Tuning PoW with Hybrid Expenditure

Itay Tsabary
Technion, Haifa, Israel
IC3

Alexander Spiegelman
Novi Research, Herzliya, Israel

Ittay Eyal
Technion, Haifa, Israel
IC3

Abstract

Proof of Work (*PoW*) is a Sybil-deterrence security mechanism. It introduces an *external cost* to system participation by requiring computational effort to perform actions. However, since its inception, a central challenge was to tune this cost. Initial designs for deterring spam email and DoS attacks applied overhead equally to honest participants and attackers. Requiring too little effort does not deter attacks, whereas too much encumbers honest participation. This might be the reason it was never widely adopted.

Nakamoto overcame this trade-off in Bitcoin by distinguishing desired from malicious behavior and introducing *internal rewards* for the former. This mechanism gained popularity in securing permissionless cryptocurrencies, using virtual internally-minted tokens for rewards. However, in existing blockchain protocols the internal rewards directly compensate users for (almost) the same value of external expenses. Thus, as the token value soars, so does the PoW expenditure. Bitcoin PoW, for example, already expends as much electricity as Colombia or Switzerland. This amount of resource-guzzling is unsustainable, and hinders even wider adoption of these systems.

As such, a prominent alternative named *Proof of Stake* (*PoS*) replaces the expenditure requirement with token possession. However, PoS is shun by many cryptocurrency projects, as it is only secure under qualitatively-different assumptions, and the resultant systems are not permissionless.

In this work we present *Hybrid Expenditure Blockchain* (*HEB*), a novel PoW mechanism. *HEB* is a generalization of Nakamoto's protocol that enables tuning the external expenditure by introducing a complementary *internal-expenditure* mechanism. Thus, for the first time, HEB decouples external expenditure from the reward value.

We show a practical parameter choice by which *HEB* requires significantly less external consumption compare to Nakamoto's protocol, its resilience against rational attackers is similar, and it retains the decentralized and permissionless nature of the system. Taking the Bitcoin ecosystem as an example, *HEB* cuts the electricity consumption by half.

2012 ACM Subject Classification Security and privacy → Distributed systems security

Keywords and phrases Blockchain, Proof of work, Cryptocurrency, Environmental impact

Digital Object Identifier 10.4230/OASIcs.Tokenomics.2021.3

Related Version *Extended Version*: `https://arxiv.org/abs/1911.04124` [63]

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

Acknowledgements We thank Or Sattath for his comments on preliminary versions of this work. Part of this work was performed when the authors worked at VMware Research, Israel.

3rd International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2021).
Editors: Vincent Gramoli, Hanna Halaburda, and Rafael Pass; Article No. 3; pp. 3:1–3:17
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 Security scheme rewards-expenses comparison.

Internal Rewards	**Expenses**		
	Negligible	*External*	*External & Internal*
Exist	PoS Blockchains [39, 29, 24, 30]	PoW Blockchains [50, 10]	*HEB* (this work)
Absent	Open systems (e.g., email)	Original PoW [19, 37]	

1 Introduction

Permissionless systems are susceptible to *Sybil attacks* [18] where a single attacker can masquerade as multiple entities. To mitigate such attacks, *Proof of Work* (*PoW*) [19, 36] security schemes introduce *external costs*, making attacks expensive. To perform operations in a PoW system, users must provide a proof of computation, whose production requires resource expenditure. This makes attacks like email spam and denial of service [37] prohibitively expensive, as they require many operations. However, honest users are also subject to these costs, and the system cannot balance deterring adversarial behavior but not honest one [43].

To circumvent this trade-off, Nakamoto [50] suggested introducing *internal rewards* for honest behavior (Table 1 summarizes this taxonomy). Indeed, nowadays PoW is widely used to secure decentralized and permissionless cryptocurrencies like Bitcoin and Ethereum [10]. These are replicated state-machines [12] that facilitate monetary ecosystems of internally-minted *tokens*, maintained by principals called *miners.* Miners that follow the protocol are rewarded with tokens; tokens are scarce, hence a market forms [45, 9]; so, miners can sell their obtained tokens, compensating them for their PoW expenses.

To guarantee their security [26, 15], PoW cryptocurrencies moderate their operation rate by dynamically tuning the required computation difficulty to match miner capabilities [40, 55]. Consequently, the PoW expenditure directly depends on token values [48, 4, 42] – higher token prices imply higher mining rewards, which draw more miners to participate, leading to more expended resources. This results with the external PoW expenditure matching the internal mining rewards, balancing honest participation overhead with high attack costs.

Indeed, with exponentially-growing token values, the amount of resources spent on PoW mining has also been growing accordingly[1]. Bitcoin PoW computations alone are responsible for about 0.3% of the global electricity consumption [14, 17], surpassing medium-sized countries like Colombia and Switzerland [23]. This level of resource guzzling is unsustainable [21, 60], bears a significant ecological impact [33, 49], and prevents adoption [54, 32]. Unfortunately, Nakamoto's mechanism directly incentivizes external expenditure at the same rate as of the internal rewards, and offers no means of reducing its external effects.

Previous work (§2) explored PoW alternatives for cryptocurrencies, notably focusing on *Proof of Stake* (*PoS*) [39, 29, 24]. Such systems avoid the external resource expenditure by replacing the computational effort with internal token ownership requirements. However, PoS systems operate, and are secured under, qualitatively-different assumptions. Namely, they rely on deprecated data deletion [39], or extended user availability [29, 30, 24]. Moreover, new users need to obtain tokens to participate, requiring the permission of current stakeholders.

We note that naive adjustments of the cryptocurrency minting rate do not reduce the external expenses; that a simple reduction of rewards hampers security; and that *forcing* miners to internally-spend breaks the permissionless property of the system. We review these options in our extended report [63].

[1] https://www.blockchain.com/charts/

In this work we present *Hybrid Expenditure Blockchain* (*HEB*), the first PoW protocol with lower external costs than its internal rewards. Despite the reduced external expenditure, *HEB* provides similar security guarantees against rational attackers compared to the more-wasteful Nakamoto protocol. *HEB* is tunable, allowing to optimize for desired properties.

HEB Overview. The main challenge is to reduce the external expenses while keeping attack (and also participation) costs high. These objectives seem to contradict, as in previous work [19, 36, 50, 10] the participation costs are only external.

This is the main novelty of *HEB* – it is a generalization of Nakamoto's protocol that *enables* and *incentivizes* miners to forfeit system tokens as part of the mining process. Miners that do so increase their rewards, resulting with this being the optimal behavior. So, the external mining expenses in *HEB* are lower than existing PoW blockchains, while the total expenses (internal and external) are the same.

Similarly to Bitcoin, *HEB* constructs a tree data structure of elements named *blocks*, where the longest path defines the system state. Miners produce PoW to create blocks and broadcast them with a p2p network. However, unlike Bitcoin, *HEB* considers epochs of blocks, and the mining rewards for each epoch are distributed after its conclusion.

In *HEB* there are two types of blocks miners can create – *regular* or *factored*, which have a *weight* attribute with values 1 and $F > 1$, respectively. The epoch rewards are distributed among the miners proportionally to the relative weight of their blocks (in the epoch). Miners can always create regular blocks, but have to forfeit system tokens in epoch k to create factored blocks in epoch $k + 1$. This mechanism incentivizes miners to divert some of their total participation budget internally to create factored blocks, reducing their external PoW expenditure. The ratio between internal and external expenses is tuned with a parameter ρ, determining the expenditure required to create a factored block.

To maintain the total circulating token supply, the internally-expended tokens are distributed proportionally among all system entities (i.e., any token holder) at the epoch conclusion. This redistribution maintains the token value as in a standard PoW cryptocurrency (e.g., Bitcoin), as well as the relative wealth of all system entities. The internally-spent tokens are redistributed using a novel redistribution technique, which might be of independent interest (e.g., for regulating transaction fees, cf. [58]).

We emphasize that *HEB* draws ideas from PoS, prominently the utilization of system tokens for security, but the model assumptions, the solution, and the guarantees are distinct. In particular, *HEB* uses the standard PoW assumptions and miners expend (lose) their tokens for mining, whereas in PoS participants derive their power from maintaining ownership.

HEB Analysis. Our goal in analyzing *HEB* is twofold: show it is incentive-compatible, secure, and permissionless, similarly to Nakamoto's protocol, and; show it achieves the desired, reduced external impact. For these two purposes we lay forth the following groundwork. We begin by modeling the cryptocurrency ecosystem, and follow with the underlying data structures, participants, and execution (§3). As in previous work [20, 28, 61], we consider a set of rational miners that optimize their revenues and an adversary who is willing to expend resources in order to attack.

We then instantiate *Nakamoto*'s protocol (§4) and use it for a comparison baseline, following with the formal presentation of *HEB* (§5).

To compare and contrast *HEB* with *Nakamoto* we consider a variety of cryptocurrency metrics (§6). These include common security metrics, namely coalition resistance and tendency to encourage coalitions [20, 59, 62]. We also introduce a new metric – *external expenses*, measuring the resources spent on PoW. Instead of the binary metric permissioned/permissionless (classical-consensus-protocols/Nakamoto-blockchain, respectively), we introduce the continuous metric of *permissiveness*, describing the cost of joining the system.

Finally, we tease apart the common safety-violation security metric into two. We observe that safety-violating chain-reorganization attacks [57, 5, 35] in existing PoW blockchains require high resource investment from the attacker; however, once successful, they completely refund themselves. We therefore consider this type of attacks, as well as a sabotage variant where the attacker is not refunded. We show that in *HEB* the attack cost for the refunded variant is linear in the total expenses (as secure as *Nakamoto*), and that the sabotage variant costs are linear in the external ones.

HEB includes several parameters for the system designer to tune. As an example, we present a specific choice of parameter values (§7). Choosing the prominent Bitcoin ecosystem as a reference point, we analyze *HEB* and show this parameter choice is practical, achieves strong security guarantees, and reduces the external consumption by half – the equivalent of reducing the entire electricity consumption of Denmark [14].

In summary, we expand the PoW design space by introducing internal expenses. We present *HEB* – a PoW blockchain protocol with external expenses that are lower than its internal rewards. We prove that *HEB* offers similar security guarantees against rational attackers compared to pure PoW solutions, and show it can significantly reduce the latter's ecological impact.

2 Related Work

In Nakamoto's blockchain and all subsequent PoW protocols we are aware of, the incentives equal the value of the generated cryptocurrency tokens (and fees). We are not aware of previous work tuning PoW expenditure in cryptocurrencies – the main focus of this work.

We proceed to survey PoS and analysis approaches. Our extended report [63] discusses permissioned and trusted hardware solutions that make qualitatively-stronger assumptions; protocols that expend different external resources rather than electricity, for which *HEB* applies equally well; and protocols with several types of internal tokens that do not achieve incentive compatibility nor reduced external expenses.

Proof of Stake. *HEB* and PoS are fundamentally different: the latter limits miner participation to those with stake in the system, i.e., miners who own tokens; the former does not. Moreover, in PoS the Sybil-deterrence [18] is due to the cost of acquiring and holding the system tokens, which the participants maintain throughout the system execution. In contrast, *HEB* relies on PoW, and the participants *spend* the internal currency.

PoS systems like Algorand [39], Ouroboros [29], Tezos [30] and the forthcoming Ethereum 2.0 [24] are designed and analyzed under different assumptions than PoW. Their security is based on users' owned tokens rather than their expended resources. They assume a new participant wanting to join the system can acquire (or, alternatively, lock as a collateral [39, 30, 24]) as many system tokens as she can afford. That is, existing system miners authorize transactions that introduce new system miners, even if these result in a state less favorable from their perspective. Additionally, to combat *long-range attacks* [16, 27] and *nothing-at-stake* [56, 8], these systems assume users voluntarily delete deprecated data [39], or assume users remain online for extended periods [29, 30, 24].

In contrast, *HEB* is PoW-based, and newly-joining miners do not require the cooperation of existing miners to join. It is also resistant to said long-range attacks and the nothing-at-stake problems, and hence does not rely on voluntary data deletion or user persistence.

A parallel work [25], which draws ideas from a previous version of this report, suggests emulating PoW over PoS. The main idea is that the stake used for the consensus degrades over time and usage, mimicking the external expenditures of PoW systems. However, as built atop of PoS, it also requires the aforementioned assumptions.

Proof of Work Analysis. We use the standard techniques [20, 59, 52, 44, 28] to analyze *HEB*'s security and incentive compatibility. The evaluation metrics used are a formalization of previous ones presented by said work, and also include definition of new ones regarding the external expenditure and permissiveness level. To the best of our knowledge we are the first to define, evaluate and optimize for such metrics.

Chen et al. [13] define and analyze desired properties of reward allocation schemes in PoW cryptocurrencies. Their work focuses on the reward of a single block, and does not consider environmental impact nor malicious miners. We note that *HEB*'s reward allocation rule is incentive-compatible and Sybil-resistant, satisfying those desired properties.

3 Model

We present a model for an abstract blockchain system, instantiated with a cryptocurrency protocol. This allows us to compare different instantiations, namely *Nakamoto* and *HEB*. We first define the monetary value of system tokens using an exogenous reference-point fiat currency (§3.1). We follow by presenting the blockchain, the participating entities and how the system derives its state (§3.2). We then define how a cryptocurrency protocol instantiates that system, defining an internal system currency based on the blockchain (§3.3), and explain how the system makes progress (§3.4).

3.1 Cryptocurrency Economics

The external expenditure of a PoW cryptocurrency system depends on the rewards it grants miners and on mining costs. We note that mining rewards are internal while PoW costs are external, hence we first define the relation, or the exchange rate, of the two.

The reward is an amount of the system's internal currency *ic* (e.g., Bitcoin, Ether), and the external cost is an amount of an external currency *ec* (e.g., USD). We assume the external currency has a market capital orders of magnitude larger than that of the internal currency[2], and it effectively represents real values.

We assume there is an instantaneous and commission-free exchange service of *ec* and *ic*, where the exchange rate matches token real value. To simplify presentation we normalize the price level so the exchange rate is one, and assume the exchange is available to all participating entities. We often sum *ec* and *ic*, meaning the sum of their values in real terms.

3.2 Blockchain and System Principals

The system comprises a shared *global storage*, a *scheduler*, and two types of *entities*: system *users*, and principals maintaining the system named *miners*.

The global storage is an append-only set containing elements called *blocks*. Each block includes a reference to another block and data generated by system entities, with the only exception being a so-called *genesis block* that contains neither. The global storage initially

[2] https://fiatmarketcap.com/

contains only the genesis block, thus defining a directed tree data structure. We refer to paths in the data structure starting at the genesis block as *chains*. We partition a chain C to *epochs* of ℓ blocks.

As common [20, 59, 52, 61] we assume that the sets of entities are static during an epoch execution, that is, entities do not join or leave during the course of an epoch.

Each miner i has a local storage accessible only to her, which, like the global storage, is an append-only block set. The scheduler invokes miners, allowing them to create blocks in their local storage, and to publish their local blocks by copying them to the global storage. We denote by N_i^k the number of blocks in epoch k created by miner i. For presentation simplicity, we assume the main chain at an epoch beginning remains a prefix of the main chain throughout the entire epoch. Note this does not rule out the main chain changing during an epoch, but only that its initial prefix does not.

Entities derive the *system state* by parsing the global storage according to the block order of the main chain. They might choose to infer the state based on a chain prefix, excluding potentially-volatile suffixes [26], such as in the case of multiple longest chains. Such considerations are outside the scope of this work.

3.3 Instantiating a Cryptocurrency Protocol

The system is instantiated with a cryptocurrency protocol Π that defines a currency internal to the system, *ic*. The protocol Π maps all its internal tokens to system entities with a function $Bal^{\Pi}(C)$, taking as input a chain C. The function returns a vector where each element $Bal_i^{\Pi}(C)$ is the number of tokens mapped to entity i. When the context is clear, we often omit the protocol name Π and simply write $Bal(C)$.

We say the total value of tokens mapped to an entity is her *ic balance*, and note the total number of tokens is the sum of all balances. The protocol mints $r_k \cdot \ell$ new tokens at the end of each epoch k, and we often omit the epoch index when it is clear from context. This means the number of tokens is fixed throughout any epoch k, and increases when epoch $k+1$ begins. The protocol Π maps the newly-minted tokens to entities using $Bal(C)$.

Aside from their owned *ic*, miners also own *ec*. We use the terms *internal* and *external balances* to distinguish the different holdings, and simply *balance* for their aggregate value.

Miners expend all their balance on system maintenance. In practice, a principal can split its balance, using some of it as a miner, and the rest as a user. We model such principals as two separate entities – a miner that spends all its balance, and a user that holds the rest.

For any epoch k we denote by $B_i^{ec}(k)$, $B_i^{ic}(k)$ and $B_i(k)$ the initial external, internal and total balances of each miner i, respectively. We also denote by $b_i^{ec}(k)$, $b_i^{ic}(k)$ and $b_i(k)$ the *relative* (out of all miners' balances) the external, internal and total balances of miner i. Finally, we denote the total balances of all miners by $B_{Miners}(k)$ and of all users by $B_{Users}^{ic}(k)$.

We assume the value of expended resources by the miners on system maintenance in a single epoch k is much smaller than the system market cap. That is, the balance of all miners is negligible compared to that of all users, i.e., $B_{Miners}(k) \ll B_{Users}^{ic}(k)$. In practice, $\frac{B_{Miners}(k)}{B_{Users}^{ic}(k)} \ll 10^{-7}$ holds for both Bitcoin and Ethereum[3].

[3] https://coinmarketcap.com/currencies/

3.4 Execution

Initially, the global storage contains only the genesis block, and each miner has an empty local storage. The state variables (like the global and local storage) change over time, but we omit indexing as it is clear from the context.

The system progresses is orchestrated by a scheduler, where epoch k begins when the main chain is of length ℓk. First, the scheduler lets miners set their internal and external balances using the exchange service, achieving their preferred balance of the two. We use the term *allocate* to describe this action, and say miner i allocates her balance $B_i(k)$ with the invocation of the $\mathsf{Allocate}_i(B_i(k))$ function, returning a tuple of her internal and external balances $\langle B_i^{ic}(k), B_i^{ec}(k)\rangle$.

Note that modeling changes in the miner set and balance allocations at epoch transitions is for presentation simplicity; these occur throughout the system execution.

The rest of the epoch execution progresses in steps, until the main chain is extended by ℓ blocks. Each step begins with the scheduler selecting a single miner at random, proportionally to her relative *external expenditure*, that is $\Pr(\text{scheduler selects } i) = b_i^{ec}(k)$. Similarly to previous work [20, 1, 59, 52], these steps represent a standard PoW mechanism and its logical state changes, and entities have synchronous access to the global storage.

The scheduler invokes the selected miner i's function $\mathsf{Generate}_i^{\Pi}()$, returning a newly generated block, and adds it to miner i's local storage. The protocol Π states block validity rules in $Bal^{\Pi}(C)$, and invalid blocks do not affect the system state. Creating an invalid block or not creating one at all is sub-optimal and we only consider miners who avoid doing so.

Next, the scheduler lets any miner i publish any of her unpublished blocks by invoking $\mathsf{Publish}_i()$, returning a subset of her previously-private local blocks. The scheduler adds the returned blocks to the global storage, and repeats this process until all miners do not wish to publish any more blocks. This captures strategic-block-release behaviors [20, 59, 52].

The cryptocurrency protocol Π includes implementations of $\mathsf{Allocate}_i(B_i(k))$, $\mathsf{Generate}_i^{\Pi}()$, and $\mathsf{Publish}_i()$ that each miner i should follow. We refer to the tuple of three implementations as the *prescribed strategy* and denote it by $\sigma^{\Pi}_{prescribed}$. The protocol Π is therefore a tuple of the balance function Bal^{Π} and a prescribed strategy $\sigma^{\Pi}_{prescribed}$. Note that Π cannot force miners to follow $\sigma^{\Pi}_{prescribed}$.

4 Nakamoto's Protocol

As an example and to serve as a baseline, we instantiate an epoch-based *Nakamoto* protocol (used with $\ell = 1$ in Bitcoin, Litecoin, etc.) in our model.

The balance function of *Nakamoto* awards each miner i with r tokens per block she created in the epoch, and a total of $\ell \cdot r$ new tokens are minted. Hence, the balance of each miner i at epoch conclusion is $N_i \cdot r$.

The prescribed strategy $\sigma^{Nakamoto}_{prescribed}$ states that each miner i allocates her balance $B_i^{ec} = B_i$ and $B_i^{ic} = 0$, extends the longest chain, and publishes her blocks immediately. In case of multiple longest chains, $\sigma^{Nakamoto}_{prescribed}$ picks uniformly-at-random[4].

[4] Bitcoin defines a different tie-breaking rule – pick the first longest chain the miner became aware of. Therefore its security guarantees vary, depending on the underlying network assumptions. As in previous work [41, 59], we avoid such assumptions by considering the uniformly-at-random variation.

5 *HEB* Protocol

We are now ready to present *HEB*. Briefly, it incentivizes miners to expend their balances internally by enabling miners who do so to create higher-reward blocks. Two parameters, $\rho \in [0,1)$ and $F \in \mathbb{R}_{>1}$, control the reward distribution mechanism. We detail the different block types, the reward distribution mechanism, and the desired strategy.

Block types. Each block has a type, determined at its creation – either *regular* or *factored*. During an epoch, miner i can create regular blocks whenever the scheduler invokes $\mathsf{Generate}_i()$. However, aside from an invocation by the scheduler, creating a factored block requires an internal expenditure of $\rho \cdot r$ in ic by miner i at the previous epoch; recall we model these internal expenditures as if they occur at the start of current epoch. We emphasize that creating blocks of either type occurs only by an invocation of the scheduler, i.e., based on the external expenditure of the miner.

Consequently, if $\rho > 0$ then miner i can create at most $\left\lfloor \frac{B_i^{ic}}{\rho \cdot r} \right\rfloor$ factored blocks in an epoch on chain C. *HEB* assigns a *weight* to each block according to its type, and factored and regular blocks have weights of F and 1, respectively.

Reward distribution. *HEB* distributes the ℓr minted tokens among the miners in proportional to their block weights. Denote by $W_i(C)$ the total block weight of miner i on chain C. So, miner i gets $\frac{W_i(C)}{\sum_j W_j(C)} \ell r$ tokens for her created blocks.

The internal expenses B_{Miners}^{ic} are distributed among all system entities (i.e., including users) proportionally to their ic balances at the epoch beginning. So, miner i receives $\frac{B_i^{ic}}{B_{Miners}^{ic} + B_{Users}^{ic}} B_{Miners}^{ic}$ tokens from the redistribution. We discuss shortcomings of other distribution schemes in the extended report [63].

In summary, miner i gets $\frac{W_i()}{\sum_j W_j()} \ell r + \frac{B_i^{ic}}{B_{Miners}^{ic} + B_{Users}^{ic}} B_{Miners}^{ic}$ at the epoch conclusion.

Prescribed strategy. The prescribed strategy $\sigma_{prescribed}^{HEB}$ states that miners allocate their balance with ratio ρ and create factored blocks up to their internal balance limitation. Formally, miner i allocates $B_i^{ic} = \rho B_i$ and $B_i^{ec} = (1-\rho) B_i$. If $\rho = 0$ then the miner creates all blocks as factored, and if $\rho > 0$ then only the first $\left\lfloor \frac{B_i^{ic}}{\rho \cdot r} \right\rfloor$ are factored. As in *Nakamoto*, miner i points her created blocks to the longest chain, and publishes them immediately.

▶ **Note.** *Setting $\rho = 0$ enables miners to create all blocks as factored, and setting $F = 1$ removes motivation to create any factored blocks at all. In both cases there is only one practical block type, reducing HEB to Nakamoto.*

We discuss practical implementation aspects of *HEB* in our extended report [63] – shortening epochs, utilizing a pure PoW ramp-up period to create a sufficiently-large currency circulation, and addressing discretization issues.

6 Evaluation

We now evaluate *HEB*, showing how parameter choices affect its properties. For that, we formalize the cryptocurrency system as a game played by the miners, striving to maximize their rewards (§6.1). We use *Nakamoto* as a baseline, highlighting *HEB* parameter choices that result with significantly lower PoW expenditure while limiting undesirable side-effects.

To compare we first need to define criteria. Hence, throughout this section we present cryptocurrency evaluation metrics, each followed by its evaluation in *Nakamoto* and in *HEB*. We consider common security metrics [20, 13, 26] regarding the *incentive compatibility* of a system (§6.2 and §6.3); refine the common safety-violation security metric [38, 7, 46], measuring attack *costs* (§6.4); generalize the binary permissioned/permissionless notion [50, 3] to a continuous metric (§6.5); and conclude with a new metric for external expenses (§6.6).

6.1 Block Creation as a Game

The model gives rise to a game, played by miners for the duration of a single epoch k (hereinafter omitting the epoch index). We define the utility U_i of miner i as her expected cryptocurrency holdings at the conclusion of the epoch.

As commonly done in the analysis of cryptocurrency protocols [20, 34, 61, 59], we assume that during an epoch the system is quasi-static, where all miners participate and the total profit is constant. In operational systems miners participate for a positive profit [48, 64], but discussing the required return-on-investment ratio for such behavior is out the scope of this work, and we arbitrarily assume it to be 0 [22, 31]. Accordingly, the sum of all miner utilities equals the overall miner balances, that is, $B_{Miners} = \sum_i U_i$.

We normalize the number of newly-minted tokens per block to be one, meaning $r = 1$, so a total of ℓ tokens are created in the epoch.

The strategy space comprises choosing the allocation ratio, what blocks to generate, and when to publish them, i.e., implementations of $\mathsf{Allocate}\,()$, $\mathsf{Generate}\,()$ and $\mathsf{Publish}\,()$.

▶ **Example** (Nakamoto)**.** We demonstrate the compatibility of our definitions and modeling with previous results [51] regarding *Nakamoto*. We consider a scenario where all miners follow $\sigma^{Nakamoto}_{prescribed}$. So, all miner balances are in *ec* and consequently $b_i = b_i^{ec}$. Additionally, all miners extend the longest chain.

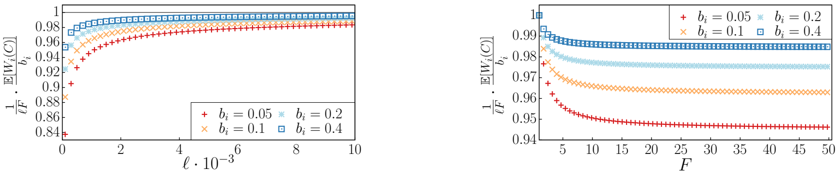
We note the scheduler picks at each step a miner proportional to her relative external balance. We can consider each pick as a Bernoulli trial where miner i is picked with success probability of b_i. So, the number of blocks a miner i creates in an epoch is binomially distributed $N_i \sim \mathrm{Bin}\,(\ell, b_i)$.

Therefore, $\mathbb{E}\,[N_i] = b_i^{ec}\ell$, and the utility of miner i is $U_i^{Nakamoto} = b_i^{ec}\ell$, matching previous analysis [50]. Summing for all miners yields $B_{Miners} = B^{ec}_{Miners} = \ell$, and the expected cost to create each block is 1, matching its reward.

6.2 *Size bias*

Cryptocurrency security relies on having multiple, independent miners, none of which has control over the system [13, 26]. For that, these systems strive to distribute their rewards in a way that is *size-indifferent*, meaning that miners get relative reward matching their relative balances, and hence have no incentive to coalesce. The metric *Size bias* measures how well a protocol satisfies this desideratum when all miners follow the prescribed strategy. Unlike the other metrics, it is evaluated for a specific balance distribution.

Formally, assume a balance distribution and that each miner i with relative balance b_i follows $\sigma^{\Pi}_{prescribed}$. The utility of such miner is U_i^{Π}, and her relative utility is $\frac{U_i^{\Pi}}{\sum_j U_j^{\Pi}}$. We define *Size bias* to be the maximal difference of each miner's relative balance and relative utility, that is, *Size bias* $\triangleq \max_i \left| b_i - \frac{U_i^{\Pi}}{\sum_j U_j^{\Pi}} \right|$.



(a) $F = 20$.

(b) $\ell = 1000$.

Figure 1 $\frac{1}{\ell F} \cdot \frac{\mathbb{E}[W_i(C)]}{b_i}$ for ρ and F values.

Systems strive for *Size bias* to be minimal, as higher values indicate more disproportionate shares. Preferably, *Size bias* $= 0$, indicating all miners are rewarded proportionally.

In practice, there is an inherent advantage for having a larger relative balance due to fork-rate [53, 47] and economy-of-scale [1] considerations, and although *Size bias* $= 0$ is a theoretical desideratum, systems like Bitcoin successfully operate even with non-zero values.

Nakamoto. Recall that in *Nakamoto* the utility of each miner i is $U_i^{Nakamoto} = b_i^{ec}\ell$, meaning $\frac{U_i^{Nakamoto}}{\sum_j U_j^{Nakamoto}} = b_i$ and in our model *Size bias* $= 0$. This matches previous analysis [50].

HEB. We move to analyze *HEB*'s *Size bias*. Throughout this section we present a summary of the analysis and its results, and bring the details in our extended report [63].

In *HEB* it holds that if all miners follow the prescribed strategy then $B_{Miners} = \ell$ and the utility of each miner i is $U_i^{HEB} = \frac{\mathbb{E}[W_i(C)]}{\sum_j \mathbb{E}[W_j(C)]}\ell$. To show the above we first show the redistributed internal currency a miner receives is negligible, allowing us to focus on the minting reward. For that we analyze the number of blocks a miner creates. Then, we derive her conditional total block weight, that is, her total block weight conditioned on the number of blocks she creates. We proceed to derive her expected total block weight, and conclude with finding her utility.

Then, we show that *HEB* achieves *Size bias* $= 0$ with sufficiently long epochs, as formalized by the following corollary:

▶ **Corollary 1.** *HEB achieves* $\lim_{\ell \to \infty}$ *Size bias* $= 0$.

The proof begins by showing that if all miners follow the prescribed strategy, then for any two miners i, j the ratio of the expected weight and relative budget is equal $\frac{\mathbb{E}[W_i(C)]}{b_i} = \frac{\mathbb{E}[W_j(C)]}{b_j}$ iff *Size bias* $= 0$. Then it shows that if all miners follow $\sigma_{prescribed}^{HEB}$, then $\lim_{\ell \to \infty} \frac{\mathbb{E}[W_i(C)]}{b_i} = \ell F$ for any miner i, and consequently, the former condition holds.

We conclude with concrete number instantiations, showing that *Size bias* improves (decreases) with longer epochs (larger ℓ) and a smaller factor (smaller F) value, while being independent of ρ. We also show more balanced distributions have lower *Size bias*, but note these are not under the control of the system designer. Considering practical parameter choices, we show that even for an extreme balance distribution, *HEB* achieves *Size bias* $< 0.3\%$. In a similar, yet balanced scenario, *Size bias* $= 0$.

For that, we calculate $\frac{\mathbb{E}[W_i(C)]}{b_i}$ for various F, ℓ and b_i values. We present the results in Fig. 1, multiplied by $\frac{1}{\ell F}$ for comparison purposes. Although we present results for specific configurations, we assert that different parameter values yield the same qualitative results.

Table 2 *Size bias* for $\ell = 1000$ and $F = 20$.

Balance distribution					*Size bias*
b_1	b_2	b_3	b_4	b_5	
0.20	0.80	–	–	–	**0.0029**
0.10	0.15	0.20	0.20	0.35	**0.0025**
0.20	0.40	0.40	–	–	**0.0015**
0.20	0.20	0.30	0.30	–	**0.0007**
0.20	0.20	0.20	0.20	0.20	**0.0000**

Fig. 1a shows for a fixed $F = 20$ the value of $\frac{1}{\ell F} \cdot \frac{\mathbb{E}[W_i(C)]}{b_i}$ as a function of ℓ. As expected, $\frac{1}{\ell F} \cdot \frac{\mathbb{E}[W_i(C)]}{b_i}$ approaches 1 as ℓ grows, leading towards *Size bias* $= 0$. However, for any fixed ℓ value, miners of different b_i have different $\frac{1}{\ell F} \cdot \frac{\mathbb{E}[W_i(C)]}{b_i}$, resulting with *Size bias* > 0.

We also illustrate the effect of F on $\frac{1}{\ell F} \cdot \frac{\mathbb{E}[W_i(C)]}{b_i}$. Fig. 1b shows $\frac{1}{\ell F} \cdot \frac{\mathbb{E}[W_i(C)]}{b_i}$ for $\ell = 1000$ as function of F. At the region of lower F values, increasing F also increases the difference of $\frac{1}{\ell F} \cdot \frac{\mathbb{E}[W_i(C)]}{b_i}$ for different b_i. However, as F becomes larger, then $\frac{1}{\ell F} \cdot \frac{\mathbb{E}[W_i(C)]}{b_i}$ tends towards a constant and the difference for different b_i remains fixed. This is expected, as for larger F values the expected weight is dominated by the expected weight of factored blocks (see [63]), and the expected weight becomes linear in F.

We dedicate the rest of this section to analyze how different balance distributions affect miners' utilities and *Size bias*. We consider various settings of at most 5 miners with epoch length of $\ell = 1000$ blocks and $F = 20$.

For each setting we numerically calculate *Size bias* and present it, along with its respective balance distribution, in Table 2. We choose these specific settings to demonstrate *Size bias* both balanced and extreme distributions.

Table 2 shows that more extreme balance distributions results in higher *Size bias*. For instance, consider the setting with only two miners where $b_1 = 0.2$ and $b_2 = 0.8$. This setting leads to the highest value of *Size bias* $= 0.0029$. Note that this is an unrealistic setting, presented only as an example for a highly-uneven distribution. Even in this extreme scenario miner 1 has a degradation of less than 0.3% in her relative utility. More balanced settings lead to lower *Size bias* values.

In summary, even a highly-unbalanced distribution results in minor deviations from proportional rewards. Increasing ℓ and decreasing F both reduce these deviations.

6.3 *Nash threshold*

Recall protocol Π provides a prescribed strategy $\sigma^{\Pi}_{prescribed}$ that miners individually choose whether to follow. The protocol properties rely on miners following this strategy [20, 59, 52, 11, 62, 28, 61], hence it should incentivize miners to do so.

The question is whether the prescribed strategy is a Nash-equilibrium, meaning no miner can benefit from individually deviating to a different strategy. Like in previous work [20, 59, 52], the *Nash threshold* metric is the maximal relative miner balance that achieves this: If all miners have relative balances smaller than the threshold, then the prescribed strategy is a Nash-equilibrium.

Formally, denote by $\sigma^{\Pi}_{\mathrm{i},best}$ the best-response strategy of miner i with relative balance b_i when all other miners follow $\sigma^{\Pi}_{prescribed}$. *Nash threshold* is the maximal value b_i such that $\sigma^{\Pi}_{\mathrm{i},best} = \sigma^{\Pi}_{prescribed}$. It follows that $\sigma^{\Pi}_{prescribed}$ is a Nash-equilibrium if all miner relative balances are not greater than *Nash threshold*.

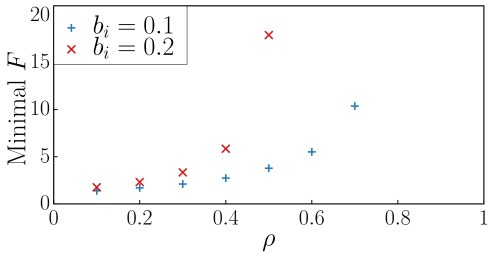


Figure 2 Minimal F for *Nash threshold.*

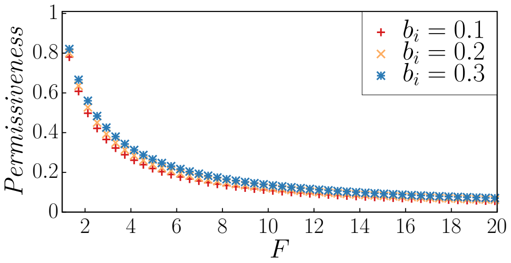


Figure 3 *HEB Permissiveness.*

Nakamoto. Sapirshtein et al. [59] showed that for *Nakamoto* with the uniform tie-breaking fork selection rule (see §4) the metric value is *Nash threshold* $= 0.232$.

HEB. An optimal miner strategy must consider how to allocate the balance, which previous blocks to point to, what block type to create, and when to publish created blocks.

Before considering the best strategy a miner can follow, we start by considering a specific, natural, *PoW-only* strategy $\sigma^{HEB}_{PoW\text{-}only}$, which simply ignores the internal expenditure. The idea of $\sigma^{HEB}_{PoW\text{-}only}$ is to maximize the block creation rate by expending all resources externally. The miner tries to create all the epoch blocks herself, and thus obtain all the epoch rewards.

This strategy is of interest as it abuses the internal expenditure mechanism; it is also simple enough to lend itself to a closed-form analysis. Specifically, we show (in our extended report [63]) that this strategy is more rewarding than $\sigma^{HEB}_{prescribed}$ if $b_i > \frac{1-\rho}{2-\rho}$. It follows the latter is an upper bound for *Nash threshold.* We note that higher ρ values lower the bound, as miner i is competing against less external balance. This result matches *Nakamoto*, as if $\rho = 0$ then $\frac{1-\rho}{2-\rho} = 0.5$, yielding the established 50% bound [59, 42]. We now move to search for the best-response strategy of a miner.

Following previous work [59, 28], we use *Markov Decision Process* (*MDP*) to search for the optimal strategy in *HEB.* The MDP includes the internal expenditure and block weights, and produces miner i's best-response strategy $\sigma^{HEB}_{\mathrm{i},best}$ based on system parameters. We defer the MDP technical details to our extended report [63].

We note the state and action spaces grow exponentially with the epoch length, limiting available analysis to relatively small epoch values. Therefore, similarly to previous work [59, 28], we also limit the state space by excluding strategies requiring longer, and thus less probable, sequences of events.

Our focus is finding the required parameter values for which $\sigma^{HEB}_{\mathrm{i},best}$ matches $\sigma^{HEB}_{\mathrm{i},prescribed}$. Recall that $\sigma^{HEB}_{\mathrm{i},best}$ is the optimal implementation of $\mathsf{Allocate}_i()$, $\mathsf{Generate}_i()$ and $\mathsf{Publish}_i()$ given B_i and the system parameters ℓ, F, ρ, hence we take the following approach.

We fix $\ell = 10$ to limit the state space, and for various values of ρ and b_i we use binary-search to find the minimal $F \in [1, 10^8]$ value such that $\sigma^{HEB}_{\mathrm{i},best} = \sigma^{HEB}_{prescribed}$. First, we consider $\mathsf{Allocate}_i()$ implementations that let miner i create a natural number of blocks (allocating balance to enable the creation of a fraction of a block is strictly dominated, enabling discretization of possible implementations). For each such implementation we use the MDP to obtain the optimal implementation of $\mathsf{Generate}_i()$ and $\mathsf{Publish}_i()$. We let the miner play the resultant strategies, and take the most rewarding to be $\sigma^{HEB}_{\mathrm{i},best}$.

We present the results in Fig. 2, showing that increasing F values and lowering ρ increases *Nash threshold.* Specifically, for $b_i = 0.2$ the required F values grow exponentially with ρ up to $\rho = 0.5$, and from there even the maximal F value does not accommodate the desired behavior. We note a similar behavior for $b_i = 0.1$, growing exponentially with ρ up to $\rho = 0.7$, being the maximal ρ that leads to $\sigma^{HEB}_{prescribed}$ being a Nash-equilibrium.

We also note that lower b_i requires lower F values, and specifically, there are no F and ρ values for which the configuration of $b_i = 0.3$ achieves a Nash-equilibrium. This is expected as the profitability threshold for selfish-mining variants is $b_i = 0.232$ [59], and indeed the resultant best-response strategies resemble selfish-mining in *Nakamoto*.

We conclude that *Nash threshold* relies on ℓ, F and ρ; by setting $F = 20$, we can obtain *Nash threshold* $= 0.2$ even for $\rho = 0.5$, close to *Nakamoto*'s value [59].

6.4 *Free safety-violation threshold* and *Safety-violation threshold*

We consider safety-violation attacks [38, 7, 46, 2, 57, 5, 35] as scenarios where an attacker causes the system to make an invalid transition. This can be achieved by creating and publishing an alternative chain, surpassing the main one. The original chain blocks are then discarded, and the system state is reinstated according to the blocks on the alternative chain.

To mount this attack in *Nakamoto* the attacker expends her resources on creating blocks to form the alternative chain; recall that each block costs its worth in reward to create. Therefore, if the attack is successful, the attacker is fully compensated for her expenditures by the rewards from her created blocks. As such, there is a threshold of required resources to mount this attack, but once met, the attack is *free*.

The metric *Free safety-violation threshold* measures the minimal required balance for a miner to deploy such a refunded safety-violation attack on the system, assuming all other miners follow the prescribed strategy $\sigma^{\Pi}_{prescribed}$. As shown in previous work [7], the attacker may rent vast computational resources for a short period of time or a moderate amount for longer periods. We therefore measure the expected cost to create a single block, disregarding the attack duration and amplitude.

The *Safety-violation threshold* metric removes the refund requirement, and simply represents the cost to create a block.

Formally, assume all miners follow $\sigma^{\Pi}_{prescribed}$. Then, *Free safety-violation threshold* is the minimal cost to create a block, guaranteeing full compensation should it be on the main chain, and *Safety-violation threshold* is this cost without any further compensation guarantees.

Nakamoto. In *Nakamoto* the cost to create each block is 1, hence *Free safety-violation threshold* $= 1$. All blocks produce the same reward, hence a miner cannot reduce the cost for a safety-violation attack by choosing to create less-rewarding blocks. Therefore, *Safety-violation threshold* $= 1$.

HEB. In equilibria the total external expenses are $1 - \rho$ of the total balances, that is $B^{ec}_{Miners} = (1 - \rho)\, B_{Miners}$. As $B_{Miners} = \ell$ it follows that the required external expenses to create a single block is $1 - \rho$.

As other miners create factored blocks, a miner also has to create a factored block to be fully compensated for her expenses, requiring additional spending of ρ. Hence, the cost to create a single block is 1, so *Free safety-violation threshold* $= 1$. That is, *HEB* is as resilient to refunded attacks as *Nakamoto*.

Alternatively, a miner can disregard compensation and choose to create regular blocks, baring no additional internal expenses, and so *Safety-violation threshold* $= 1 - \rho$, which is less secure than *Nakamoto*. However, the lack of direct compensation makes these attacks very expensive, hence they are only available to a well-funded adversary with an exogenous utility, e.g., interested in destabilizing or short-selling a cryptocurrency.

Indeed, previous attack instances [57, 5, 35] were on relatively-small systems and were of the former, refunded type. We are not aware of such sabotage attacks happening in practice; this is possibly because the required expenditure surpasses the potential profit [2].

6.5 *Permissiveness*

Cryptocurrency protocols implement their own reward distribution mechanisms [13], and may choose to condition rewards on a miner having the internal system currency *ic*. For example, in PoS systems [39, 29], owning *ic* is a requisite, and miners without *ic* cannot participate and get rewards. In contrast, in PoW systems this is not the case.

Acquisition of *ic* involves an update of the new currency ownership in the system state. This requires the cooperation of the present system miners: They decide which state updates occur when placing data in their created blocks. So, if token ownership is a mining requirement, then a new miner wishing to participate requires the cooperation of existing miners.

Previous work considered either permissioned systems that require token ownership [29, 39, 30] (some also require explicitly locking owned tokens as a collateral), or permissionless systems [50, 10] that do not.

We generalize this binary differentiation to a continuous metric, *Permissiveness*, measuring the revenue of a newly-joining miner without cooperation from the incumbents. The metric is the ratio between the revenues of a miner where she failed or managed to obtain *ic*.

Formally, consider a miner i with balance B_i, and assume that all other miners follow $\sigma^{\Pi}_{prescribed}$. Denote by $\sigma^{\Pi}_{prescribed\text{-}no\text{-}ic}$ a strategy identical to $\sigma^{\Pi}_{prescribed}$ with the exception that the $\mathsf{Allocate}()$ implementation returns $\langle 0, B_i \rangle$. Note this captures the inability of miner i to obtain *ic*. Denote by $U^{\Pi}_{i,prescribed\text{-}no\text{-}ic}$ and by $U^{\Pi}_{i,prescribed}$ the utility of miner i if she follows $\sigma^{\Pi}_{prescribed\text{-}no\text{-}ic}$ and $\sigma^{\Pi}_{prescribed}$, respectively. We then define $\mathit{Permissiveness} \triangleq \frac{U^{\Pi}_{i,prescribed\text{-}no\text{-}ic}}{U^{\Pi}_{i,prescribed}}$.

If $\mathit{Permissiveness} = 1$ then a miner's utility is not affected by her inability to obtain *ic*, meaning the protocol is permissionless. In contrast, $\mathit{Permissiveness} = 0$ indicates that a miner who cannot obtain *ic* is completely prevented from participation.

Nakamoto. As a pure PoW blockchain protocol, *Nakamoto* miners do not require *ic* balance, so $\mathit{Permissiveness} = 1$.

HEB. Calculating both utilities, we get that $\mathit{Permissiveness} = (b_i + F(1 - b_i))^{-1}$ ([63]), which Fig. 3 presents for different values of b_i as a function of F. It shows that higher factor values F lead to lower *Permissiveness* values, making the system more permissioned. It also shows that miners with higher relative balances are slightly less susceptible to these effects.

Although failure to obtain *ic* results with a lower reward, it still enables the new miner to create blocks herself, removing the requirement for cooperation from the incumbents in the subsequent epochs. The reduced reward in the first epoch is a one-time cost that is negligible for a long-running miner. This is a significant and qualitative improvement over permissioned systems, where a miner that cannot obtain tokens [29] or lock them as a collateral [39, 30, 24] is blocked from all future participation.

6.6 *External expenses*

External expenses evaluates the external expenditure of the protocol, and lower values indicate a lower environmental impact. Formally, assume all miners follow $\sigma^{\Pi}_{prescribed}$. *External expenses* is the total of miner external expenses, measured in *ec*, normalized by the epoch length, i.e, $\mathit{External\ expenses} \triangleq \frac{B^{ec}_{Miners}}{\ell}$.

Nakamoto. The total miner expenses are $B^{ec}_{Miners} = \ell$, so $External\ expenses = 1$.

HEB. When all miners follow $\sigma^{HEB}_{prescribed}$ then $B^{ec}_{Miners} = (1-\rho)\,B_{Miners}$ and $External\ expenses = 1-\rho$. This is the main advantage of *HEB* over *Nakamoto*.

7 Practical Parameters

As we have seen, *HEB* presents several knobs for the system designer. Longer epoch length ℓ improves *Size bias*, however, also means that reward distribution takes longer. Higher factored block weight F improves *Nash threshold* at the expense of *Permissiveness*. Higher internal expenditure rate ρ reduces the external expenditures, but makes the system less robust against rational miners, and reduces the required costs for sabotage attacks.

The choice of parameter values should be according to the desired system properties. Each system has different goals, and we emphasize that determining optimal parameter values is not a goal of this work. Nevertheless, in this section we consider a specific parameter choice. We compare this instantiation to Bitcoin, and use the latter's miner balance distribution [6] as a representative example.

We choose the external cost parameter to be $\rho = 0.5$, the epoch length to be $\ell = 1000$, and the factor to be $F = 20$. First and foremost, this setting results with only half of the external resource consumption ($External\ expenses = 0.5$), which is equivalent to reducing the entire power consumption of Denmark [14]. This choice incentivizes rational miners with up to 0.2 relative balance to follow the prescribed strategy ($Nash\ threshold = 0.2$) down from Bitcoin's 0.232 [59]. Note that the largest miner in Bitcoin has a relative balance of 0.16 [6], so rational miners would follow the prescribed, honest mining behavior.

With Bitcoin's expected block creation interval of 10 minutes, having epochs of $\ell = 1000$ means mining rewards are distributed on a weekly basis. This is longer than the seventeen hours Bitcoin miners wait today, but arguably still an acceptable time frame.

The threshold for a refunded safety-violation ($Free\ safety\text{-}violation\ threshold = 1$), is as in Bitcoin, but the non-refunded variation is twice as cheap ($Safety\text{-}violation\ threshold = 0.5$). We note that we are not aware of attacks of either type on prominent cryptocurrency systems, and that the non-refunded type is unlikely due to the lack of endogenous compensation (§6.4).

In regards to permissiveness, a miner with 10% budget that fails to obtain any *ic* due to incumbent censorship is expected to get 5% of what she would have had with *ic* ($Permissiveness = 0.05$). Recall this is a one-time entry cost (§6.5), and a qualitative improvement on a permissioned system.

Finally, for the current Bitcoin miner balance distribution [6] the maximal relative advantage from size differences is 0.1% ($Size\ bias = 0.001$). We consider modifications to further decrease this value in our extended report [63].

8 Conclusion

We propose a new PoW paradigm that utilizes internal expenditure as a balancing mechanism for the external impact. We present *HEB* – a generalization of Nakamoto's protocol that allows its designer to tune external resource expenditure. We formalize evaluation metrics including a blockchain's resilience to sabotage and revenue-seeking attacks and permissiveness on a continuous scale. We propose practical parameters based on Bitcoin's ecosystem that cut down by half the PoW expenditure (equivalent to reducing the power consumption of an entire country) while maintaining similar security guarantees against practical attacks.

Natural questions that arise from the introduction of *HEB* are what should be the security target for cryptocurrency protocols, how to set the parameters dynamically, and how to govern them [30]. Beyond these, *HEB* extends the design space of decentralized systems, and is a step forward in realizing secure PoW systems with a sustainable environmental impact.

References

1 Nick Arnosti and S Matthew Weinberg. Bitcoin: A natural oligopoly. *arXiv*, 2018. `arXiv:1811.08572`.

2 Shehar Bano et al. Sok: Consensus in the age of blockchains. In *AFT*, 2019.

3 Mathieu Baudet et al. State machine replication in the libra blockchain, 2018.

4 Jörg Becker et al. Can we afford integrity by proof-of-work? In *WEIS*, 2012.

5 Bitcoin.com. Etc team finally acknowledges the 51% attack on network, 2020.

6 blockchain.info. Hashrate distribution, 2020. URL: `https://tinyurl.com/55nb9w2v`.

7 Joseph Bonneau. Why buy when you can rent? In *FC*, 2016.

8 Joseph Bonneau et al. Research perspectives on Bitcoin and second-generation cryptocurrencies. In *Symposium on Security and Privacy*, 2015.

9 Timothy C Brock. Implications of commodity theory for value change. In *Psychological foundations of attitudes*, pages 243–275. Elsevier, 1968.

10 Vitalik Buterin. Ethereum whitepaper, 2013.

11 Miles Carlsten et al. On the instability of bitcoin without the block reward. In *CCS*, 2016.

12 Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

13 Xi Chen et al. An axiomatic approach to block rewards. In *AFT*, 2019.

14 Alex de Vries. Bitcoin's energy consumption is underestimated: A market dynamics approach. *Energy Research & Social Science*, 2020.

15 C. Decker and R. Wattenhofer. Information propagation in the Bitcoin network. In *P2P*, 2013.

16 Evangelos Deirmentzoglou, Georgios Papakyriakopoulos, and Constantinos Patsakis. A survey on long-range attacks for proof of stake protocols. *IEEE Access*, 7:28712–28725, 2019.

17 digiconomist. Bitcoin energy consumption, 2021. URL: `https://tinyurl.com/ehvat7fz`.

18 John R Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, 2002.

19 C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In *Crypto*, 1992.

20 Ittay Eyal and Emin Gün Sirer. Majority is not enough. In *FC*, 2014.

21 Peter Fairley. Blockchain world-feeding the blockchain beast if bitcoin ever does go mainstream, the electricity needed to sustain it will be enormous. *IEEE Spectrum*, 2017.

22 Amos Fiat et al. Energy equilibria in proof-of-work mining. In *EC*, 2019.

23 Cambridge Centre for Alternative Finance. Bitcoin electricity consumption index, 2019.

24 Ethereum Foundation. Ethereum 2, 2019. URL: `https://tinyurl.com/srr67va`.

25 Chaya Ganesh et al. Virtual asics: Generalized proof-of-stake mining in cryptocurrencies. ePrint, 2020.

26 Juan A. Garay et al. The Bitcoin backbone protocol. In *Eurocrypt*, 2015.

27 Peter Gaži et al. Stake-bleeding attacks on proof-of-stake blockchains. In *CVCBT*, 2018.

28 Arthur Gervais et al. Security and performance of proof of work blockchains. In *CCS*, 2016.

29 Yossi Gilad et al. Algorand. In *SOSP*, 2017.

30 LM Goodman. Tezos: a self-amending crypto-ledger white paper, 2014.

31 Guy Goren and Alexander Spiegelman. Mind the mining. In *EC*, 2019.

32 Toby Hill. Blackrock goes green? investment giant joins climate action, 2020.

33 Nicolas Houy. Rational mining limits bitcoin emissions. *Nature Climate Change*, 2019.

34 G Huberman et al. Monopoly without a monopolist: An economic analysis of the bitcoin payment system. *Social Science Research Network*, 2017.

35 Digital Currency Initiative. 51% attacks, 2020. URL: `https://dci.mit.edu/51-attacks`.

36 Markus Jakobsson and Ari Juels. Proofs of work and bread pudding protocols. In *Secure information networks*, pages 258–272. Springer, 1999.

37 Ari Juels. Client puzzles: A cryptographic countermeasure against connection depletion attacks. In *NDSS*, 1999.

38 Ghassan Karame et al. Double-spending fast payments in bitcoin. In *CCS*, 2012.

39 Aggelos Kiayias et al. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Crypto*, 2017.

40 Lucianna Kiffer et al. A better method to analyze blockchain consistency. In *CCS*, 2018.

41 Eleftherios Kokoris Kogias et al. Enhancing bitcoin security and performance with strong consistency via collective signing. In *USENIX*, 2016.

42 Joshua A Kroll et al. The economics of Bitcoin mining or, Bitcoin in the presence of adversaries. In *WEIS*, 2013.

43 Ben Laurie and Richard Clayton. Proof-of-work proves not to work. In *WEIS*, 2004.

44 K. Liao and J. Katz. Incentivizing blockchain forks via whale transactions. In *FC*, 2017.

45 Michael Lynn. Scarcity effects on value: A quantitative review of the commodity theory literature. *Psychology & Marketing*, 1991.

46 Patrick McCorry et al. Smart contracts for bribing miners. In *FC*, 2018.

47 Andrew Miller et al. Nonoutsourceable scratch-off puzzles to discourage bitcoin mining coalitions. In *CCS*, 2015.

48 Michael Mirkin et al. Bdos: Blockchain denial-of-service. In *CCS*, 2020.

49 Camilo Mora et al. Bitcoin emissions alone could push global warming above 2 c. *Nature Climate Change*, 2018.

50 Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

51 Arvind Narayanan et al. *Bitcoin and cryptocurrency technologies: a comprehensive introduction.* Princeton, 2016.

52 Kartik Nayak et al. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *IEEE European SP*, 2016.

53 T. Neudecker and H. Hartenstein. Short paper: An empirical analysis of blockchain forks in bitcoin. In *FC*, 2019.

54 BBC news. Tesla will no longer accept bitcoin over climate concerns, says musk, 2021.

55 Shunya Noda et al. An economic analysis of difficulty adjustment algorithms in proof-of-work blockchain systems. In *EC*, 2020.

56 Andrew Poelstra et al. Distributed consensus from proof of stake is impossible. *Self-published Paper*, 2014.

57 Jamie Redman. Bitcoin gold 51% attacked - network loses $70,000 in double spends, 2020.

58 Tim Roughgarden. Transaction fee mechanism design. *arXiv*, 2021.

59 Ayelet Sapirshtein et al. Optimal selfish mining strategies in Bitcoin. In *FC*, 2016.

60 Jon Truby. Decarbonizing bitcoin: Law and policy choices for reducing the energy consumption of blockchain technologies and digital currencies. *Energy research & social science*, 2018.

61 Itay Tsabary et al. Mad-htlc: Because htlc is crazy-cheap to attack. *S&P*, 2021.

62 Itay Tsabary and Ittay Eyal. The gap game. In *CCS*, 2018.

63 Itay Tsabary, Alexander Spiegelman, and Ittay Eyal. Tuning pow with hybrid expenditure – extended version. *arXiv preprint*, 2021. `arXiv:1911.04124`.

64 Aviv Yaish and Aviv Zohar. Pricing asics for cryptocurrency mining. *arXiv*, 2020. `arXiv:2002.11064`.