$ using a Markov-Decision-Process (MDP), returning said strategy for a fixed set of system parameters. We detail the suitable MDP for (ℓ,α,F)-*IIE*, including the participating agents, action and state spaces, and the reward function. We present several parameter settings and their manifested best-response strategies, indicating how parameter choices affect *coalition-resistance.*

a) MDP best-response strategy search: As in previous work [1], [25], [26], [41], [43], [27], [28], [29], [55], [69], [29] we analyze the system from a perspective of a rational coalition of agents, modeled as a single *rational* agent, controlling a budget $B_r(k)$ striving to maximize her utility. We model all the other agents as a single *naive* agent, controlling a budget $B_n(k)$, following a predetermined strategy.

Note that $\sigma_{br}^{(\ell,\alpha,F)\text{-}IIE}$ is specific implementations of $\mathsf{Apportion}_r\left(S^g, B_r(k)\right)$, $\mathsf{Generate}_r^{(\ell,\alpha,F)\text{-}IIE}\left(S^g, S_r^l\right)$ and $\mathsf{Publish}_r\left(S^g, S_r^l\right)$, which we find in the following manner. First, we fix $\ell, F, \alpha, B_r(k), B_n(k)$. We then consider all possible instances of $\mathsf{Apportion}_r\left(S^g, B_r(k)\right)$ — recall that agent r can apportion her budget at will, however we can discretize and consider only implementations such that $B_r^{ic}(k) = 0, \ldots, \left\lfloor \frac{B_r(k)}{\alpha r_k} \right\rfloor$. For each $\mathsf{Apportion}_r\left(S^g, B_r(k)\right)$ instance we use the MDP to find the optimal $\mathsf{Generate}_r^{(\ell,\alpha,F)\text{-}IIE}\left(S^g, S_r^l\right)$ and $\mathsf{Publish}_r\left(S^g, S_r^l\right)$ implementations. For each such combination we let the agents play for 5000 games, denoting the most-rewarding combination as $\sigma_{br}^{(\ell,\alpha,F)\text{-}IIE}$.

Throughout the rest of this section we detail the MDP implementation and its results.

b) Rational agent: The rational strategy space includes all implementations of $\mathsf{Apportion}_r\left(S^g, B_r(k)\right)$, $\mathsf{Generate}_r^{(\ell,\alpha,F)\text{-}IIE}\left(S^g, S_r^l\right)$ and $\mathsf{Publish}_r\left(S^g, S_r^l\right)$ functions.

The implementation of $\mathsf{Apportion}_r\left(S^g, B_r(k)\right)$ stated by $\sigma_{desired}^{(\ell,\alpha,F)\text{-}IIE}$ is that the agent apportioning her budget such that $B_r^{ec}(k) = (1-\alpha)B_r(k), B_r^{ic}(k) = \alpha B_r(k)$. The rational agent might increase her expected income by allocating more external budget (and less internal), enabling her to create more blocks overall, at the expense of the number of factored blocks. Contrarily, higher internal budget will allow more factored blocks at the expense of expectedly creating less blocks overall.

Algorithm 5: $\sigma_{pc}^{(\ell,\alpha,F)\text{-}IIE}$ of agent i

```
Function Apportion(S^g, B_i(k)):
    return (1 − α) B_i(k)

Function Generate(S^g, S_i^l):
    C ← a uniformly-at-random chain from the longest chains
      with minimal score.
    pointer ← last(C)
    if N_i^k(C) < ⌊ B_i^ic(k) / (α r_k) ⌋ then
        factored ← true
    else
        factored ← false

    return NewBlock(pointer, factored)

Function Publish(S^g, S_i^l):
    return All previously unpublished blocks
```

The rational agent might also gain by using different implementations of $\mathsf{Generate}_r^{(\ell,\alpha,F)\text{-}IIE}\left(S^g, S_r^l\right)$ and $\mathsf{Publish}_r\left(S^g, S_r^l\right)$. For instance, *Selfish-mining* strategies [25], [26], [41], [43], [27], [29], [29] might prove profitable as they decrease the blocks and thus score of the opposing naive agent, yielding higher expected income for the rational agent.

c) Naive agent: As in [25], [27] the naive agent comprises infinitely-many, non-colluding infinitely-small budget agents. She follows a *petty-compliant* [27] strategy $\sigma_{pc}^{(\ell,\alpha,F)\text{-}IIE}$ that is a variant of $\sigma_{desired}^{(\ell,\alpha,F)\text{-}IIE}$, detailed in Algorithm 5.

In principal, $\sigma_{pc}^{(\ell,\alpha,F)\text{-}IIE}$ identifies with $\sigma_{desired}^{(\ell,\alpha,F)\text{-}IIE}$ except at the tie-breaking decision regarding conflicting longest chains. The implementation of $\mathsf{Generate}_n^{(\ell,\alpha,F)\text{-}IIE}\left(S^g, S_n^l\right)$ by $\sigma_{desired}^{(\ell,\alpha,F)\text{-}IIE}$ states the agent tie breaks uniformly-at-random from the multiple longest chains, and by $\sigma_{pc}^{(\ell,\alpha,F)\text{-}IIE}$ the agent chooses uniformly-at-random from the longest chains that *with the minimum accumulated score*.

Such strategy is more logical from an agent's perspective as it is expected to increase her expected income. On a chain with lower score the relative reward of future blocks is higher, hence increasing her expected income. As in [27], the petty-compliant strategy by itself is not harmful (even if we did consider spontaneous forks), yet it enables the rational agent to utilize a wider range of strategies.

As in [27], [25], [26], [41] we limit the analysis to strategies considering at most two chains at any given time. The first is the *public* chain C_n, followed by the naive player, while the other is known only to the rational player, named the *secret chain* C_r.

By modeling the naive player to comprise

infinitely-many, non-colluding infinitely-small budget agents, and to follow $\sigma_{pc}^{(\ell,\alpha,F)\text{-IIE}}$ we only increase the ability of the rational player to increase her expected income by deviating from $\sigma_{desired}^{(\ell,\alpha,F)\text{-IIE}}$. We illustrate that with the following example.

Example. *Assume the last block last* (C_n) *on the public chain* C_n *is a factored block created by the naive player. According to* $\sigma_{desired}^{(\ell,\alpha,F)\text{-IIE}}$ *the next block should point to last* (C_n) *and extend* C_n*. Assume the scheduler picks the rational agent to create the next block, and she creates a regular block that point to the same block as last* (C_n)*.*

Also assume the rational agent publishes her created blocks immediately. That is, the rational agent has created a conflicting longest chain, so $LC\left(S^g\right) = \{C_n, C_r\}$*. Assume that by the rational agent's strategy she will point her next created block to last* (C_r)*.*

Note that $S\left(C_n\right) - S\left(C_r\right) = F - 1 > 0$ *and that the naive player follows* $\sigma_{pc}^{(\ell,\alpha,F)\text{-IIE}}$*, therefore if she is picked to create the next block she will deterministically choose to point it to last* (C_r)*.*

That means that independently of which agent gets to create the next block, last (C_n) *will not be pointed by following blocks, effectively removing it from any future longest chain. That means the rational player had managed to replace a factored block of the naive agent with her own regular block on the main chain, increasing her score while decreasing that of the naive agent, both effectively increasing her expected income.*

If the naive player had followed $\sigma_{desired}^{(\ell,\alpha,F)\text{-IIE}}$ *then she would have pointed her next block to last* (r) *with probability of only* $\frac{1}{2}$*, as both* C_n *and are* C_r *are of the same length. That means that block last* (C_n) *might still end on the concluding main chain (as a function of the future decisions of both agents), which will eventually decrease the rational agent's expected income.*

In a running system it is less likely that an agent will prefer to extend a chain excluding a previous block she already created, as that lowers her score and expected income. Moreover, different agents may pick randomly different chains. By modeling all the other agents in the system as infinitely-small budget agents we can neglect the effect of any single agent with different incentives than that of the others.

d) Action space: Agent r actions are elements in the form $\{$*chain_manipulation*, *block_type*$\}$. The field *chain_manipulation* describes how the rational player interacts with the secret and public chains, and may contain either one of the three values — *publish*, *adopt* and *wait*. The value of *publish* indicates the agent publishes the blocks of the secret chain, a value of *adopt* indicates the agent abandons the secret chain and adopts the public chain, and *wait* indicates the agent does neither the former nor the latter. The field *block_type* has a binary value, describing whether the next block the agent creates is factored.

e) State space: States are elements in the form of $\{$*attack_chain*, *main_chain*, *fork*$\}$. The fields *attack_chain* and *main_chain* represent the content of the secret and public chains. Note these fields may have a common prefix. The field *fork* has a binary value indicating whether the naive player is partitioned with regards to which of the two chains to extend. Note that *fork* is true only if the rational player had previously published her chain.

f) Reward function: Rewarding states are those where either the secret or the public chain are of ℓ blocks, that is *length* $($*attack_chain*$) = \ell$ or *length* $($*main_chain*$) = \ell$. Note that this restricts the rational agents to strategies bounded by the creation of ℓ blocks, which are assumed to be feasible only with negligible probability [29]. These states indicate the epoch conclusion and hence the reward distribution of (ℓ, α, F) *-IIE*.

Note. *In [26], [29] the authors analyze an infinite game and introduce a* truncation parameter T*, capping the length of the secret and public chains. In their system the publication (adoption) of the secret (public) chain leads to the initial state, and it accumulates rewards at* state-transitions *rather than at a set of final states. Their state space includes counters of the blocks in the secret and public chains, resulting in a state space complexity of* $\mathcal{O}\left(T^2\right)$*.*

Such an analysis is inapplicable in our system as it is described by a finite game, and the reward is distributed at the final concluding states. However, we cannot simply count blocks on the two chains but have to maintain their order, resulting in a state space complexity of $\mathcal{O}\left(2^\ell\right)$*.*

As an example, consider the situation where the rational agent has created a factored and a regular blocks on the secret chain, and then the naive agent creates a factored block on the public chain. If the honest block precedes the factored block on the secret chain, then the naive agent can publish the regular block, resulting in the naive agent deterministically adopting it and forfeiting her recently found factored block. However, if the factored block precedes the regular one, then publishing the first block will result

in a fork, resembling Lead-Stubborn Mining *[41].*

g) Search results: We conduct the following experiment. We fix $\ell = 10$ and for various values of $\alpha, B_r(k), B_n(k)$ we use binary-search for the minimal F such that $\sigma_{br}^{(\ell,\alpha,F)\text{-IIE}}$ identifies with $\sigma_{desired}^{(\ell,\alpha,F)\text{-IIE}}$. We search range is $F \in [1, 10^8]$ with a stopping criteria of $1e-6$.

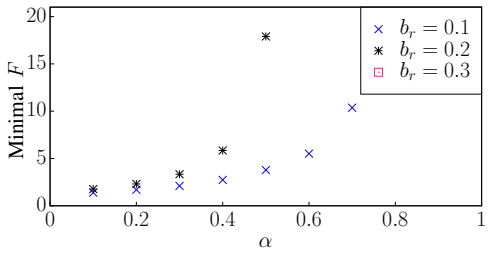


Fig. 3: Minimal F for obtaining $\sigma_{br}^{(\ell,\alpha,F)\text{-IIE}} = \sigma_{desired}^{(\ell,\alpha,F)\text{-IIE}}$ when $\ell = 10$. Missing data points indicate $\sigma_{br}^{(\ell,\alpha,F)\text{-IIE}} \neq \sigma_{desired}^{(\ell,\alpha,F)\text{-IIE}}$ for the search upper bound $\left(F = 10^8\right)$.

Fig. 3 brings the results of the mentioned search. A missing point indicates that no F is sufficient to ensure $\sigma_{br}^{(\ell,\alpha,F)\text{-IIE}} = \sigma_{desired}^{(\ell,\alpha,F)\text{-IIE}}$.

The results show lower α values require smaller F values to ensure $\sigma_{br}^{(\ell,\alpha,F)\text{-IIE}} = \sigma_{desired}^{(\ell,\alpha,F)\text{-IIE}}$. We note that F grows exponentially with α up to $\alpha = 0.6$, and from there even the maximal F values do not accommodate the desired behavior.

We also note that lower $b_r(k)$ requires lower F values. This settles with intuition, as an agent with lower relative budget is expected to gain less from selfish-mining-like strategies, hence they do not require higher F values to decrease their profitability. Note that for $b_r(k) = 0.3$ there is no F achieving the desired behavior, settling with with previous results. That is expected as the profitability threshold for selfish-mining is $b_r(k) = 0.232$ [26].

We conclude that *coalition-resistance* relies on ℓ, F and α, and we can obtain *coalition-resistance* $= 0.2$ even for $\alpha = 0.5$ by setting F appropriately.

A. Mounting a %51 Attack

Consider the following strategy of the rational agent. The agent creates blocks on her secret chain and ignores the public chain. When the secret chain is extended by ℓ the agent publishes it. This essentially embarks a race between the public and secret chains, in which the secret chain is expected to win should $B_r^{ec}(k) > B_n^{ec}(k)$. If in fact the secret chain grows faster then the rational agent gets the entirety of the block reward, as only she created blocks on that chain. This is known as a *%51 attack* [1], [67].

We explore the ability of the rational agent to mount such an attack. We therefore seek the minimal $b_r(k)$ for which this attack is feasible, that is the minimal $b_r(k)$ for which $B_r^{ec}(k) > B_n^{ec}(k)$.

As the naive agent follows $\sigma_{pc}^{(\ell,\alpha,F)\text{-IIE}}$ and therefore $B_n^{ec}(k) = (1-\alpha) B_n(k)$.

As the rational player expects to create a chain with only her blocks, she will receive the entirety of the reward as the sole block contributor anyhow, regardless of the score of her blocks. Hence, she can apportion her budget such that $B_r^{ec}(k) = B_r(k)$.

We get that the condition for this attack is $B_r(k) > (1-\alpha) B_n(k)$. Recall that $B_n(k) + B_r(k) = 1$ and $b_r(k) = B_r(k)$, landing the aforementioned inequation to $b_r(k) > (1-\alpha)(1-b_r(k))$ or $b_r(k) > \frac{1-\alpha}{2-\alpha}$. We present this lower bound pictorially in Fig. 4.

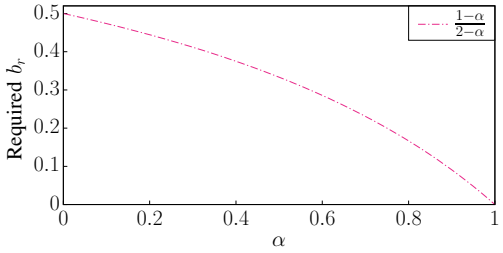


Fig. 4: Lower bound for $b_r(k)$ for %51 attack.

As expected, higher α values render the bound lower, as now the rational agent requires to surpass a lesser amount of *ec*. This result correlates with ℓ*-NB*, as when $\alpha = 0$ then $\frac{1-\alpha}{2-\alpha} = 0.5$, yielding the traditional bound [1], [67].

Trivially, if $b_r(k) > \frac{1-\alpha}{2-\alpha}$ then the rational agent maximizes her utility by performing this attack, and we therefore conclude *coalition-resistance* $< \frac{1-\alpha}{2-\alpha}$.