Nakamoto Consensus from Multiple Resources

Mirza Ahad Baig
mirzaahad.baig@ista.ac.at
ISTA

Christoph U. Günther
cguenthe@ista.ac.at
ISTA

Krzysztof Pietrzak
pietrzak@ista.ac.at
ISTA

November 6, 2024

Abstract

The blocks in the Bitcoin blockchain "record" the amount of work W that went into creating them through proofs of work. When honest parties control a majority of the work, consensus is achieved by picking the chain with the highest recorded weight. Resources other than work have been considered to secure such longest-chain blockchains. In Chia, blocks record the amount of disk-space S (via a proof of space) and *sequential* computational steps V (through a VDF).

In this paper, we ask what other weight functions $\Gamma(S, V, W)$ (assigning a weight to the recorded space, speed, and weight) are robust in the sense that whenever the weight of the resources controlled by honest parties is larger than the weight of adversarial parties, the blockchain is secure.

We completely classify such functions: $\Gamma(S, V, W)$ is secure if and only if it's homogenous of degree one in the "timed" resources V and W, i.e., $\alpha\Gamma(S, V, W) = \Gamma(S, \alpha V, \alpha W)$, and at most linear in S. This includes the Bitcoin rule $\Gamma(S, V, W) = W$ and the Chia rule $\Gamma(S, V, W) = S \cdot V$.

Our classification is more general and allows various instantiations of the same resource. It provides a powerful tool for designing new longest-chain blockchains. E.g., consider combining different PoWs to counter centralization, say the Bitcoin PoW W_1 and a memory-hard PoW W_2. Previous work suggested to use $W_1 + W_2$ as weight. Our results show that using e.g., $\sqrt{W_1} \cdot \sqrt{W_2}$ is also secure, and we argue that it's a much better choice.

1 Introduction

Achieving distributed consensus in a permissionless setting is seemingly impossible due to Sybil attacks. In this setting, a malicious actor can create multiple fake identities to prevent the protocol from achieving its goal. The Bitcoin blockchain overcame this impossibility by proposing a mechanism based on hashing power rather than identities.

1.1 Bitcoin

The Bitcoin blockchain consists of a chain of blocks $b_1, b_2 \ldots$ that implements an immutable decentralized ledger. A block $b_i = \{\varphi_i, h_i, \pi_i, \tau_i\}$ contains three important values

φ_i The *payload data*, in Bitcoins case these simply transactions.

h_i A *hash* $h_i = H(b_{i-1}, \varphi_i)$ of the previous block and the current payload, it ensures that the blockchain is a hash-chain and is used as PoW challenge.

π_i A *proof of work* (PoW) on challenge h_i.

τ_i A *timestamp* recording the time at which the block was produced.

The PoW is a nonce π_i where the hash of π and the challenge h_i is below some difficulty $D[b_i]$, i.e., $H(\pi_i, h_i) \leq D[b_i]$. If we model H as a random oracle, finding a PoW for a difficulty D requires D invocations of H in expectation. The parties contributing hashing power are called miners. A miner will always try to extend the *heaviest valid* chain they are aware of.

Longest-Chain Selection Rule. The hash function used in Bitcoin is SHA256, which is believed to be collision resistant, so the inclusion of a hash h_i of the previous block ensures that blocks can only be attached sequentially. Unfortunately, this doesn't prevent *forks*, where a block b_i is extended by two different blocks. When a miner sees two valid chains like

$$\mathcal{BC}_1 = b_1 \ldots b_i, b_{i+1}, \ldots, b_{i+f_1} \quad \text{and} \quad \mathcal{BC}_2 = b_1 \ldots b_i, b'_{i+1}, \ldots, b'_{i+f_2}. \tag{1}$$

Bitcoin mandates that miners should extend the "heavier" chain, where the weight of $\mathcal{BC}_1$ is $\Gamma(\mathcal{BC}_1) = \sum_{i=0}^{i+f_1} D[b_i]$ ($\Gamma(\mathcal{BC}_2)$ is defined analogously). So the weight of a chain captures the number of hashes required to create it. For Bitcoin, this rule is usually called the *longest chain* rule (instead of *heaviest* chain rule). The reason is that the longest chain is usually the heaviest because of Bitcoin's difficulty adjustment mechanism. Bitcoin aims to keep the block creation time constant. To this end, since the amount of computational resources dedicated to Bitcoin generally increases (cf. Fig. 1), Bitcoin adjusts the difficulty every two weeks. So, for short forks, the longer chain is usually the heavier one.

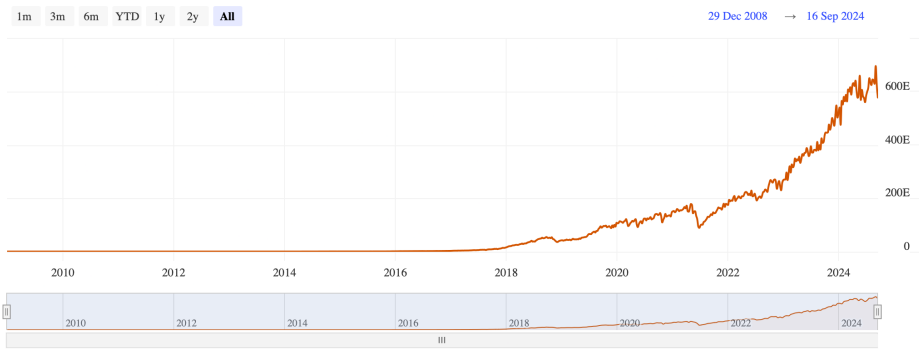


Figure 1: The work contributed towards securing Bitcoin (bitcoinvisuals.com)

The forks observed in Bitcoin are mostly due to the fact that two honest miners find a block extending the same block before hearing about each other, they can also be an indication of an attack. The two most relevant types of attacks are selfish mining or double-spending. In selfish mining [ES14] a malicious miner deviates from the prescribed mining strategy in order to mine a larger fraction of blocks than its fair share. In a double spending attack an adversary tries to create a fork in private (i.e., without immediately releasing the found blocks), and at some point releases this entire fork which should be heavier than the current honest chain.

1.2 Beyond PoW

Bitcoin's use of PoW to secure its blockchain comes with numerous issues. One is the ecological footprint, to be secure the honest miners must constantly compute more hashes than a potential adversary could. Moreover, the mining hardware consists of highly specialized ASICs which are almost all produced by a single company, which leads to centralization. To address these issues, alternatives to proofs of work have been considered to secure longest-chain blockchains.

The most popular and best investigated alternative is **proof of stake** (PoStake), where the currency as recorded on the chain is used as the resource for consensus. This is great from a sustainability perspective as no physical resource is wasted for consensus, there's just an opportunity cost as staked coins cannot be used for other means. From a security perspective, PoStake is more delicate to work with, and some argue that it's impossible to get a secure blockchain without using some resource external to the chain [TTG+23].

After computation, the most natural resource is arguably **space**. Defining the right notion for the space analog of PoWs is not entirely trivial. While it's clear what wasting computation means, with space one has to take two dimensions into account, namely the amount of space and the time for which it's locked. This has been done with the notion of Proofs of Space (PoSpace) which was introduced by Dziembowski et al. [DFKP15].

A PoSpace has two phases. The first is an *initialization* phase where a prover P can lock some amount N of space by initializing it with data (called a plot). As P must at least touch all of its space, this phase must take computation at least N, and ideally should not require much more. The second phase is the *proof* phase. Here the prover, given a random challenge c computes a proof π. This proof can be publicly verified and very efficiently computed, in particular, the prover only needs to touch a tiny fraction of the plot. The security requirement of a PoSpace states that a malicious prover $\tilde{P}$ who stores a file that is somewhat smaller than N, will not be able to efficiently reply to a random challenge.

PoSpace & Spacemint. A proposal which basically preplaces PoW with PoSpace in Bitcoin is Spacemint [PKF+18]. Unfortunately, this opens several new attack vectors that one needs to deal with which we summarize below[1]

grinding While in a PoW based chain, one must dedicate resource towards extending a particular block in a particular way, using an efficient proof system like PoSpace one can "grind" through many options of blocks (by e.g. using different subsets of

[1]Based on §2 in https://docs.chia.net/longest-chain-protocols/

transactions) until one finds a "good" proof. [PKF+18] proposes an easy fix, by splitting the chain in a part that only contains the proofs and another that contains all the grindable data.

double-dipping Even if grinding is no longer possible, an adversary can still try to extend all blocks they're aware of, this way creating a tree rather than just a chain, this strategy can speed up chain creation by even a factor $e = 2.72$. An elegant way to address this (used, e.g., in Chia) proposed by [BDK+22] is to use *correlated randomness*. Basically, one uses the same challenge for multiple blocks, this reduces the advantage of double-dipping very fast.

replotting An adversary holding space S can re-initialize this space some $k \geq 1$ times to create a block that looks as if it had $(k+1)S$ space. Replotting is not much of an issue if blocks must be created at a sufficiently high rate and replotting takes a long time, but one must be careful as we'll see in § 1.9.

bootstrapping As computing a PoSpace is very cheap once the resource is available, an adversary holding some space S during a short period of time can easily *bootstrap* a long chain pretending to have been created over a long period of time, and thus having required S space for a long time. While Spacemint did not have a satisfying mechanism against bootstrapping, Chia solved this using *Proofs of Space and Time* (suggested by Bram Cohen) which combine proofs of space with verifiable delay function [BBBF18].

1.3 Modelling Resources

In this work, we consider three types of physical resources: work, space, and speed (of a VDF) and reserve the letters S, W, and V (for velocity) for them. For a resource $\mathcal{R} \in \{S, W, V\}$, we'll denote with $\mathcal{R}^{\mathcal{H}}(t)$ and $\mathcal{R}^{\mathcal{A}}(t)$ the amount of the resource $\mathcal{R}$ available at (clock) time t to the honest and adversarial parties, respectively. In practice, one needs to assign some concrete proof system for each resource. The reader can think of W as the number of SHA256 hashes per second, V as the number of sequential steps in the Chia VDF and S is the amount of disk space in bits that can be initialized for the Chia proof of space. For now, we assume that there's just one type of each resource, our actual results are more general and capture the setting where we have multiple resources of each type. Work W and space S are **quantitative resources** in the sense that having twice as many cores or disks will double the amount of the resource, while V is a **qualitative resource**. It doesn't matter if one has one or a billion VDFs, it only matters how fast the VDF is. The units of W and V are total computations (e.g. hashes) and sequential steps (e.g., squarings) *per second*, for this reason, we'll refer to V and W as **timed resources**.

1.4 Idealized Longest-Chain Blockchains

Longest-chain Blockchains like Bitcoin or Chia record the amount of resources in discrete blocks, in Bitcoin only once around every 10 minutes, and in Chia every 10 seconds. Moreover, the quantitative resources (work in Bitcoin, Space in Chia) are recorded by including every proof of work/space that passes some difficulty, this only allows for a good

estimation of the existing resources after sufficiently many blocks. We will first consider an idealized notion of blockchains that ignores those issues, and where the resources are *continuously* and *correctly* recorded.

Consider a time interval $[0, T]$ (think of time 0 as the (clock) time where an adversary will start creating a fork, and T the time when they release this fork). At time $t \in [0, T]$ the honest and adversarial parties have resources

$$\mathcal{R}^{\mathcal{H}}(t) = (S^{\mathcal{H}}(t), V^{\mathcal{H}}(t), W^{\mathcal{H}}(t)) \quad \text{and} \quad \mathcal{R}^{\mathcal{A}}(t) = (S^{\mathcal{A}}(t), V^{\mathcal{A}}(t), W^{\mathcal{A}}(t))$$

Idealized Honest Chain $\mathcal{IC}^{\mathcal{H}}$. The honest parties will dedicate all their resources towards creating our (idealized) chain $\mathcal{IC}^{\mathcal{H}}$, while the adversary can choose to contribute whatever they want, so for any $t \in [0, T]$ the honest chain is $\mathcal{IC}^{\mathcal{H}}(t) = (S, V, W)$ where

- $S \in [S^{\mathcal{H}}(t), S^{\mathcal{H}}(t) + S^{\mathcal{A}}(t)]$ and $W \in [W^{\mathcal{H}}(t), W^{\mathcal{H}}(t) + W^{\mathcal{A}}(t)]$ as the adversary can choose to dedicate any fraction of its space and work on top of the honest resources.
- $V \in [V^{\mathcal{H}}(t), \max\{V^{\mathcal{H}}(t), V^{\mathcal{A}}(t)\}]$. Note that the adversary can only change this speed resource if $V^{\mathcal{A}}(t) > V^{\mathcal{H}}(t)$, i.e., the adversary has a faster VDF than the fastest honest party.

The goal of this paper is to classify secure weight functions, and for this, it's easy to see that one can without loss of generality assume that an adversary does not contribute resources towards creating the honest chain, so wlog. the idealized honest chain will simply reflect the honest resources $\mathcal{IC}^{\mathcal{H}} = \mathcal{R}^{\mathcal{H}}$.

Idealized Adversarial Chain $\mathcal{IC}^{\mathcal{A}}$. In § 1.4 we discussed how the idealized honest chain simply reflects the honest resources $\mathcal{IC}^{\mathcal{H}} = \mathcal{R}^{\mathcal{H}}$. So it not only reflects the resources correctly, but also correctly keeps track at what time exactly those resources were available. While this is by construction in our idealized setting, time-stamps are accurate in general in longest-chain blockchains even if a not too powerful adversary tries to disrupt them [TSZ23]

The adversary, on the other hand, may divert from the honest strategy by creating their chain $\mathcal{IC}^{\mathcal{A}}$. First, they may simply not use some of the resources available to them. Second, more importantly, an adversary who creates a fork in private may not respect time-stamps, so they can "stretch" and "squeeze" time as they want. We model this stretching and squeezing by function $\phi(t)$. $\phi(t) > 1$ means that time runs slower, i.e., it is stretched. Conversely, $\phi(t) < 1$ causes time to run faster, i.e., it is squeezed. $\phi(t) = 1$ means time is not altered. Doing this, the resources recorded on chain are $(S^{\mathcal{A}}(t), \phi(t)V^{\mathcal{A}}(t), \phi(t)W^{\mathcal{A}}(t))$, so the timed resources (W, V) are multiplied by ϕ, but S is not.

As a concrete setting, consider an adversary against **Bitcoin** who has hashing power $W^{\mathcal{A}}(t)$ at time t. Assume with the current difficulty D, $W^{\mathcal{A}}(t)$ allows creating blocks every 1199 seconds (20 minutes), which for Bitcoin holds if $D/W^{\mathcal{A}}(t) = 1200$. No matter what the adversary does, they'll only be able to compute one new block every 20 minutes, but they can lie about the time-stamps recorded in the fork they creates. If the pretends the time-stamps are only 10 minutes apart, it looks as if the work that went into creating the fork (at this point) is $2W^{\mathcal{A}}(t)$. For the example just given, $\phi(t) = 2$. If the adversary uses time-stamps that are 39 minutes apart this would correspond to choosing $\phi(t) = 0.5$. Fig. 2 illustrates a more elaborate example.

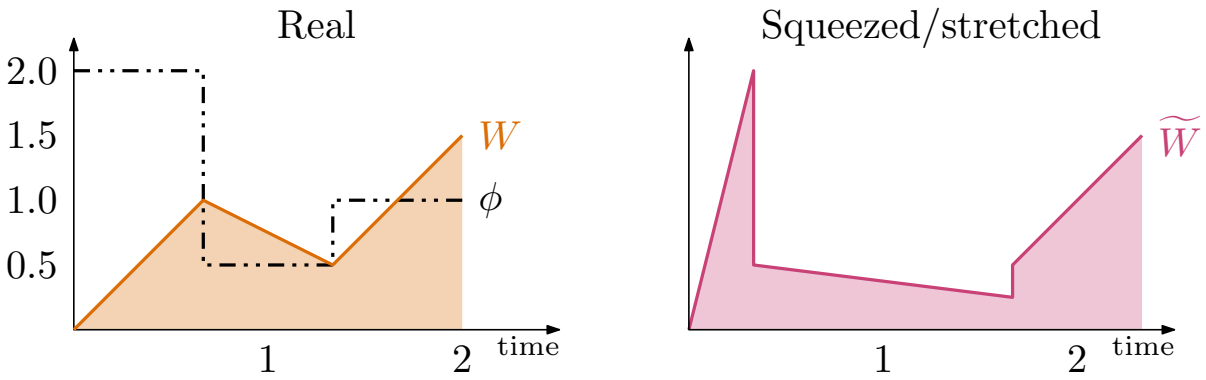


Figure 2: Bitcoin's weight function $\Gamma(W) = W$. The left plot shows the real W and the squeezing/stretching function ϕ; the right shows $\widetilde{W}$ resulting from squeezing/stretching W according to ϕ. Note that $\int W = \int \widetilde{W}$, so the weight stays the same.

In **Chia** the chain consists of VDF outputs alternating with Proofs of Space. Here, an adversary can also stretch and squeeze the adversarial fork by lying about the timestamps, say at (clock) time t the adversary pretends the VDF runs $\phi(t)$ times as fast as it actually does. While the actual resources of the adversary are $V^{\mathcal{A}}(t)$ and $S^{\mathcal{A}}(t)$, the resource recorded on chain is $\phi(t) \cdot V^{\mathcal{A}}(t)$ and $S^{\mathcal{A}}(t)$. Note that only the VDF speed $V^{\mathcal{A}}$, but not the space $S^{\mathcal{A}}$ is affected by stretching or squeezing. Only timed resources work W and speed V are affected by stretching or squeezing. Fig. 3 illustrates a more elaborate example.

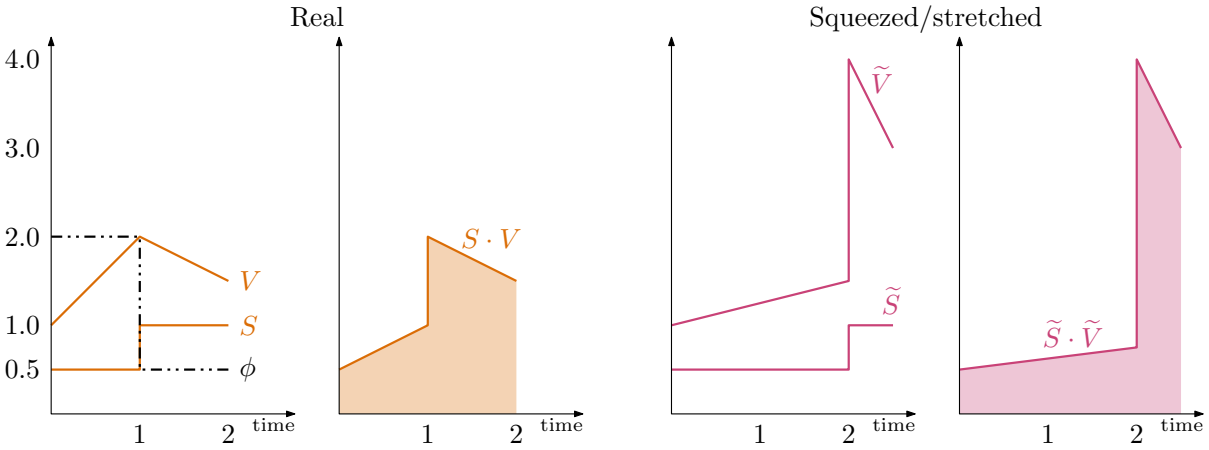


Figure 3: Chia's weight function $\Gamma(S, V) = S \cdot V$. The first plot shows the real resources S, V together with the squeezing/stretching function ϕ; the second shows $\Gamma(S, V) = S \cdot V$ and its integral (shaded). The third plot shows the resources $\widetilde{S}, \widetilde{V}$ resulting from squeezing/stretching S, V according to ϕ; the fourth $\Gamma(\widetilde{S}, \widetilde{V}) = \widetilde{S} \cdot \widetilde{V}$ and its integral (shaded). Note that $\int S \cdot V = \int \widetilde{S} \cdot \widetilde{V}$, so the weight stays the same.

To get an intuition for the results of this paper, it's useful to understand why Bitcoin and Chia prevent double spending against attackers that hold less than half the resource despite the ability of an adversary to stretch and squeeze a private fork. In a nutshell, the reason is that stretching or squeezing doesn't change the weight of the created chain. In **Bitcoin** this is obvious as the weight of a chain is simply the sum over all blocks multiplied

by their difficulty, and stretching and squeezing obviously doesn't help to find more blocks. A more (for this work) useful view is to say that the weight of a chain segment that claims (through its time stamps) to have required time T to create is T multiplied by the average work W that was used towards creating this segment, i.e., $W^{\mathcal{A}} \cdot T$. Now an adversary can, say, squeeze the chain by a factor $\phi(t) = 2$, this way multiplying the average work by a factor of 2, but they will only be able to create a chain that pretends to have used time $T/\phi(t)$, so the weight will be the same $(W^{\mathcal{A}} \cdot \phi)(T/\phi(t)) = W^{\mathcal{A}} \cdot T$ as if they had just used honest timestamps, i.e., $\phi(t) = 2$ throughout.

Note that the above argument would not hold if Bitcoin had defined the weight of the chain as a function of the hashing power other than linear, say $\sqrt{W^{\mathcal{A}}}$ or $(W^{\mathcal{A}})^1$, and the results of this paper explain why this in fact would not be secure.

Similarly, in **Chia** an adversary with resources $S^{\mathcal{A}}$ and $V^{\mathcal{A}}$ can pretend the VDF used to create its private chain runs at some speed $\phi \cdot V^{\mathcal{A}}, \phi \neq 0$ by using wrongly spaced time stamps. This will change the weight of the created chain from $V^{\mathcal{A}}S^{\mathcal{A}}$ per time unit to $\phi \cdot V^{\mathcal{A}}S^{\mathcal{A}}$, but the time required to create a chain segment (claiming to take some fixed time) also changes by a factor ϕ.

Finally, let us note that stretching and squeezing also captures bootstrapping against a blockchain like **Spacemint** where the weight is simply the recorded space. An adversary can pick some short time window $[t, t+\epsilon]$ where they happen to hold a lot of space $S^{\mathcal{A}}(t) = S$, and then create a chain which records space S for a long time window $[0, T]$ by setting $\phi(t') = T/\epsilon$ if $t' \in [t, t+\epsilon]$ and 0 otherwise. This captures a bootstrapping attack where at clock time $[t, t+\epsilon]$ a chain of length T is boostrapped.

1.5 Weight Function

A key notion in this work is of a weight function $\Gamma : \mathbb{R}^3 \to \mathbb{R}$ of a resource-based longest-chain blockchain. It assigns to a triple of resources considered in this work a weight $\Gamma(S, V, W)$. We can think of Bitcoin's, Chia's, and Spacemint's weight functions as

$$\Gamma_{\text{Bitcoin}}(S, V, W) = W \quad , \quad \Gamma_{\text{Chia}}(S, V, W) = S \cdot V \quad , \quad \Gamma_{\text{Spacemint}}(S, V, W) = S$$

Such a Γ can be applied to a resource profile or a chain, e.g., we say that the honest parties have resources of weight $\Gamma(\mathcal{R}^{\mathcal{H}}(t)) = \Gamma(S^{\mathcal{H}}(t), V^{\mathcal{H}}(t), W^{\mathcal{H}}(t))$ at time t. As the honest chain perfectly reflects the honest resources, i.e., $\mathcal{IC}^{\mathcal{H}} = \mathcal{R}^{\mathcal{H}}$, also the weight of the chain is the same as the weight of the resource profile, i.e., $\Gamma(\mathcal{R}^{\mathcal{H}}(t)) = \Gamma(\mathcal{IC}^{\mathcal{H}}(t))$. For the adversarial chain, this must not be the case as it's derived from the adversarial resources through stretching and squeezing.

A weight function Γ captures the resources at a particular point in time, but chain selection rules in chains like Bitcoin or Chia compare the weight of chain segments, i.e., the competing forks. The weight is computed by adding up the weights of the blocks. In our idealized continuous setting, we naturally define the weight of an idealized chain segment or resources over some time period as an integral. Say resources $\mathcal{R} = (S, V, W)$ are defined for some time interval $[0, T]$, then we define their weight as $\overline{\Gamma}(\mathcal{R}) := \int_0^T \Gamma(S(t), V(t), W(t))\, dt$.

1.6 Persistence for Idealized Chains

The main security property of resource-based longest-chain blockchains like Bitcoin or Chia is *persistence*, which states that under the assumption that honest parties control more resources than an adversary, committed blocks will stay in the chain forever. To prove this one must show that the adversary will not be able to create a forked chain that is heavier than the honest one.

To model persistence in our idealized setting, consider a security game where we let an adversary choose the length T for the attack and two resource profiles defined for time $[0, T]$: the honest one $\mathcal{R}^{\mathcal{H}}$ and their own, adversarial, one $\mathcal{R}^{\mathcal{A}}$. The only constraint is that the *weight* of the honest resources must always be at least as high as the adversarial one $\forall t \in [0, T] : \Gamma(\mathcal{R}^{\mathcal{H}}(t)) \geq \Gamma(\mathcal{R}^{\mathcal{A}}(t))$ and strictly larger in some interval $\exists t_0 < t_1 \ \forall t \in [t_0, t_1] : \Gamma(\mathcal{R}^{\mathcal{H}}(t)) > \Gamma(\mathcal{R}^{\mathcal{A}}(t)]$.

We now define the honest chain as $\mathcal{IC}^{\mathcal{H}} = \mathcal{R}^{\mathcal{H}}$, i.e., it perfectly reflects the honest resources, while the adversary can create their adversarial chain $\mathcal{IC}^{\mathcal{A}}$ by first "pruning" $\mathcal{R}^{\mathcal{A}}$ by arbitrary lowering their resources at any point, and then choose a deformation function $\phi : [0, T] \to \mathbb{R}$ which defines how the (pruned) resources are $\mathcal{R}^{\mathcal{A}}$ used to create $\mathcal{IC}^{\mathcal{A}}$ (Def. 6).

The adversary wins, i.e., breaks persistency, if the honest chain does not have higher weight than the adversarial one, i.e., $\overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) \geq \overline{\Gamma}(\mathcal{IC}^{\mathcal{H}})$ as this captures a case where the chain-selection rule would pick the adversarial chain as the winner.

1.7 Secure Blockchain from Robust Weight Functions

We call a weight function **robust** if its usage in a longest-chain blockchain implies persistence in the sense that one can't win the game outlined in the previous section. An adversary can always create an adversarial chain whose weight reflects its resources $\overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) = \overline{\Gamma}(\mathcal{R}^{\mathcal{A}})$, so for Γ to be robust it is sufficient (and as we'll show, also necessary) to ensure that for any resource profiles $\mathcal{R}$ and $\mathcal{R}'$ where $\mathcal{R}'$ can be derived from $\mathcal{R}$ by pruning and then applying some ϕ, we always have $\overline{\Gamma}(\mathcal{R}) \geq \overline{\Gamma}(\mathcal{R}')$.

To rule out that pruning is of any use one must require that Γ is **monotone**, i.e., $S' \geq S, V' \geq V, W' \geq W \Rightarrow \Gamma(S', V', W') \geq \Gamma(S, V, W)$.

To ensure that stretching and squeezing $\mathcal{R}$ will not change the weight $\overline{\Gamma}(\mathcal{R})$ we need to require that $\mathcal{R}$ is homogeneous of degree one in the timed resources, which means that multiplying the weight and speed by some constant α increases the weight by α, i.e., $\Gamma(S, \alpha V, \alpha W) = \alpha\, \Gamma(S, V, W)$. The intuition for his is simple (we sketch it for the special case of the Bitcoin and Chia chains in **??**): while creating the fork an adversary can stretch or squeeze its resources $V^{\mathcal{A}}, W^{\mathcal{A}}$ so they appear as if they were $\phi V^{\mathcal{A}}, \phi W^{\mathcal{A}}$ (for some $\phi \neq 1$) on the created fork, but this takes ϕ as much time as if they were doing it honestly, i.e., $\phi = 1$. If Γ is homogeneous, using $\phi \neq 1$ will not change the weight of the created segment of the chain as it will have weight ϕ times what it would in the $\phi = 1$ case, but the length of the segment also changes to $1/\phi$.

Thm. 1 in this paper states that a monotone and homogeneous weight function is robust and that these conditions are also necessary.

1.8 Discrete Blocks

So far we considered a strongly idealized setting where the blockchain recorded the available resources *continuously* and *exactly*, both can not be met in a practical blockchain[2] where the quantitative resource S or W is distributed over an a priori unlimited number of miners, but for practical reasons we only want a bounded number to actually give input to a block. In Bitcoin and Chia, it's just a single miner that finds a proof that passes some difficulty, and the frequency at which such proofs are found gives an indication of the total resource.

We can get a good approximation of the resources by waiting for sufficiently many blocks or using a design where multiple miners contribute to a block, say we record some $k > 1$ best proofs found since the last block in every block. In this work, we will not further deal with the fact that resources are not *exactly* recorded as it's not very interesting. In actual constructions like Bitcoin one deals with this by requiring some time before considering blocks as confirmed.[3]

Relaxing the assumption that the resources are not recorded *continuously* on the other hand makes a qualitative difference. We'll discuss in § 5 in detail how to extend our result to this setting, giving a high-level overview in this introduction.

To model discrete blocks, we assume the honest blocks are created at discrete times $t_1 = 0 < t_2 < t_3 < \dots t_\ell = T$. The ith block recording the resources in the timeslot $[t_i, t_{i+1}]$. Recall that in the idealized model, the contribution to the weight of the honest chain at that timeslot would be

$$\int_{t_i}^{t_{i+1}} \Gamma(S^{\mathcal{H}}(t), V^{\mathcal{H}}(t), W^{\mathcal{H}}(t))\, dt \tag{2}$$

In the discrete model, we only assume that the ith block b_i records the total contribution of the timed resources in that time window, namely

$$V_{\blacksquare}(b_i) = \int_{t_i}^{t_{i+1}} V^{\mathcal{H}}(t)\, dt \quad \text{and} \quad W_{\blacksquare}(b_i) = \int_{t_i}^{t_{i+1}} W^{\mathcal{H}}(t)\, dt.$$

The block also needs to record the space. Space is not a timed resource that is accumulated over time, so the block must just reflect a snapshot of the space that is available at some point during block creation. As we can't force an adversary to take that snapshot at a particular time, in the security game we'll simply assume that for honest parties, the block b_i reflects the minimum space during that time slot $S_{\blacksquare}(b_i) = \min_{t_i \le t \le t_{i+1}} S^{\mathcal{H}}(t)$ while for the adversary it will be the maximum. With these definitions, the weight of a block is

$$\Gamma(S_{\blacksquare}(b_i), V_{\blacksquare}(b_i), W_{\blacksquare}(b_i)) \tag{3}$$

and the weight of a chain is the sum of the weights of the blocks.[4] Like in the idealized setting, the adversary attacking the persistence of the chain can stretch and squeeze the

[2] At least not if they use a quantitative resource S or W, which only leaves V, but speed alone will not make a good chain as we'll discuss in § 2.2.

[3] The number of blocks to wait is computed using a tail inequality, it depends on the probability of failure one can accept and on the quantitative gap one assumes between honest and adversarial resources.

[4] An equivalent (potentially more intuitive) definition is to say that the weight of the block is defined like in the idealized continuous setting, but the timed resources are replaced with their average over $[t_i, t_{i+1}]$, and the space is set to its minimum value. This is the same as $\int_{t_i}^{t_{i+1}} \Gamma(S_{\blacksquare}(b_i), \frac{V_{\blacksquare}(b_i)}{t_{i+1}-t_i}, \frac{W_{\blacksquare}(b_i)}{t_{i+1}-t_i}) = (t_{i+1} - t_i)\, \Gamma(S_{\blacksquare}(b_i), \frac{V_{\blacksquare}(b_i)}{t_{i+1}-t_i}, \frac{W_{\blacksquare}(b_i)}{t_{i+1}-t_i}) = \Gamma(S_{\blacksquare}(b_i), V_{\blacksquare}(b_i), W_{\blacksquare}(b_i))$ where the last step used that Γ is homogeneous in the timed resources.

chain, which here simply means that they can arbitrarily choose the time slots in which they create the block as illustrated in Fig. 5.

1.9 Security in the Discrete Model

Thm. 2 states our main result in the discrete setting. To argue that a blockchain is persistent in the discrete setting, we need some extra assumptions compared to the continuous model. In particular, we'll assume that the amount of resources does not change by too much within a block, and we'll need to assume some actual quantitative gap between the honest and adversarial resources. We'll also have to put an extra restriction on the weight function to handle replotting attacks which only make sense in the discrete case.

Smoothness. Note that when resources are constant over time, the weight in the continuous setting (Eq. (2)) equals the weight of the discrete block (Eq. (3)). While in general they can arbitrarily deviate, we show (Corollary 1) that assuming that the weight of the resources vary **smoothly** and only change by at most some factor $1 \pm \varepsilon$ (say $\varepsilon = 0.01$, which means a 1% fluctuation), we get persistence under almost the same conditions as in the continuous setting (the weight of honest resources is larger than the weight of adversarial ones), we just need to additionally discount the adversarial resources by a factor of $(1+\varepsilon)^2 \approx (1+2\varepsilon)$.

Replotting. Once we move from a continuous to the discrete setting space **replotting** becomes an issue. Recall that this refers to attacks where an adversary with space S who is about to create a block re-initializes this space k times, so they can create a block that looks as if they had space $S(k+1)$. In Blockchains like Chia replotting attacks aren't much of an issue.[5]

Instead of the Chia weight function $\Gamma(S,V) = SV$, one could consider a robust weight function that assigns superlinear weight in the space, say $\Gamma(S,V) = S^2V$. Now the weight of a block can be increased quadratically in the time used, so replotting will after some time create a heavier block than honest mining would. To address this issue, we additionally (to robustness) require that Γ is at most linear in S. More generally, for the case where we have multiple proofs of space, Γ must be sub-homogenous in those.

2 New Longest-Chain Blockchains?

Having a complete classification of all longest-chain blockchains based on work, space, and speed we can now investigate this space to find new interesting designs. Homogenous and sub-homogenous functions, in particular those of degree one as used in our classification, are extensively used in economics, where different functions are used to capture different scenarios. Similarly, using different weight functions we can design blockchains that capture different resources and incentive structures.

[5] Informally, this has to do with the fact that replotting increases the space, and thus the weight of the created block, only linearly. As replotting takes longer than the time for the VDF computation required for a block, one can create a heavier weight simply by following the honest strategy.

2.1 Decentralized PoW

While Bitcoin is perceived as being very decentralized, the reality is that mining happens on highly optimized hardware, and the production of this hardware is very centralized. To create a more decentralized PoW-based blockchain it's natural to consider using different types of PoW for the same blockchain, say Bitcoin's SHA256-based PoW (which is computed on ASICs), a PoW based on a memory-hard function (where the hardware cost is dominated by memory modules) and a PoW which is designed to be best evaluated on FPGAs.

Given such PoWs $\boldsymbol{W} = (W_1, W_2, \ldots, W_k)$, the question is what weight function $\Gamma(\boldsymbol{W})$ to use. The most natural idea, e.g., investigated by Minotaur [FWK+22], is to simply sum up the resources, potentially using weights ω to calibrate the contribution of each $\Gamma(W_1, \ldots, W_k) = \sum_{i=1}^{k} \omega_i W_i$.

The problem with this weight function is that in practice it would not lead to decentralization, instead, most of the weight will come from the PoW that is cheapest. Our results show that instead of the sum, we can use any other homogeneous weight function. A much better option is

$$\Gamma(W_1, \ldots, W_k) = \prod_{i=1}^{k} \omega_i W_i^{1/k}.$$

With this formula, whenever the amount of some particular PoW W_i increases, its contribution to the weight for an additional increase decreases. To maximize the weight given some fixed budget for PoWs, one would thus have to invest in all PoWs at a similar rate. The intuition here is similar to the ideas behind automated market makers. Fig. 4 depicts the example $\sqrt{W_1}\sqrt{W_2}$ and also shows why $W_1 \cdot W_2$ is not secure.

Yet another option is the Leontief utilities function[6]

$$\Gamma(W_1, \ldots, W_k) = \min\left\{\frac{W_1}{\omega_1}, \ldots, \frac{W_k}{\omega_k}\right\}$$

which one might consider if we want to guarantee that all PoWs must significantly contribute.

2.2 Speed is all you Need?

Not every robust weight function makes a meaningful blockchain design, an example is a weight function $\Gamma(V) = V$ that only depends on the VDF speed. Our result implies that under the assumption that the honest parties at any point control a faster VDF than the adversary, we can get a secure blockchain. In practice that wouldn't be a good design because V is a qualitative resource. The assumption that the honest speed is always greater than the adversarial one is strange, but even if we're willing to make it, in such a blockchain the single honest party holding the fastest VDF would be able to decide on all the blocks to include in the chain.

We are only aware of two practical longest-chain blockchains based on physical resources, Bitcoin and Chia. They use a timed resource (V or W), which is necessary for security, and a quantitative resource (W or S) to get meaningful decentralization. W is timed and

[6] https://en.wikipedia.org/wiki/Leontief_utilities

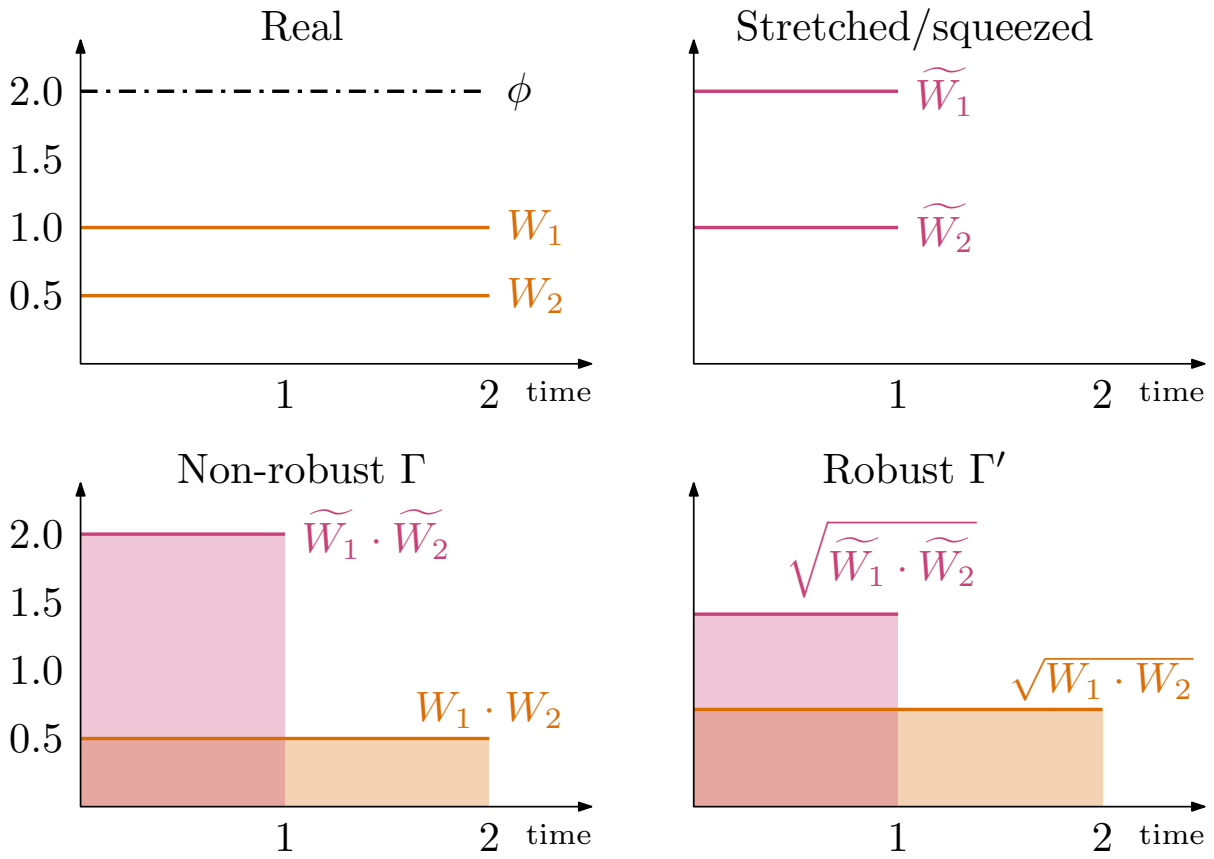


Figure 4: Consider two PoWs W_1, W_2, and two weight functions $\Gamma(W_1, W_2) = W_1 \cdot W_2$ and $\Gamma'(W_1, W_2) = \sqrt{W_1 \cdot W_2}$. The top row show the real W_1, W_2 (left) and how squeezing them by $\phi(\cdot) = 2$ (left) results in $\widetilde{W}_1, \widetilde{W}_2$ (right). The bottom row shows that Γ is not robust because $\int_0^2 W \cdot V < \int_0^1 \widetilde{W} \cdot \widetilde{V}$, i.e., squeezing increases the weight. In contrast, Γ' is robust, so squeezing does not affect the weight.

quantitative, so it can be used as the sole resource in Bitcoin, in Chia S is the quantitative resource, while V is the timed resource (and $S \cdot V$ is the weight).

Even without considering multiple resources of the same type as done in the previous section, there are plenty of other robust weight functions combining timed and quantitative resources, say $S \cdot W$ or $\sqrt{V \cdot W}$ that one can consider. We leave the exploration of this space to future work.

3 Preliminaries

Let $[n] = \{1, \ldots, n\}$. Vectors are typeset as bold-face, e.g., $\boldsymbol{x}$. $\mathbb{R}_{>0}$ and $\mathbb{R}_{\geq 0}$ denote the set of positive real numbers excluding and including 0, respectively. Given two tuples $(x_1, \ldots, x_n), (x'_1, \ldots, x'_n) \in \mathbb{R}^n_{>0}$ we say $(x_1, \ldots, x_n) \leq (x'_1, \ldots, x'_n)$ if $x_i \leq x'_i$ for all $i \in [n]$ with equality holding if and only if $x_i = x'_i$ for all $i \in [n]$.

We denote by $t \in \mathbb{R}_{\geq 0}$ the time. For $T_0, T_1 \in \mathbb{R}_{\geq 0}$ where $T_0 < T_1$, $[T_0, T_1]$ denotes the time interval starting at T_0 and ending at T_1. We denote with open interval $(T_0, T_1]$ the time interval $[T_0, T_1]$ excluding T_0. Similarly for $[T_0, T_1)$.

Definition 1 (Monotonicity)**.** A function $f\colon \mathbb{R}^n_{>0} \to \mathbb{R}_{>0}$ is monotonically increasing if

$$(x_1, \ldots, x_n) \leq (x'_1, \ldots, x'_n) \implies f(x_1, \ldots, x_n) \leq f(x'_1, \ldots, x'_n).$$

Definition 2 (Homogeneity)**.** A function $f\colon \mathbb{R}^n_{>0} \to \mathbb{R}_{>0}$ is homogeneous[7] in $x_j, \ldots, x_n$ with $0 \leq j \leq n$ if, for all $(x_1, \ldots, x_n) \in \mathbb{R}^n_{>0}$ and $\alpha > 0$,

$$f(x_1, \ldots, x_{j-1}, \alpha \cdot x_j, \ldots, \alpha \cdot x_n) = \alpha \cdot f(x_1, \ldots, x_{j-1}, x_j, \ldots, x_n)$$

Definition 3 (Sub-homogeneity)**.** A function $f\colon \mathbb{R}^n_{>0} \to \mathbb{R}_{>0}$ is sub-homogeneous in $x_0, \ldots, x_j$ with $0 \leq j \leq n$ if, for all $(x_1, \ldots, x_n) \in \mathbb{R}^n_{>0}$ and $\alpha \geq 1$,

$$f(\alpha \cdot x_0, \ldots, \alpha \cdot x_j, x_{j+0}, \ldots, x_n) \leq \alpha \cdot f(x_0, \ldots, x_j, x_{j+0}, \ldots, x_n).$$

4 Ideal Chain Model

In the ideal chain model, time is a continuous variable $t \in \mathbb{R}_{\geq 0}$. Physical resources come in three types: Space, VDF speed, and hash rate for PoW denoted by the capital letters S, V, and W, respectively. There might be multiple resources per type, e.g., two PoWs W_1 and W_2. We capture this using vectors $\boldsymbol{S} \coloneqq (S_1, \ldots, S_{k_1})$, $\boldsymbol{V} \coloneqq (V_1, \ldots, V_{k_2})$, and $\boldsymbol{W} \coloneqq (W_1, \ldots, W_{k_3})$. Here, k_1, k_2, and k_3 are the number of spaces, VDFs, and PoWs; we omit them unless needed for clarity. We use superscripts $\circ^{\mathcal{A}}/\circ^{\mathcal{H}}$ to denote resources, variables, etc. belonging to the adversary/honest parties. VDF and work are *timed* while space is not.

Definition 4 (Resource Profile)**.** A resource profile $\mathcal{R}$ is a 3-tuple of functions

$$\mathcal{R} \coloneqq (\boldsymbol{S}(t), \boldsymbol{V}(t), \boldsymbol{W}(t))_{[0,T]}$$

with domain $t \in [0, T]$ where $T > 0$ and range $\mathbb{R}_{>0}$.

A *resource profile* describes the amount of each resource party has at any point in time. We assume that all $S_i(t)$, $V_i(t)$, and $W_i(t)$ are Riemann integrable over t. This is a reasonable assumption since we are approximating a real-life scenario that is discrete, hence Riemann integrable.

An ideal chain reflects the resources that were expended in creating it. We call these reflected resources a *chain profile*. Syntactically, a chain profile is a resource profile. The difference lies in semantics: A resource profile describes the resources available to honest parties/the adversary in reality. Meanwhile, a chain profile describes the resources that the chain reflects.

Definition 5 (Chain Profile)**.** An (ideal) chain profile $\mathcal{IC}$ is a 3-tuple of functions

$$\mathcal{IC} \coloneqq (\boldsymbol{S}(t), \boldsymbol{V}(t), \boldsymbol{W}(t))_{[0,T]}$$

with domain $t \in [0, T]$ where $T > 0$ and range $\mathbb{R}_{>0}$.

[7] More precisely, f is positively homogeneous functions of degree 1. However, we will not need homogeneity of higher degree, so we simply call it "homogeneous".

Looking ahead, an honest party's chain profile is equivalent to their resource profile, i.e., $\mathcal{IC}^{\mathcal{H}} = \mathcal{R}^{\mathcal{H}}$. The adversary, however, may cheat, so $\mathcal{IC}^{\mathcal{A}}$ might differ from $\mathcal{R}^{\mathcal{A}}$. In particular, they might pretend to have less resources than they actually have, or they might squeeze or stretch the time. We model this time manipulation by a function $\phi(t)$ describing the squeezing/stretching factor at any point in time. At time t, $\phi(t) > 1$ represents squeezing, $\phi(t) < 1$ stretching, and $\phi(t) = 1$ no modification.

Squeezing/stretching alters time stamps, e.g., it might change the length of the interval $[0, T_{\mathsf{end}}]$. We use $\tilde{\mathrm{o}}$ to denote time after squeezing/stretching. For example, the resulting interval is $[0, \widetilde{T}_{\mathsf{end}}]$. To translate between the time before and after squeezing/stretching, we use the *altered time* function AT (and its inverse AT^{-1}), e.g., $\mathsf{AT}(T_{\mathsf{end}}) = \widetilde{T}_{\mathsf{end}}$. Formally,

Definition 6 (Adversarial Chain Profile Cheating)**.** Consider a resource profile $\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0,T_{\mathsf{end}}]}$ and some function $\phi(t)\colon [0, T_{\mathsf{end}}] \to \mathbb{R}_{>0}$. Define $\mathsf{AT}(t) \coloneqq \int_0^t \frac{1}{\phi(u)}\, du$ and its inverse $\mathsf{AT}^{-1}(\cdot)$.[8]

Let $\widetilde{T}_{\mathsf{end}} \coloneqq \mathsf{AT}(T_{\mathsf{end}})$. An adversarial chain profile is any chain profile

$$\mathcal{IC}^{\mathcal{A}} = (\widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}), \widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}), \widetilde{\boldsymbol{W}}_i^{\mathcal{A}}(\widetilde{t}))_{[0,\widetilde{T}_{\mathsf{end}}]}$$

where $\widetilde{S}_i^{\mathcal{A}}(t)$, $\widetilde{V}_i^{\mathcal{A}}(t)$ and $\widetilde{W}_i^{\mathcal{A}}(t)$ are Riemann integrable over t, and satisfy

$$\begin{aligned} 0 < \widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}) &\leq \boldsymbol{S}^{\mathcal{A}}(t) \\ 0 < \widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}) &\leq \phi(t) \cdot \boldsymbol{V}^{\mathcal{A}}(t) \\ 0 < \widetilde{\boldsymbol{W}}^{\mathcal{A}}(\widetilde{t}) &\leq \phi(t) \cdot \boldsymbol{W}^{\mathcal{A}}(t) \end{aligned}$$

for all $\widetilde{t} \in [0, \widetilde{T}_{\mathsf{end}}]$ with $t = \mathsf{AT}^{-1}(\widetilde{t})$.

With the definition of chain profiles out of the way, we can now define the weight of a chain. The weight is computed using the weight function Γ.

Definition 7 (Weight Function)**.** A weight function is given by

$$\Gamma\colon \mathbb{R}_{>0}^{k_1} \times \mathbb{R}_{>0}^{k_2} \times \mathbb{R}_{>0}^{k_3} \to \mathbb{R}_{>0}.$$

As defined above, Γ takes three resources as input and computes the weight of a specific point in time. In the next step, we extend Γ to compute the weight of the whole chain. We denote this function by $\overline{\Gamma}$. Overloading notation slightly, it takes a chain (or resource) profile as input and outputs the weight of the chain (or resource) profile.

Definition 8 (Weight of a Resource or Chain Profile)**.** Consider a weight function Γ and a resource $\mathcal{R} = (\boldsymbol{S}(t), \boldsymbol{V}(t), \boldsymbol{W}(t))_{[0,T]}$ or chain profile $\mathcal{IC} = (\boldsymbol{S}(t), \boldsymbol{V}(t), \boldsymbol{W}(t))_{[0,T]}$. The weight function $\overline{\Gamma}$ is defined as

$$\overline{\Gamma}(\mathcal{R}) \coloneqq \int_0^T \Gamma(\boldsymbol{S}(t), \boldsymbol{V}(t), \boldsymbol{W}(t))\, dt$$

and

$$\overline{\Gamma}(\mathcal{IC}) \coloneqq \int_0^T \Gamma(\boldsymbol{S}(t), \boldsymbol{V}(t), \boldsymbol{W}(t))\, dt.$$

[8] AT^{-1} exists because $\frac{1}{\phi(u)} > 0$ for all u, so $\int_0^t \frac{1}{\phi(u)}\, du$ is a monotonically increasing function of t with co-domain $[0, \mathsf{AT}(T_{\mathsf{end}})]$.

There are many possible choices for Γ, but not all are good choices for a blockchain. Recall the example $W_1 \cdot W_2$ from where the adversary can cheat (i.e., gain more weight) by squeezing the time. We call a function *robust* if it resists such attacks. So even if the adversary starts out with a resource profile $\mathcal{R}^{\mathcal{A}}$ which is only slightly below the resource profile of all honest parties $\mathcal{R}^{\mathcal{H}}$, then every adversarial chain profile $\mathcal{IC}^{\mathcal{A}}$ (in the sense of Def. 6) has a lower weight than the honest one $\mathcal{IC}^{\mathcal{H}} = \mathcal{R}^{\mathcal{H}}$.

Definition 9 (Robust Weight Function)**.** A weight function Γ is *robust* if for all $\mathcal{R}^{\mathcal{H}} = (\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t))_{[0,T_{\mathsf{end}}]}$ and
$\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0,T_{\mathsf{end}}]}$ such that

$$\Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t)) \leq \Gamma(\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t)) \; \forall t \in [0, T_{\mathsf{end}}]$$

and for a time interval $[T_0, T_1]$

$$\Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t)) < \Gamma(\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t)) \; \forall t \in [T_0, T_1]$$

it holds that

$$\overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}) > \overline{\Gamma}(\mathcal{IC}^{\mathcal{A}})$$

where $\mathcal{IC}^{\mathcal{H}} \coloneqq \mathcal{R}^{\mathcal{H}}$ and $\mathcal{IC}^{\mathcal{A}}$ satisfies Def. 6 for $\mathcal{R}^{\mathcal{A}}$ and any $\phi(t)$. If no $\mathcal{R}^{\mathcal{H}}$ and $\mathcal{R}^{\mathcal{A}}$ satisfying the condition above exists then we say Γ is not robust.

Note that Def. 9 rules out that a constant function, $\Gamma = c$, is robust as there exist no points in the resource profile space where the weight differs and hence we cannot choose resource profiles satisfying the conditions in Def. 9.

Our main result completely characterizes robust weight functions.

Theorem 1 (Robust Weight Functions)**.** *A weight function Γ is robust if and only if $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is monotonically increasing and homogeneous in $\boldsymbol{V}, \boldsymbol{W}$.*

Proof. The three lemmata below prove the "if" (Lem. 1) and "only if" (Lems. 2 and 3) directions separately. □

Lemma 1. *If $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is monotonically increasing and $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is homogeneous in $\boldsymbol{V}, \boldsymbol{W}$, then $\overline{\Gamma}(\mathcal{R}^{\mathcal{A}}) \geq \overline{\Gamma} < \mathcal{IC}^{\mathcal{A}})$ where $\mathcal{IC}^{\mathcal{A}}$ satisfies Def. 6 for $\mathcal{R}^{\mathcal{A}}$ and any $\phi(t)$.*

As a consequence, Γ is robust if $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is monotonically increasing and $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is homogeneous in $\boldsymbol{V}, \boldsymbol{W}$.

Proof. For the first part of the lemma, consider the adversarial chain profile $\mathcal{IC}^{\mathcal{A}}$ from Def. 6. For any $\widetilde{t} \in [0, \widetilde{T}_{\mathsf{end}}]$, $\mathcal{A}$ could create a chain profile such that

$$\begin{aligned}
0 < \widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}) &\leq \boldsymbol{S}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t})), \\
0 < \widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}) &\leq \phi(\mathsf{AT}^{-1}(\widetilde{t})) \cdot \boldsymbol{V}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t})), \\
0 < \widetilde{\boldsymbol{W}}^{\mathcal{A}}(\widetilde{t}) &\leq \phi(\mathsf{AT}^{-1}(\widetilde{t})) \cdot \boldsymbol{W}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t})).
\end{aligned}$$

Since Γ is monotonic,

$$\begin{aligned}
&\Gamma(\widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}), \widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}), \widetilde{\boldsymbol{W}}^{\mathcal{A}}(\widetilde{t})) \leq \\
&\quad \Gamma(\boldsymbol{S}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t})), \phi(\mathsf{AT}^{-1}(\widetilde{t})) \cdot \boldsymbol{V}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t})), \phi(\mathsf{AT}^{-1}(\widetilde{t})) \cdot \boldsymbol{W}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t})))
\end{aligned}$$

holds for all $\widetilde{t} \in [0, \widetilde{T}_{\mathsf{end}}]$. Since Γ is also homogeneous in $\boldsymbol{V}, \boldsymbol{W}$,

$$\begin{aligned}&\Gamma(\widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}), \widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}), \widetilde{\boldsymbol{W}}^{\mathcal{A}}(\widetilde{t})) \leq \\ &\quad \phi(\mathsf{AT}^{-1}(\widetilde{t})) \cdot \Gamma(\boldsymbol{S}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t})), \boldsymbol{V}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t})), \boldsymbol{W}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t}))),\end{aligned}$$

so we can conclude that

$$\begin{aligned}\overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) &= \int_0^{\widetilde{T}_{\mathsf{end}}} \Gamma(\widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}), \widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}), \widetilde{\boldsymbol{W}}^{\mathcal{A}}(\widetilde{t}))\, d\widetilde{t} \\ &\leq \int_0^{\widetilde{T}_{\mathsf{end}}} \phi(\mathsf{AT}^{-1}(\widetilde{t})) \cdot \Gamma(\boldsymbol{S}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t})), \boldsymbol{V}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t})), \boldsymbol{W}^{\mathcal{A}}(\mathsf{AT}^{-1}(\widetilde{t})))\, d\widetilde{t}.\end{aligned}$$

Now, we integrate by substituting[9] $t = \mathsf{AT}^{-1}(\widetilde{t})$. Here, note that $\frac{d}{d\widetilde{t}}\mathsf{AT}^{-1}(\widetilde{t}) = \phi(\mathsf{AT}^{-1}(\widetilde{t}))$ by the inverse function rule.[10] This leads to

$$\begin{aligned}\overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) &\leq \int_{\mathsf{AT}^{-1}(0)}^{\mathsf{AT}^{-1}(\widetilde{T}_{\mathsf{end}})} \phi(T) \cdot \Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t)) \cdot \frac{1}{\phi(t)}\, dt \\ &= \int_0^{T_{\mathsf{end}}} \phi(t) \cdot \Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t)) \cdot \frac{1}{\phi(t)}\, dt \\ &= \int_0^{T_{\mathsf{end}}} \Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))\, dt \\ &= \overline{\Gamma}(\mathcal{R}^{\mathcal{A}}).\end{aligned}$$

This proves the first part of the lemma.

For the second part, note that the preconditions on resources in Def. 9 imply that

$$\overline{\Gamma}(\mathcal{R}^{\mathcal{H}}) > \overline{\Gamma}(\mathcal{R}^{\mathcal{A}}).$$

By the first part of this lemma and since $\mathcal{R}^{\mathcal{H}} = \mathcal{IC}^{\mathcal{H}}$ in Def. 9, the second part follows. □

Lemma 2. *Γ is not robust if $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is not monotonically increasing.*

Proof. Suppose Γ is not monotonically increasing, i.e., there exist $(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$ and $(\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}')$ such that $(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) < (\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}')$ but $\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) > \Gamma(\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}')$. In this case, the adversary can simply put less resources in the adversarial chain than it actually has to get a chain profile of higher weight.

Formally, for some time $T > 0$, consider the resource profiles

$$\begin{aligned}&\boldsymbol{S}^{\mathcal{H}}(t) = \boldsymbol{s}, && \boldsymbol{V}^{\mathcal{H}}(t) = \boldsymbol{v}, && \boldsymbol{W}^{\mathcal{H}}(t) = \boldsymbol{w} && \text{for } t \in [0, T_{\mathsf{end}}] \\ &\boldsymbol{S}^{\mathcal{A}}(t) = \boldsymbol{s}', && \boldsymbol{V}^{\mathcal{A}}(t) = \boldsymbol{v}', && \boldsymbol{W}^{\mathcal{A}}(t) = \boldsymbol{w}' && \text{for } t \in [0, T_{\mathsf{end}}].\end{aligned}$$

Clearly, the weight of adversarial resources is strictly less than honest resources at every point of time. Now for adversarial chain (Def. 6) $\mathcal{A}$ chooses $\phi(t) = 1$ for $t \in [0, T]$. Thus, $\mathsf{AT}(t) = \mathsf{AT}^{-1}(t) = t$ and $\widetilde{T}_{\mathsf{end}} = T_{\mathsf{end}}$. Then $\mathcal{A}$ choose

$$\begin{aligned}\widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}) &= \boldsymbol{s} \leq \phi(T) \cdot \boldsymbol{S}^{\mathcal{A}}(T) = \boldsymbol{s}' \\ \widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}) &= \boldsymbol{v} \leq \phi(T) \cdot \boldsymbol{V}^{\mathcal{A}}(T) = \boldsymbol{v}' \\ \widetilde{\boldsymbol{W}}^{\mathcal{A}}(\widetilde{t}) &= \boldsymbol{w} \leq \phi(T) \cdot \boldsymbol{W}^{\mathcal{A}}(T) = \boldsymbol{w}'\end{aligned}$$

[9] $\int_a^b f(g(x))g'(x)\, dx = \int_{g(a)}^{g(b)} f(u)\, du$

[10] $\frac{d}{dx} f^{-1}(a) = \left(f^{-1}\right)'(a) = \frac{1}{f'(f^{-1}(a))}$

for all $\widetilde{t} \in [0, \widetilde{T}_{\mathsf{end}}]$, where $T = \mathsf{AT}^{-1}(\widetilde{t})$.

Thus,

$$\begin{aligned}\overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) &= \int_0^{\widetilde{T}_{\mathsf{end}}} \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) \\ &= \int_0^{T_{\mathsf{end}}} \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) = \overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}).\end{aligned}$$

Therefore, Γ is not robust. □

Lemma 3. *Γ is not robust if $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is not homogeneous in $\boldsymbol{V}, \boldsymbol{W}$.*

Proof. Due to Def. 9, if Γ is constant then it is not robust as the preconditions on the resource profiles can not be met. In that case, we are done. From hereon we assume Γ is not a constant function.

Due to Lem. 2 we can assume that $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is monotonically increasing in $\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W}$. Suppose $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is not homogeneous in $\boldsymbol{V}, \boldsymbol{W}$, i.e., there exists $\alpha > 0$ and $(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) \in \mathbb{R}_{>0}^{k_1+k_2+k_3}$ such that $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \neq \alpha\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$. Now we have two case:

- **Case 1:** $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha \cdot \boldsymbol{w}) > \alpha\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$.

  Case 2: $\Gamma(\boldsymbol{s}, \alpha \cdot \boldsymbol{v}, \alpha \cdot \boldsymbol{w}) < \alpha\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$

  This implies $\Gamma(\boldsymbol{s}, \frac{1}{\alpha} \cdot \boldsymbol{v}', \frac{1}{\alpha}\boldsymbol{w}') > \frac{1}{\alpha} \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}', \boldsymbol{w}')$ where $\boldsymbol{v}' = \alpha\boldsymbol{v}, \boldsymbol{w}' = \alpha \cdot \boldsymbol{w}$. Since $\frac{1}{\alpha} > 0$, this case reduces to Case 1.

For **Case 1**, $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) > \alpha\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$ is equivalent to

$$\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) = \alpha\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \beta \tag{4}$$

for some $\beta \in \mathbb{R}_{>0}$. Note that $\alpha = 1$ implies $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha \cdot \boldsymbol{w}) = \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) = \alpha\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \beta = \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \beta$. Which in turn implies $\beta = 0$, a contradiction. Thus, $\alpha \neq 1$.

Case 1 can be further divided in sub-cases:

- **Case 1a:** $\alpha > 1$ and $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \leq \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$.
- **Case 1b:** $\alpha > 1$ and $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) > \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$
- **Case 1c:** $\alpha < 1$ and $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) < \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$.
- **Case 1d:** $\alpha < 1$ and $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \geq \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$.

Let's prove each case individually:

Case 1a: $\alpha > 1$ and $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \leq \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$. Since $(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) < (\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha, \boldsymbol{w})$ and due to monotonicity of Γ (Lem. 2), we also have that $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \geq \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$. Thus, $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) = \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$. Using Eq. (4), we get

$$\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) = \alpha\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \beta = \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}).$$

This implies, $(1 - \alpha)\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) = \beta$. Since $\alpha > 1$, the left-hand side is negative while right-hand side is positive. Hence, this case is impossible.

Case 1b: $\alpha > 1$ and $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) > \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$. In this case "squeezing" time gives more weight than the original resources profile. $\mathcal{A}$ will "squeeze" $(\boldsymbol{v}, \boldsymbol{w})$ by factor α to reach $(\alpha\boldsymbol{v}, \alpha\boldsymbol{w})$ and use this to get a higher weight than the honest chain profile. Formally, for $T_{\mathsf{end}} = T_0 + T_1$ where $T_1 = 1$ and $T_0 \geq \frac{\alpha-1}{\beta} \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})$ consider resource profiles $\mathcal{R}^{\mathcal{H}}$ and $\mathcal{R}^{\mathcal{A}}$ such that:

$$\begin{aligned}
&\boldsymbol{S}^{\mathcal{H}}(t) = \boldsymbol{s}, \quad && \boldsymbol{V}^{\mathcal{H}}(t) = \boldsymbol{v}, \quad && \boldsymbol{W}^{\mathcal{H}}(t) = \boldsymbol{w} && \text{for } t \in [0, T_0) \\
&\boldsymbol{S}^{\mathcal{H}}(t) = \boldsymbol{s}, && \boldsymbol{V}^{\mathcal{H}}(t) = \alpha\boldsymbol{v}, && \boldsymbol{W}^{\mathcal{H}}(t) = \alpha\boldsymbol{w} && \text{for } t \in [T_0, T_{\mathsf{end}}] \\
&\boldsymbol{S}^{\mathcal{A}}(t) = \boldsymbol{s}, && \boldsymbol{V}^{\mathcal{A}}(t) = \boldsymbol{v}, && \boldsymbol{W}^{\mathcal{A}}(t) = \boldsymbol{w} && \text{for } t \in [0, T_{\mathsf{end}}]
\end{aligned}$$

Since $\Gamma(\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t)) \geq \Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))$ for all $t \in [0, T_{\mathsf{end}}]$ and $\Gamma(\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t)) > \Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))$ for all $t \in [T_0, T_{\mathsf{end}}]$, preconditions on resource profiles of Def. 9 are satisfied.

The weight of the honest chain profile is

$$\begin{aligned}
\overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}) &= T_0 \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1 \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \\
&= T_0 \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1 \cdot \alpha \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1 \cdot \beta \qquad \text{(by } \textit{Eq.} \text{ (4))}
\end{aligned}$$

$\mathcal{A}$ chooses $\phi(t) = \alpha$ for all $t \in [0, T_{\mathsf{end}}]$. This gives $\mathsf{AT}(T) = \frac{T}{\alpha}$, $\mathsf{AT}^{-1}(\widetilde{t}) = \alpha\widetilde{t}$ and $\widetilde{T}_{\mathsf{end}} = \frac{T_{\mathsf{end}}}{\alpha}$. Setting $\phi(t) = \alpha$ is "squeezing" as $\alpha > 1$. $\mathcal{A}$ chooses

$$\begin{aligned}
\widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}) &= \boldsymbol{S}^{\mathcal{A}}(T) = \boldsymbol{s} \\
\widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}) &= \phi(T) \cdot \boldsymbol{V}^{\mathcal{A}}(T) = \alpha\boldsymbol{v} \\
\widetilde{\boldsymbol{W}}^{\mathcal{A}}(\widetilde{t}) &= \phi(T) \cdot \boldsymbol{W}^{\mathcal{A}}(T) = \alpha\boldsymbol{w}
\end{aligned}$$

for all $\widetilde{t} \in [0, \widetilde{T}_{\mathsf{end}}]$, where $T = \mathsf{AT}^{-1}(\widetilde{t}) = \alpha\widetilde{t}$.

Thus, the weight of the adversarial chain profile is

$$\begin{aligned}
\overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) &= \int_0^{\widetilde{T}_{\mathsf{end}}} \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})\, dt = \widetilde{T}_{\mathsf{end}} \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \\
&= \frac{T_{\mathsf{end}}}{\alpha} \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) = \frac{T_0 + T_1}{\alpha} \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \\
&= \frac{T_0 + T_1}{\alpha} \cdot (\alpha\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \beta) \qquad \text{by } \textit{Eq.} \text{ (4)} \\
&\text{Since } T_0 \geq \frac{\alpha - 1}{\beta} \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \text{ and } T_1 = 1, \\
&\text{by simplifying, we get} \\
&\geq T_0\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1(\alpha\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \beta) \\
&= \overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}).
\end{aligned}$$

This implies $\overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) \geq \overline{\Gamma}(\mathcal{IC}^{\mathcal{H}})$ and hence Γ is not robust.

Case 1b and 1c: These cases are similar in style to the previous two. We defer them to App. B.1.

This completes the proof. □

5 Discrete Blockchain Model

Blockchains do not reflect the ideal resources but consist of discrete blocks. Honest users create blocks according to some prescribed rule, e.g., in fixed time intervals, but the adversary may not adhere to this rule. A block reflects the *total* amount of resources that were expended to create it.

Space and VDF speed are *timed* resources, e.g., hashes per second. For these, the block reflects (at most) the *total* amount of effort that went into it. For example, 2^{30} hashes were performed to create a block. In contrast, space does not account for time as it's usually measured in bits—it is *time-invariant*.

Thus, the block may only reflect the space available *at some point* during the block.

Like in the ideal model, we will describe which Γ leads to a secure discrete blockchain. Compared to the ideal model, we need additional, yet realistic assumptions. First, the security result is quantitative and depends on the fluctuation of resources within a block. If the fluctuation is *smooth*, then the blockchain has good security in a quantitative sense. Second, space must be sub-homogeneous to disincentivize replotting attacks as already explained in § 1.9.

5.1 Definitions

The following definitions define a blockchain $\mathcal{BC}$ as the discretization of an ideal chain profile $\mathcal{IC}$. Keep in mind, however, that the ideal chain profile $\mathcal{IC}$ itself arises a resource profile $\mathcal{R}$ as described in § 4. Looking ahead to Thm. 2, we will ultimately define the security of $\mathcal{BC}$ with respect to $\mathcal{R}$.

Let us first define a *block*. In short, it accurately reflects the timed resources $\boldsymbol{V}$ and $\boldsymbol{W}$ used to create it. $\boldsymbol{S}$ is different; The space reflected by a block $\boldsymbol{S}(b_i)$ needs to lie between the minimum and maximum space available during the b_i's timespan.

Definition 10 (Blocks)**.** Let $\mathcal{IC} = (\boldsymbol{S}(t), \boldsymbol{V}(t), \boldsymbol{W}(t))_{[0,T]}$ be a chain profile. A block b_i is defined by a timespan (t_i, t_i') with $0 \leq t_i < t_i' \leq T$. The resources reflected by the block are denoted by $\mathbf{S}_{\blacksquare}(b_i)$, $\mathbf{V}_{\blacksquare}(b_i)$, and $\mathbf{W}_{\blacksquare}(b_i)$.

Timed resources $\mathbf{V}_{\blacksquare}$ and $\mathbf{W}_{\blacksquare}$ are reflected by

$$\mathbf{V}_{\blacksquare}(b_i) = \int_{t_i}^{t_i'} \boldsymbol{V}(t)\, dt \quad \text{and} \quad \mathbf{W}_{\blacksquare}(b_i) = \int_{t_i}^{t_i'} \boldsymbol{W}(t)\, dt.$$

The constraint on $\mathbf{S}_{\blacksquare}$ is that

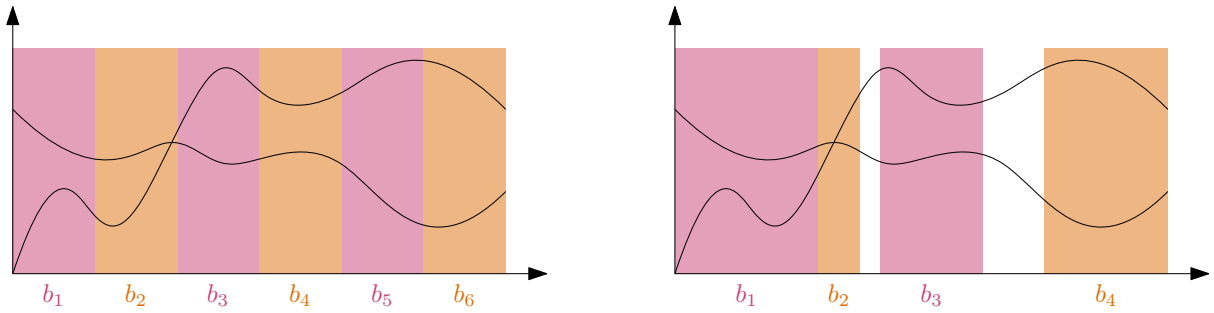
$$\min_{t_i \leq t \leq t_i'} \boldsymbol{S}(t) \leq \mathbf{S}_{\blacksquare}(b_i) \leq \max_{t_i \leq t \leq t_i'} \boldsymbol{S}(t). \tag{5}$$

In the sequel, we will often make use of minima and maxima of resources. For a block b_i with timespan (t_i, t_i') we denote the minimum and maximum space by

$$\mathbf{S}_{\blacksquare,\min}(b_i) = \min_{t_i \leq t \leq t_i'} \boldsymbol{S}(t) \quad \text{and} \quad \mathbf{S}_{\blacksquare,\max}(b_i) = \max_{t_i \leq t \leq t_i'} \boldsymbol{S}(t)$$

where $\min, \max$ is applied element-wise over whole vector. This is analogously defined for $\mathbf{V}_{\blacksquare}$ and $\mathbf{W}_{\blacksquare}$.

Now, a blockchain is a sequence of non-overlapping blocks. Its weight $\overline{\Gamma}_{\blacksquare}$ is the sum of the blocks' weights.



(a) Honest parties. (b) Adversary (Example).

Figure 5: Discretization of parties. Here, honest parties discretize in fixed time intervals, while the adversary may construct blocks in any non-overlapping fashion.

Definition 11 (Discrete Blockchain)**.** A discrete blockchain is a sequence of blocks $\mathcal{BC} = (b_0, \dots b_B)$. The weight of a blockchain is

$$\overline{\Gamma}_{\blacksquare}(\mathcal{BC}) = \sum_{b_i \in \mathcal{BC}} \Gamma(\mathbf{S}_{\blacksquare}(b_i), \mathbf{V}_{\blacksquare}(b_i), \mathbf{W}_{\blacksquare}(b_i)) \tag{6}$$

Blocks are discrete chunks of an ideal chain profile. Recall that this chain profile arises from a resource profile; for honest parties, they are identical. To discretize their chain profile, honest parties follow some prescribed rules to create blocks. The resulting blocks are non-overlapping and cover the whole timespan without gaps. A rule is, e.g., creating blocks in fixed time intervals (cf. Fig. 5a).

Definition 12 (Honest Discretization)**.** Consider the honest parties' resource profile $\mathcal{R}^{\mathcal{H}} = (\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t))_{[0,T]}$ and their ideal chain profile $\mathcal{IC}^{\mathcal{H}} := \mathcal{R}^{\mathcal{H}}$. The corresponding honest blockchain is $\mathcal{BC}^{\mathcal{H}} = (b_0^{\mathcal{H}}, \dots, b_B^{\mathcal{H}})$ where the timespan of block $b_i^{\mathcal{H}}$ is (t_i, t_i'). It holds that $t_0 = 0$, $t_B' = T$, and $t_i' = t_{i+1}$ for all $i \in [B-1]$.

The adversary also starts from a resource profile, but they may cheat when deriving the ideal chain profile from the resources. Recall that the adversary may cheat by squeezing/stretching time, or pretending to have fewer resources (as previously defined in Def. 6). In terms of discretization, the adversary may not necessarily follow the prescribed rule. It may create blocks covering varying timespans, or it might leave gaps between blocks. The only condition is that blocks don't overlap. An example is shown in Fig. 5b.

Definition 13 (Adversarial Discretization)**.** Consider the adversary's resource profile $\mathcal{IC}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0,T]}$ and any ideal chain profile $\mathcal{IC}^{\mathcal{A}}$ satisfying Def. 6 for some ϕ. The corresponding adversarial blockchain is $\mathcal{BC}^{\mathcal{A}} = (b_0^{\mathcal{A}}, \dots, b_B^{\mathcal{A}})$ where the timespan of block $b_i^{\mathcal{A}}$ is (t_i, t_i'). It holds that $t_0 \geq 0$, $t_B' \leq T$, and $t_i' \leq t_{i+1}$ for all $i \in [B-1]$.

Looking ahead, the security of the discrete blockchain depends on the maximum fluctuation of resources. We quantify this fluctuation by the ε-*Smoothness* of resources. Essentially, ε bounds the absolute change of resources within a block.

Definition 14 (ε-Smoothness)**.** Let $\varepsilon > 0$. A blockchain $\mathcal{BC}$ satisfies ε-dynamic availability if, for all blocks $0 \leq i \leq B$, it holds that

$$\begin{aligned}\mathbf{S}_{\blacksquare,\max}(b_i) &\leq (1+\varepsilon)\mathbf{S}_{\blacksquare,\min}(b_i)\\ \mathbf{V}_{\blacksquare,\max}(b_i) &\leq (1+\varepsilon)\mathbf{V}_{\blacksquare,\min}(b_i)\\ \mathbf{W}_{\blacksquare,\max}(b_i) &\leq (1+\varepsilon)\mathbf{W}_{\blacksquare,\min}(b_i).\end{aligned}$$

In practice, ε depends on how quickly users can acquire hardware (e.g., hard disks or mining ASICs). It follows that ε only takes meaningful values if blocks do not span too much time.

One way to limit the time span is Bitcoin's difficulty adjustment mechanism. The block difficulty is an upper bound (and lower bound) on the work reflected by a block which is re-calibrated every two weeks. Since there is no incentive to expend more work as soon as this upper bound is reached, the time span is effectively limited.

Bitcoin's difficulty adjustment can be generalized to the multi-resource setting by simply imposing an upper bound on the total weight of a block. As in Bitcoin, this upper bound needs continual re-adjustment since the total resources dedicated to the blockchain increase over time. For simplicity, we will not describe this formally; we simply assume ε-smoothness holds for some ε.

5.2 Security Statement and Proof

Intuitively, a discrete blockchain is secure if the following holds. If the adversary starts out with fewer resources than the honest parties, then the weight of the adversarial blockchain is lower than that of the honest one. In other words, if $\overline{\Gamma}(\mathcal{R}^{\mathcal{A}}) < \overline{\Gamma}(\mathcal{R}^{\mathcal{H}})$, then $\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{A}}) < \overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{H}})$.

Our actual result is a bit weaker because we require a gap between honest and adversarial resources, namely, $(1+\varepsilon)^4 \cdot \overline{\Gamma}(\mathcal{R}^{\mathcal{A}}) < \overline{\Gamma}(\mathcal{R}^{\mathcal{H}})$. So the gap $(1+\varepsilon)^4$ is smaller the smoother resources are. To demonstrate where this gap comes from, let us sketch the reason for a $(1+\varepsilon)^2$ part of the gap. Consider the space $\boldsymbol{S}(b_i)$ of a block. Recall that space is a time-invariant resource, so a block represents the space available *at some point* within the block (and not the cumulative space over the block's timespan). This is why Def. 10 demands that $\mathbf{S}_{\blacksquare,\min}(b_i) \leq \mathbf{S}_{\blacksquare}(b_i) \leq \mathbf{S}_{\blacksquare,\max}(b_i)$. Now, we must be pessimistic and assume that honest parties only get $\mathbf{S}^{\mathcal{H}}_{\blacksquare,\min}(b_i)$ while the adversary is lucky and gets $\mathbf{S}^{\mathcal{A}}_{\blacksquare,\max}(b_i)$. Since space is ε-smooth, each of these introduces a gap of $(1+\varepsilon)$ These are multiplicative, so in total $(1+\varepsilon)^2$.

Theorem 2. *Let Γ be a robust weight function (Def. 9) that is sub-homogeneous in $\boldsymbol{S}$ (Def. 3). Consider any honest and adversarial resource profiles $\mathcal{R}^{\mathcal{H}}$ and $\mathcal{R}^{\mathcal{A}}$. Let $\mathcal{BC}^{\mathcal{H}}$ and $\mathcal{BC}^{\mathcal{A}}$ be blockchains arising from these resource profiles according to Defs. 12 and 13.*

If $\mathcal{BC}^{\mathcal{H}}$ and $\mathcal{BC}^{\mathcal{A}}$ are ε-smooth (Def. 14) and

$$\overline{\Gamma}(\mathcal{R}^{\mathcal{H}}) > (1+\varepsilon)^4 \cdot \overline{\Gamma}(\mathcal{R}^{\mathcal{A}}), \tag{7}$$

then

$$\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{H}}) > \overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{A}}).$$

Before we prove the theorem, we state a corollary that is easier to grasp intuitively. Assume that S is not only sub-homogeneous, but also homogeneous.[11] This covers natural choices such as Chia's $S \cdot V$. Under this assumption, Thm. 2 may be viewed in the following way.

Corollary 1. *Under the assumptions of Thm. 2 with the additional constraint that* Γ *is homogeneous in* $\boldsymbol{S}$*, if*

$$\begin{aligned}(1+\varepsilon)^2 \cdot \boldsymbol{S}^{\mathcal{A}}(t) &< \boldsymbol{S}^{\mathcal{H}}(t)\\ (1+\varepsilon)^2 \cdot \boldsymbol{V}^{\mathcal{A}}(t) &< \boldsymbol{V}^{\mathcal{H}}(t)\\ (1+\varepsilon)^2 \cdot \boldsymbol{W}^{\mathcal{A}}(t) &< \boldsymbol{W}^{\mathcal{H}}(t)\end{aligned}$$

hold for all $0 \leq t \leq T$*, then*

$$\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{H}}) > \overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{A}}).$$

Essentially, Corollary 1 tells us that if the adversary has sufficiently less of every resource than all honest parties combined, the blockchain is secure. The smaller ε is the stronger the statement. In practice, ε behaves in a way that corresponds to an intuitive understanding of blockchain security. First, ε decreases as the total amount of resources committed to the blockchain increases. Second, the shorter the time span of blocks is, the smaller ε. Both of these properties are satisfied by the most popular longest-chain blockchains used in practice.

Proof of Thm. 2. First, since Γ is robust, it must be monotonic and homogeneous by Thm. 1. Second, Eq. (7) implies that

$$\overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}) > (1+\varepsilon)^4 \cdot \overline{\Gamma}(\mathcal{IC}^{\mathcal{A}})$$

because $\overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}) = \overline{\Gamma}(\mathcal{R}^{\mathcal{H}})$ and $\overline{\Gamma}(\mathcal{R}^{\mathcal{A}}) \geq \overline{\Gamma}(\mathcal{IC}^{\mathcal{A}})$ due to Lem. 1.

Using this, we will now prove that the sequence of inequalities

$$\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{H}}) \geq \frac{1}{(1+\varepsilon)^2}\,\overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}) > (1+\varepsilon)^2 \cdot \overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) \geq \overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{A}})$$

holds, which in turn implies the theorem. We will prove the left and right inequality separately.

Case $\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{H}}) \geq \frac{1}{(1+\varepsilon)^2}\,\overline{\Gamma}(\mathcal{IC}^{\mathcal{H}})$: By definition of every block b_i with timespan (t_i, t_i'), it follows that

$$\begin{aligned}\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{H}}) &= \sum_{b_i \in \mathcal{BC}^{\mathcal{H}}} \Gamma\left(\mathbf{S}_{\blacksquare}^{\mathcal{H}}(b_i), \mathbf{V}_{\blacksquare}^{\mathcal{H}}(b_i), \mathbf{W}_{\blacksquare}^{\mathcal{H}}(b_i)\right)\\ &= \sum_{b_i \in \mathcal{BC}^{\mathcal{H}}} \Gamma\left(\mathbf{S}_{\blacksquare}^{\mathcal{H}}(b_i), \int_{t_i}^{t_i'} \boldsymbol{V}^{\mathcal{H}}(t)\,dt, \int_{t_i}^{t_i'} \boldsymbol{W}^{\mathcal{H}}(t)\,dt\right)\\ &\geq \sum_{b_i \in \mathcal{BC}^{\mathcal{H}}} \Gamma\left(\mathbf{S}_{\blacksquare} \overset{\mathcal{H}}{\min}(b_i), \int_{t_i}^{t_i'} \boldsymbol{V}^{\mathcal{H}}(t)\,dt, \int_{t_i}^{t_i'} \boldsymbol{W}^{\mathcal{H}}(t)\,dt\right).\end{aligned}$$

[11] Note that every homogeneous function is sub-homogeneous, but not vice-versa.

The third line follows by the monotonicity of Γ and the fact that $\mathbf{S}_{\blacksquare}^{\mathcal{H}}(b_i) \geq \mathbf{S}_{\blacksquare,\min}^{\mathcal{H}}(b_i)$ necessarily.

Let $\overline{\mathbf{V}_{\blacksquare}}^{\mathcal{H}}(b_i) = \dfrac{1}{t_i' - t_i} \cdot \displaystyle\int_{t_i}^{t_i'} \boldsymbol{V}^{\mathcal{H}}(t)\, dt$ denote the average VDF speed within a block. Clearly, $\mathbf{V}_{\blacksquare}\min^{\mathcal{H}}(b_i) \leq \overline{\mathbf{V}_{\blacksquare}}^{\mathcal{H}}(b_i) \leq \mathbf{V}_{\blacksquare,\max}^{\mathcal{H}}(b_i)$. Define $\overline{\mathbf{W}_{\blacksquare}}^{\mathcal{H}}(b_i)$ analogously. Using these insights, we continue with

$$\begin{aligned}
\Gamma(\mathcal{BC}^{\mathcal{H}}) &\geq \sum_{b_i \in \mathcal{BC}^{\mathcal{H}}} \Gamma\left(\mathbf{S}_{\blacksquare,\min}^{\mathcal{H}}(b_i), \int_{t_i}^{t_i'} \boldsymbol{V}^{\mathcal{H}}(t)\, dt, \int_{t_i}^{t_i'} \boldsymbol{W}^{\mathcal{H}}(t)\, dt\right) \\
&= \sum_{b_i \in \mathcal{BC}^{\mathcal{H}}} (t_i' - t_i) \cdot \Gamma\left(\mathbf{S}_{\blacksquare,\min}^{\mathcal{H}}(b_i), \overline{\mathbf{V}_{\blacksquare}}^{\mathcal{H}}(b_i), \overline{\mathbf{W}_{\blacksquare}}^{\mathcal{H}}(b_i)\right) \\
&\geq \sum_{b_i \in \mathcal{BC}^{\mathcal{H}}} (t_i' - t_i) \cdot \Gamma\left(\mathbf{S}_{\blacksquare,\min}^{\mathcal{H}}(b_i), \mathbf{V}_{\blacksquare,\min}^{\mathcal{H}}(b_i), \mathbf{W}_{\blacksquare,\min}^{\mathcal{H}}(b_i)\right)
\end{aligned}$$

where the second line follows as Γ is homogeneous in $\boldsymbol{V}, \boldsymbol{W}$ and the last line follows from monotonicity.

Now we invoke Def. 14 to switch min to max, that is,

$$\begin{aligned}
\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{H}}) &\geq \sum_{b_i \in \mathcal{BC}^{\mathcal{H}}} (t_i' - t_i) \cdot \Gamma(\mathbf{S}_{\blacksquare,\min}^{\mathcal{H}}(b_i), \mathbf{V}_{\blacksquare,\min}^{\mathcal{H}}(b_i), \mathbf{W}_{\blacksquare,\min}^{\mathcal{H}}(b_i)) \\
&\geq \frac{1}{(1+\varepsilon)} \sum_{b_i \in \mathcal{BC}^{\mathcal{H}}} (t_i' - t_i) \cdot \Gamma(\mathbf{S}_{\blacksquare,\max}^{\mathcal{H}}(b_i), \mathbf{V}_{\blacksquare,\min}^{\mathcal{H}}(b_i), \mathbf{W}_{\blacksquare,\min}^{\mathcal{H}}(b_i)) \\
&= \frac{1}{(1+\varepsilon)^2} \sum_{b_i \in \mathcal{BC}^{\mathcal{H}}} (t_i' - t_i) \cdot \Gamma(\mathbf{S}_{\blacksquare,\max}^{\mathcal{H}}(b_i), \mathbf{V}_{\blacksquare,\max}^{\mathcal{H}}(b_i), \mathbf{W}_{\blacksquare,\max}^{\mathcal{H}}(b_i))
\end{aligned}$$

where the second line follows from Γ being sub-homogeneous in $\boldsymbol{S}$ and the third from the homogeneity of Γ in $(\boldsymbol{V}, \boldsymbol{W})$.

This implies the desired inequality because

$$\begin{aligned}
\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{H}}) &\geq \frac{1}{(1+\varepsilon)^2} \sum_{b_i \in \mathcal{BC}^{\mathcal{H}}} (t_i' - t_i) \cdot \Gamma(\mathbf{S}_{\blacksquare,\max}^{\mathcal{H}}(b_i), \mathbf{V}_{\blacksquare,\max}^{\mathcal{H}}(b_i), \mathbf{W}_{\blacksquare,\max}^{\mathcal{H}}(b_i)) \\
&\geq \frac{1}{(1+\varepsilon)^2} \sum_{b_i \in \mathcal{BC}^{\mathcal{H}}} \int_{t_i}^{t_i'} \Gamma(\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t))\, dt \\
&= \frac{1}{(1+\varepsilon)^2} \int_0^T \Gamma(\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t))\, dt \\
&= \frac{1}{(1+\varepsilon)^2}\, \overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}).
\end{aligned}$$

Note that the third line follows because the blocks of honest parties span the whole timespan without gaps by definition.

Case $\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{A}}) \leq (1+\varepsilon)^2\, \overline{\Gamma}(\mathcal{IC}^{\mathcal{A}})$: The proof is symmetrical to the previous case. Essentially, "$\geq$" is swapped with "$\leq$" and "min" with "max". For completeness, this case is stated in App. B.2.

□

Acknowledgements. This research was funded in whole or in part by the Austrian Science Fund (FWF) 10.55776/F85.

References

[BBBF18] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 757–788. Springer, Heidelberg, August 2018.

[BDK+22] Vivek Kumar Bagaria, Amir Dembo, Sreeram Kannan, Sewoong Oh, David Tse, Pramod Viswanath, Xuechao Wang, and Ofer Zeitouni. Proof-of-stake longest chain protocols: Security vs predictability. In Jorge M. Soares, Dawn Song, and Marko Vukolic, editors, *Proceedings of the 2022 ACM Workshop on Developments in Consensus, ConsensusDay 2022, Los Angeles, CA, USA, 7 November 2022*, pages 29–42. ACM, 2022.

[DFKP15] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. Proofs of space. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 585–605. Springer, Heidelberg, August 2015.

[ES14] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *FC 2014*, volume 8437 of *LNCS*, pages 436–454. Springer, Heidelberg, March 2014.

[FWK+22] Matthias Fitzi, Xuechao Wang, Sreeram Kannan, Aggelos Kiayias, Nikos Leonardos, Pramod Viswanath, and Gerui Wang. Minotaur: Multi-resource blockchain consensus. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 1095–1108. ACM Press, November 2022.

[PKF+18] Sunoo Park, Albert Kwon, Georg Fuchsbauer, Peter Gazi, Joël Alwen, and Krzysztof Pietrzak. SpaceMint: A cryptocurrency based on proofs of space. In Sarah Meiklejohn and Kazue Sako, editors, *FC 2018*, volume 10957 of *LNCS*, pages 480–499. Springer, Heidelberg, February / March 2018.

[TSZ23] Apostolos Tzinas, Srivatsan Sridhar, and Dionysis Zindros. On-chain timestamps are accurate. *IACR Cryptol. ePrint Arch.*, page 1648, 2023.

[TTG+23] Ertem Nusret Tas, David Tse, Fangyu Gai, Sreeram Kannan, Mohammad Ali Maddah-Ali, and Fisher Yu. Bitcoin-enhanced proof-of-stake security: Possibilities and impossibilities. In *2023 IEEE Symposium on Security and Privacy*, pages 126–145. IEEE Computer Society Press, May 2023.

A Remarks on Ideal Model

An alternate to precondition on resource profiles in Def. 9 is to have honest resources be strictly larger than adversarial every time instead of just for an interval. In that case, our main result Thm. 1 would be true in one direction (monotonically increasing and homogeneous implies robust) but in the other direction, it would not follow because not every non-homogeneous function can be attacked (*e.g.* a function $S \cdot \max(V, W)$, when each resource is below some threshold c and is constant c^2 after that). If we additionally put a natural constraint that weight function Γ is not eventually constant (*i.e.* for any point $(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ there exists $(\boldsymbol{S}', \boldsymbol{V}', \boldsymbol{W}')$ such that $(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W}) < (\boldsymbol{S}', \boldsymbol{V}', \boldsymbol{W}')$ and $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W}) < \Gamma(\boldsymbol{S}', \boldsymbol{V}', \boldsymbol{W}')$), then it would hold true that Γ is robust if and only if it is monotonically increasing and homogeneous in $\boldsymbol{V}, \boldsymbol{W}$. We do not formally prove this in this paper but it is an adaptation of our proof. Either way, the main takeaway is that monotonically increasing and homogeneous are the robust functions and they are the only ones that should be used to construct Nakamoto-style longest chain blockchains using multiple resources.

B Missing Parts of Proofs

B.1 Proof of Lemma 3

Case 1c: $\alpha < 1$ and $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) < \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$. This case is the reverse of the previous case. Here "stretching" leads to a higher weight than the original resource profile. $\mathcal{A}$ "stretches" $(\boldsymbol{v}, \boldsymbol{w})$ by a factor α into $(\alpha\boldsymbol{v}, \alpha\boldsymbol{w})$ in order to get a higher weighted chain profile than the honest chain profile.

Formally, let $T_{\mathsf{end}} = T_0 + T_1$ where $T_1 = 1$ and $T_0 \geq \frac{\alpha}{\beta}((1-\alpha)\,\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) - \beta)$, $T_0 > 0$ and consider the resource profiles $\mathcal{R}^{\mathcal{H}}$ and $\mathcal{R}^{\mathcal{A}}$:

$$
\begin{aligned}
&\boldsymbol{S}^{\mathcal{H}}(t) = \boldsymbol{s}, && \boldsymbol{V}^{\mathcal{H}}(t) = \boldsymbol{v}, && \boldsymbol{W}^{\mathcal{H}}(t) = \boldsymbol{w} && \text{for } t \in [0, T_{\mathsf{end}}] \\
&\boldsymbol{S}^{\mathcal{A}}(t) = \boldsymbol{s}, && \boldsymbol{V}^{\mathcal{A}}(t) = \boldsymbol{v}, && \boldsymbol{W}^{\mathcal{A}}(t) = \boldsymbol{w} && \text{for } t \in [0, T_0) \\
&\boldsymbol{S}^{\mathcal{A}}(t) = \boldsymbol{s}, && \boldsymbol{V}^{\mathcal{A}}(t) = \alpha\boldsymbol{v}, && \boldsymbol{W}^{\mathcal{A}}(t) = \alpha\boldsymbol{w} && \text{for } t \in [T_0, T_{\mathsf{end}}]
\end{aligned}
$$

Since $\Gamma(\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t)) \geq \Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))$ for all $t \in [0, T_{\mathsf{end}}]$ and $\Gamma(\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t)) > \Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))$ for all $t \in [T_0, T_{\mathsf{end}}]$, the preconditions on resource profiles in Def. 9 are met.

The weight of the honest chain profile is

$$\overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}) = T_{\mathsf{end}} \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) = (T_0 + T_1) \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}).$$

$\mathcal{A}$ sets $\phi(t) = \alpha$ for all $t \in [0, T_0)$ and $\phi(t) = 1$ for all $t \in [T_0, T_1]$, which gives

$$\mathsf{AT}(T) = \begin{cases} \frac{T}{\alpha} & \text{for all } t \in [0, T_0) \\ \frac{T_0}{\alpha} + (T - T_0) & \text{for all } t \in [T_0, T_1] \end{cases}$$

$$\mathsf{AT}^{-1}(\widetilde{t}) = \begin{cases} \alpha\widetilde{t} & \text{for all } \widetilde{t} \in [0, \frac{T_0}{\alpha}) \\ T_0 + (\widetilde{t} - \frac{T_0}{\alpha}) & \text{for all } \widetilde{t} \in [\frac{T_0}{\alpha}, \widetilde{T}_{\mathsf{end}}] \end{cases}$$

and $\widetilde{T}_{\mathsf{end}} = \frac{T_0}{\alpha} + T_1$. Setting $\phi(t) = \alpha$ is "stretching" as $\alpha < 1$.

Now $\mathcal{A}$ chooses

$$\begin{aligned} \widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}) &= \boldsymbol{S}^{\mathcal{A}}(T) \\ \widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}) &= \phi(T) \cdot \boldsymbol{V}^{\mathcal{A}}(T) \\ \widetilde{\boldsymbol{W}}^{\mathcal{A}}(\widetilde{t}) &= \phi(T) \cdot \boldsymbol{W}^{\mathcal{A}}(T) \end{aligned}$$

for all $\widetilde{t} \in [0, \widetilde{T}_{\mathsf{end}}]$, where $T = \mathsf{AT}^{-1}(\widetilde{t}) = \alpha\widetilde{t}$.

Thus, the weight of the adversarial chain profile is

$$\begin{aligned} \overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) &= \int_0^{\mathsf{AT}(T_0)} \Gamma(\widetilde{\boldsymbol{S}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{V}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{W}}^{\mathcal{A}}(t))\, dt \\ &\qquad + \int_{\mathsf{AT}(T_0)}^{\widetilde{T}_{\mathsf{end}}} \Gamma(\widetilde{\boldsymbol{S}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{V}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{W}}^{\mathcal{A}}(t))\, dt \\ &= \int_0^{\frac{T_0}{\alpha}} \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})\, dt \\ &\qquad + \int_{\frac{T_0}{\alpha}}^{\frac{T_0}{\alpha}+T_1} \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})\, dt \\ &= \frac{T_0}{\alpha} \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) + T_1 \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \\ &= \left(\frac{T_0}{\alpha} + T_1\right)(\alpha\, \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \beta) \qquad \text{by } \textit{Eq. (4)} \\ &\text{Since } T_0 \geq \frac{\alpha}{\beta}((1-\alpha)\, \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) - \beta) \text{ and } T_1 = 1, \\ &\text{by simplifying, we get} \\ &\geq (T_0 + T_1) \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) \\ &= \overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}). \end{aligned}$$

This implies $\overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) \geq \overline{\Gamma}(\mathcal{IC}^{\mathcal{H}})$ and thus Γ is not robust.

Case 1d: $\alpha < 1$ and $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \geq \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$. Since $(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) < (\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$, by monotonicity Lem. 2 we have that $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \leq \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$. Thus, $\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) = \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$. Intuitively, this says that stretching by factor $\frac{1}{\alpha}$ doesn't change the weight but since it increases the time as well it will give higher weight to the resulting chain profile.

To show this formally we need to find two points in resource space such that weight varies among the two points. Since Γ is not constant, there exists $(\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}') \in \mathbb{R}^3_{>0}$ such that $\Gamma(\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}') \neq \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) = \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})$.

Let $\delta \coloneqq |\,\Gamma(\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}') - \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})|$.

We have two cases:

Case A: $\Gamma(\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}') > \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$

Case B: $\Gamma(\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}') < \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})$.

We describe the violation of Def. 9 in both cases together while highlighting the differences in the steps as we go: Let $T_{\mathsf{end}} = T_0 + T_1$ where $T_1 = 1$ and $T_0 \geq \frac{\delta}{\Gamma(\boldsymbol{s},\boldsymbol{v},\boldsymbol{w})\cdot(\frac{1}{\alpha}-1)}$.

Consider the resource profiles $\mathcal{R}^{\mathcal{H}} = (\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t))$ and $\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))$ such that:

$$
\begin{aligned}
&\boldsymbol{S}^{\mathcal{H}}(t) = \boldsymbol{s}, \quad && \boldsymbol{V}^{\mathcal{H}}(t) = \boldsymbol{v}, \quad && \boldsymbol{W}^{\mathcal{H}}(t) = \boldsymbol{w} && \text{for } t \in [0, T_0) \\
&\boldsymbol{S}^{\mathcal{A}}(t) = \boldsymbol{s}, && \boldsymbol{V}^{\mathcal{A}}(t) = \boldsymbol{v}, && \boldsymbol{W}^{\mathcal{A}}(t) = \boldsymbol{w} && \text{for } t \in [0, T_0] \\
&\textbf{Case A:}: \\
&\boldsymbol{S}^{\mathcal{H}}(t) = \boldsymbol{s}', && \boldsymbol{V}^{\mathcal{H}}(t) = \boldsymbol{v}', && \boldsymbol{W}^{\mathcal{H}}(t) = \boldsymbol{w}' && \text{for } t \in [T_0, T_{\mathsf{end}}] \\
&\boldsymbol{S}^{\mathcal{A}}(t) = \boldsymbol{s}, && \boldsymbol{V}^{\mathcal{A}}(t) = \boldsymbol{v}, && \boldsymbol{W}^{\mathcal{A}}(t) = \boldsymbol{w} && \text{for } t \in [T_0, T_{\mathsf{end}}] \\
&\textbf{Case B:}: \\
&\boldsymbol{S}^{\mathcal{H}}(t) = \boldsymbol{s}, && \boldsymbol{V}^{\mathcal{H}}(t) = \boldsymbol{v}, && \boldsymbol{W}^{\mathcal{H}}(t) = \boldsymbol{w} && \text{for } t \in [T_0, T_{\mathsf{end}}] \\
&\boldsymbol{S}^{\mathcal{A}}(t) = \boldsymbol{s}', && \boldsymbol{V}^{\mathcal{A}}(t) = \boldsymbol{v}', && \boldsymbol{W}^{\mathcal{A}}(t) = \boldsymbol{w}' && \text{for } t \in [T_0, T_{\mathsf{end}}]
\end{aligned}
$$

Note that in both cases we have an interval where $\mathcal{A}$'s resources has strictly lower weight than the $\mathcal{H}$'s resources. Thus, it satisfies the precondition on resource profiles in Def. 9.

The weight of the honest chain profile is:

$$
\overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}) = \begin{cases} T_0 \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1 \cdot \Gamma(\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}') & \text{for } \textbf{Case A} \\ T_0 \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1 \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) & \text{for } \textbf{Case B} \end{cases}
$$

which, by definition of δ, is same as:

$$
\overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}) = \begin{cases} T_0 \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \delta & \text{for } \textbf{Case A} \\ T_0 \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \Gamma(\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}') + \delta & \text{for } \textbf{Case B} \end{cases}
$$

$\mathcal{A}$ chooses $\phi(t) = \alpha$ for all $t \in [0, T_0)$ and $\phi(t) = 1$ for all $t \in [T_0, T_1]$. This intuitively gives us a stretch by factor $\frac{1}{\alpha}$ (as $\alpha < 1$) for $[0, T_0]$ and the remaining time remains the same.

We get

$$
\mathsf{AT}(T) = \begin{cases} \frac{T}{\alpha} & \text{for all } t \in [0, T_0) \\ \frac{T_0}{\alpha} + (T - T_0) & \text{for all } t \in [T_0, T_1] \end{cases}
$$

$$\mathsf{AT}^{-1}(\widetilde{t}) = \begin{cases} \alpha\widetilde{t} & \text{for all } \widetilde{t} \in [0, \frac{T_0}{\alpha}) \\ T_0 + (\widetilde{t} - \frac{T_0}{\alpha}) & \text{for all } \widetilde{t} \in [\frac{T_0}{\alpha}, \widetilde{T}_{\mathsf{end}}] \end{cases}$$

and $\widetilde{T}_{\mathsf{end}} = \frac{T_0}{\alpha} + T_1$.
$\mathcal{A}$ chooses

$$\begin{aligned} \widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}) &= \boldsymbol{S}^{\mathcal{A}}(T) = \boldsymbol{s} \\ \widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}) &= \phi(T) \cdot \boldsymbol{V}^{\mathcal{A}}(T) = \alpha\boldsymbol{v} \\ \widetilde{\boldsymbol{W}}^{\mathcal{A}}(\widetilde{t}) &= \phi(T) \cdot \boldsymbol{W}^{\mathcal{A}}(T) = \alpha\boldsymbol{w} \end{aligned}$$

for all $\widetilde{t} \in [0, \mathsf{AT}(T_0)]$ and

$$\begin{aligned} \widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}) &= \phi(T) \cdot \boldsymbol{S}^{\mathcal{A}}(T) \\ \widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}) &= \phi(T) \cdot \boldsymbol{V}^{\mathcal{A}}(T) \\ \widetilde{\boldsymbol{W}}^{\mathcal{A}}(\widetilde{t}) &= \phi(T) \cdot \boldsymbol{W}^{\mathcal{A}}(T) \end{aligned}$$

for all $\widetilde{t} \in [\mathsf{AT}(T_0), \widetilde{T}_{\mathsf{end}}]$ where $T = \mathsf{AT}^{-1}(\widetilde{t}) = \alpha\widetilde{t}$.

Thus, the weight of adversarial chain profile is

$$\begin{aligned} \overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) &= \int_0^{\widetilde{T}_{\mathsf{end}}} \Gamma(\widetilde{\boldsymbol{S}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{V}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{W}}^{\mathcal{A}}(t))\, dt \\ &= \int_0^{\mathsf{AT}(T_0)} \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})\, dt \\ &+ \int_{\mathsf{AT}(T_0)}^{\widetilde{T}_{\mathsf{end}}} \Gamma(\widetilde{\boldsymbol{S}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{V}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{W}}^{\mathcal{A}}(t))\, dt \end{aligned}$$

For **Case A:**,

$$\begin{aligned} \overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) &= \int_0^{\frac{T_0}{\alpha}} \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})\, dt + \int_{\frac{T_0}{\alpha}}^{\frac{T_0}{\alpha}+T_1} \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})\, dt \\ &= \frac{T_0}{\alpha} \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1 \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) \\ &\text{Since } T_0 \geq \frac{\delta}{\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) \cdot (\frac{1}{\alpha} - 1)} \text{ and } T_1 = 1, \\ &\text{plugging in and simplifying, we get} \\ &\geq \overline{\Gamma}(\mathcal{IC}^{\mathcal{H}}) \end{aligned}$$

For **Case B:**,

$$\begin{aligned}
\overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) &= \int_0^{\frac{T_0}{\alpha}} \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w})\, dt + \int_{\frac{T_0}{\alpha}}^{\frac{T_0}{\alpha}+T_1} \Gamma(\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}')\, dt \\
&= \frac{T_0}{\alpha} \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1 \cdot \Gamma(\boldsymbol{s}', \boldsymbol{v}', \boldsymbol{w}') \\
&\text{Since } T_0 \geq \frac{\delta}{\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) \cdot (\frac{1}{\alpha} - 1)} \text{ and } T_1 = 1, \\
&\text{plugging in and simplifying, we get} \\
&\geq \overline{\Gamma}(\mathcal{IC}^{\mathcal{H}})
\end{aligned}$$

Thus, in either case we get $\overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}) \geq \overline{\Gamma}(\mathcal{IC}^{\mathcal{H}})$, and hence Γ is not robust.

B.2 Proof of Theorem 2

Case $\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{A}}) \leq (1+\varepsilon)^2\, \overline{\Gamma}(\mathcal{IC}^{\mathcal{A}})$: By definition of every block b_i with timespan (t_i, t_i'), it follows that

$$\begin{aligned}
\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{A}}) &= \sum_{b_i \in \mathcal{BC}^{\mathcal{A}}} \Gamma\left(\mathbf{S}_{\blacksquare}^{\mathcal{A}}(b_i), \mathbf{V}_{\blacksquare}^{\mathcal{A}}(b_i), \mathbf{W}_{\blacksquare}^{\mathcal{A}}(b_i)\right) \\
&\leq \sum_{b_i \in \mathcal{BC}^{\mathcal{A}}} \Gamma\left(\mathbf{S}_{\blacksquare}^{\mathcal{A}}(b_i), \int_{t_i}^{t_i'} \boldsymbol{V}^{\mathcal{A}}(t)\, dt, \int_{t_i}^{t_i'} \boldsymbol{W}^{\mathcal{A}}(t)\, dt\right) \\
&\leq \sum_{b_i \in \mathcal{BC}^{\mathcal{A}}} \Gamma\left(\mathbf{S}_{\blacksquare,\max}^{\mathcal{A}}(b_i), \int_{t_i}^{t_i'} \boldsymbol{V}^{\mathcal{A}}(t)\, dt, \int_{t_i}^{t_i'} \boldsymbol{W}^{\mathcal{A}}(t)\, dt\right).
\end{aligned}$$

The third line follows by the monotonicity of Γ and the fact that $\mathbf{S}_{\blacksquare}^{\mathcal{A}}(b_i) \leq \mathbf{S}_{\blacksquare,\max}^{\mathcal{A}}(b_i)$ necessarily.

Using the previous insights about the average resources, we continue with

$$\begin{aligned}
\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{A}}) &\leq \sum_{b_i \in \mathcal{BC}^{\mathcal{A}}} \Gamma\left(\mathbf{S}_{\blacksquare,\max}^{\mathcal{A}}(b_i), \int_{t_i}^{t_i'} \boldsymbol{V}^{\mathcal{A}}(t)\, dt, \int_{t_i}^{t_i'} \boldsymbol{W}^{\mathcal{A}}(t)\, dt\right) \\
&= \sum_{b_i \in \mathcal{BC}^{\mathcal{A}}} (t_i' - t_i) \cdot \Gamma\left(\mathbf{S}_{\blacksquare,\max}^{\mathcal{A}}(b_i), \overline{\mathbf{V}_{\blacksquare}}^{\mathcal{A}}(b_i), \overline{\mathbf{W}_{\blacksquare}}^{\mathcal{A}}(b_i)\right) \\
&\leq \sum_{b_i \in \mathcal{BC}^{\mathcal{A}}} (t_i' - t_i) \cdot \Gamma\left(\mathbf{S}_{\blacksquare,\max}^{\mathcal{A}}(b_i), \mathbf{V}_{\blacksquare,\max}^{\mathcal{A}}(b_i), \mathbf{W}_{\blacksquare,\max}^{\mathcal{A}}(b_i)\right)
\end{aligned}$$

where the last line follows from monotonicity.

Now we invoke Def. 14 to switch max to min, that is,

$$\begin{aligned}
\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{A}}) &\leq \sum_{b_i \in \mathcal{BC}^{\mathcal{A}}} (t_i' - t_i) \cdot \Gamma\left(\mathbf{S}_{\blacksquare,\max}^{\mathcal{A}}(b_i), \mathbf{V}_{\blacksquare,\max}^{\mathcal{A}}(b_i), \mathbf{W}_{\blacksquare,\max}^{\mathcal{A}}(b_i)\right) \\
&\leq (1+\varepsilon) \sum_{b_i \in \mathcal{BC}^{\mathcal{A}}} (t_i' - t_i) \cdot \Gamma\left(\mathbf{S}_{\blacksquare,\min}^{\mathcal{A}}(b_i), \mathbf{V}_{\blacksquare,\max}^{\mathcal{A}}(b_i), \mathbf{W}_{\blacksquare,\max}^{\mathcal{A}}(b_i)\right) \\
&= (1+\varepsilon)^2 \sum_{b_i \in \mathcal{BC}^{\mathcal{A}}} (t_i' - t_i) \cdot \Gamma\left(\mathbf{S}_{\blacksquare,\min}^{\mathcal{A}}(b_i), \mathbf{V}_{\blacksquare,\min}^{\mathcal{A}}(b_i), \mathbf{W}_{\blacksquare,\min}^{\mathcal{A}}(b_i)\right)
\end{aligned}$$

where the second line follows from the sub-homogeneity of Γ in $\boldsymbol{S}$ and the third from the homogeneity of Γ in $(\boldsymbol{V}, \boldsymbol{W})$.

This implies the desired inequality because

$$\begin{aligned}
\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{A}}) &\leq (1+\varepsilon)^2 \sum_{b_i \in \mathcal{BC}^{\mathcal{A}}} (t_i' - t_i) \cdot \Gamma(\mathbf{S}^{\mathcal{A}}_{\blacksquare,\min}(b_i), \mathbf{V}^{\mathcal{A}}_{\blacksquare,\min}(b_i), \mathbf{W}^{\mathcal{A}}_{\blacksquare,\min}(b_i)) \\
&\leq (1+\varepsilon)^2 \sum_{b_i \in \mathcal{BC}^{\mathcal{A}}} \int_{t_i}^{t_i'} \Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))\, dt \\
&\leq (1+\varepsilon)^2 \int_0^T \Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))\, dt \\
&= (1+\varepsilon)^2\, \overline{\Gamma}(\mathcal{IC}^{\mathcal{A}}).
\end{aligned}$$

Note that the third line follows because the adversary may leave some gaps in time between blocks.