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

April 9, 2025

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 $\Gamma(S, V, W)$ (that assign a weight to a block as a function of the recorded space, speed, and work) are secure 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 in an idealized “continuous” model: $\Gamma(S, V, W)$ is secure if and only if it is homogeneous of degree one in the “timed” resources V and W, i.e., $\alpha\Gamma(S, V, W) = \Gamma(S, \alpha V, \alpha W)$. This includes the Bitcoin rule $\Gamma(S, V, W) = W$ and the Chia rule $\Gamma(S, V, W) = S \cdot V$. In a more realistic model where blocks are created at discrete time-points, one additionally needs some mild assumptions on the dependency on S (basically, the weight should not grow too much if S is slightly increased, say linear as in Chia).

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., $\sqrt{W_1} \cdot \sqrt{W_2}$ or $\min\{W_1, W_2\}$ are also secure, and we argue that in practice these are much better choices.

1 Introduction

Achieving consensus in a permissionless setting is a famously difficult problem. Nakamoto [Nak09] solved it by introducing the Bitcoin blockchain which achieves consensus on a chain of blocks by having parties expend a *resource*: parallelizable computation (commonly called *work*).

In Bitcoin, appending a block to a chain requires a *proof-of-work* (PoW), i.e., solving a computationally-expensive puzzle. This puzzle is designed such that each block represents the (expected) amount of computation that was expended to append it. As a consequence, each chain represents the total amount of computation required to create it. This allows for a simple consensus mechanism commonly called the *longest-chain rule*: Given two different chains, pick the one that required more computation to create. Note that a more accurate term is *heaviest-chain rule*, which we will use interchangeably throughout the paper.

While this design achieves consensus, more importantly it also achieves a property called *persistence* [GKL15] under a simple economic assumption: As long as honest parties control more than half of the computational resources committed to Bitcoin, a block that has been part of the chain for some time will always be part of the chain. Since Bitcoin blocks contain transactions, this effectively means that an adversary cannot *double-spend* a coin.

While Bitcoin's design is simple, its reliance on PoW has its flaws. For example, it wastes a lot of energy, and the manufacturing of the PoW hardware has become increasingly centralized.

Amongst other reasons, this has lead to the development of other blockchain protocols. These protocols can be broadly categorized along three axes.

The Underlying Resources Bitcoin relies on parallelizable computation, which is a *physical resource*. Two natural alternatives are *disk space* and *sequential computation*. A different class of resources is not physical, but *on-chain* [ACL+23, LPR24]. The most well-known is *stake*, which comprises different approaches that essentially rely on the on-chain coin balance of a party.

The Consensus Design A broad distinction is between Byzantine-fault-tolerant-style (e.g., Algorand [GHM+17] or Filecoin [Fil24]) and longest-chain protocols. Within themselves, longest-chain protocols differ in their *fork-choice rule*, i.e., how they select the longest chain. Some are *Nakamoto-like*, i.e., like Bitcoin they pick the heaviest chain, i.e., the one whose blocks cumulatively required the most resources to create (e.g., Chia [CP19]). Others rely on more complex fork-choice rules, which, e.g., take into account where two chains fork (e.g., Ouroboros [BGK+18a]).

The Degree of Permissionlessness Roughgarden and Lewis-Pye [LPR24] observe that "permissionless" is colloquially used to describe different settings that vary in how permissionless they are. *Fully-permissionless* protocols function obliviously to current protocol participants (e.g., Bitcoin or Chia). These differ from protocols that require some information about participants (e.g., how many coins they are staking in Algorand, or commitments to disk space in Filecoin). While the latter are still permissionless in the sense that anyone can participate, they impose stricter requirements on participants.

1.1 Our Contributions

In this paper, we completely characterize the design space of *Nakamoto-like* protocols operating in the *fully-permissionless* setting using *physical resources* (i.e., disk space, sequential computation, and parallelizable computation) that are secure against *double-spending* attacks.

We observe that Nakamoto-like protocols only differ in what resources their blocks record, and—especially if multiple resources are used—how they decide which of two blocks required *more resources* to create. We model these differences using an abstraction called the *weight function* $\Gamma\colon \mathbb{R}^3 \to \mathbb{R}$. It takes as input the three resources possibly recorded by a block, i.e., disk space S, sequential computation V[1] and parallelizable computation/work W, and outputs the *weight* $\Gamma(S, V, W)$ of a block. In the context of weight functions, the heaviest-chain rule now picks the chain with the highest weight where a chain's weight is defined as the sum of the weight of all its blocks.

To get an intuition for the weight function abstraction, let us provide some examples. The weight function $\Gamma_{\text{Bitcoin}}(S, V, W) = W$ describes Bitcoin (or any other similar PoW-based Nakamoto-like chain, e.g., Litecoin). More interesting is Chia [CP19], a Nakamoto-like chain combining disk space and sequential computation following $\Gamma_{\text{Chia}}(S, V, W) = S \cdot V$.

Our main result, informally stated in Theorem 1 below, fully characterizes which weight functions result in a Nakamoto-like blockchain that is secure against private double-spending attacks [Nak09, DKT+20] in the fully-permissionless setting [LPR24].

In this work we just address double spending, but not economic attacks as selfish mining [ES14]. Preventing double spending gives some additional guarantees, like the fact that one can trust the timestamps on the chain [TSZ23].

To achieve a broad and simple characterization, we necessarily have to abstract implementation details and generalize over different blockchain designs. In particular, we ignore issues like network delays or the fact that the quantitative resources (parallel work W and space S) can only be approximately recorded in a block. The reason is that taking such issues into account would only quantitatively affect our result and not give any interesting insights. There are attacks like grinding and double dipping[2] against chains that use space, but as we have techniques to prevent them [PKF+18, BDK+22], we also ignore those.

One annoying issue are so called replotting attacks. As we'll discuss in §4.3, in practice replotting can be prevented with a careful design putting bounds on the total weight of individual blocks. But as replotting is not as well understood as the other issues, we will explicitly exclude replotting in the statement of the theorem below. We'll discuss our model in more detail in §1.3.

Theorem 1 (Main, (Informal))**.** *In the fully-permissionless [LPR24] setting and ignoring replotting attacks, a Nakamoto-like blockchain is secure against private double-spending attacks under the honest majority assumption (cf. below)* if and only if *the weight function* $\Gamma(S, V, W)$ *fulfills the following conditions:*

1. ***Monotone:*** Γ *is monotonically increasing*

2. ***Homogeneous in*** V, W***:*** $\alpha\Gamma(S, V, W) = \Gamma(S, \alpha V, \alpha W)$ *for* $\alpha > 0$

The honest majority assumption states that at any point in time during the attack Γ *applied to the resources of the honest parties is larger than* Γ *applied to the resources of the adversary.*

[1] Think V as in velocity of the sequential computation or V as in verifiable delay function (VDF), the cryptographic primitive usually used to capture the number of sequential computation steps.

[2] A high level overview on these attacks can be found on https://docs.chia.net/longest-chain-protocols/

1.2 Implications of our Result

Space-based Blockchains. There exist three blockchains whose main resource is disk space: Chia [CP19], Filecoin [Fil24], and Spacemint [PKF$^+$18]. The first two are deployed and running in practice, while the latter is an academic proposal.

Of the above, Chia is the only one captured by our result, i.e., it is a Nakamoto-like protocol in the fully-permissionless setting. Its weight function is $\Gamma_{\mathsf{Chia}}(S, V, W) = S \cdot V$ which is secure against double-spending attacks by our Thm. 1.

In contrast, Filecoin is not captured by our result because it is not fully-permissionless, but instead operates in the stronger quasi-permissionless setting (cf. [LPR24]).[3] Since Filecoin's weight function is $\Gamma_{\mathsf{Filecoin}}(S, V, W) = S$, Thm. 1 essentially shows that a setting stronger than the fully-permissionless one is necessary. Finally Spacemint also uses basically $\Gamma_{\mathsf{Spacemint}}(S, V, W) = S$ and is fully-permissionless, as a consequence it's not secure (when using the heaviest-chain rule) as stated in the theorem.

Combining multiple PoWs. Since the production of Bitcoin mining-hardware has become increasingly centralized, one might consider combining two different PoWs, e.g., W_1 using SHA256 and W_2 using Argon2. As stated above, Thm. 1 only considers one parallel work W, but it naturally extends to multiple resources of each type. In particular, our results capture weight functions such as $\Gamma(W_1, \ldots, W_k)$ with Condition 2 being $\alpha\Gamma(W_1, \ldots, W_k) = \Gamma(\alpha W_1, \ldots, \alpha W_k)$ for $\alpha > 0$.

Prior work [FWK$^+$22] suggested the weight function $\Gamma(W_1, \ldots, W_k) = \sum_{i=1}^{k} \omega_i W_i$ with constants ω_i. By Thm. 1, this is secure, but not a desirable weight function in practice. Even though the constants ω_i can be used to calibrate the contribution of each PoW, it seems difficult to realize this in a way that would prevent miners to ultimately only invest in the cheapest PoW.

Our result show that more interesting combinations of $W_1, \ldots, W_k$ are possible. The first draws inspiration from automated-market-makers[4] and is defined as

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

To maximize this weight function (for a given budget), one would have to invest into mining hardware for all PoWs at a similar rate.

Another option is the Leontief utilities function[5]

$$\Gamma(W_1, \ldots, W_k) = \min\{W_1, \ldots, W_k\}$$

which ensures that all PoWs must significantly contribute.

Our work just classifies the weight functions that are secure in the sense that we get security whenever the honest parties control resources of higher weight (as specified by the weight function). A question that is mostly orthogonal to this work is to investigate which

[3] Filecoin is also not Nakamoto-like since it is a DAG-based protocol (using GHOST, the *Greediest Heaviest-Observed Sub-Tree* rule [SZ13]) together with a finality gadget [ADH$^+$23]. Note that the finality gadget is not essential, and GHOST is the DAG-analogue to Bitcoin's longest/heaviest-chain rule. So, for our purposes, Filecoin could easily be modified to be Nakamoto-like (this has also been mentioned in [ACL$^+$23]). As we'll elaborate in §4.3, it seems running a space based chain in the quasi-permissionless setting is quite expensive as to prevent reploting parties must constantly prove they hold the committed space and this proofs need to be recorded on chain.

[4] https://en.wikipedia.org/wiki/Constant_function_market_maker

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

of those weight functions are also interesting from a practical perspective, say because they incentivize decentralization or other desirable properties. Let us observe that the class of secure weight functions does contain functions that make little sense in practice, for example the function $\Gamma(V) = V$ which simply counts the number of VDF steps. A blockchain based on this weight function would be secure assuming some honest party holds a VDF that is faster than the VDF held by the adversary.

1.3 Model and Modelling Rationale

Modelling Resources. Our model captures resources that are *external* to the chain, i.e., *physical resources.* In particular, we consider disk space S, sequential computation V, and parallelizable computation W where we allow multiple resources per type, e.g., W_1 and W_2.[6] Each resource is modelled as a function mapping time $\mathbb{R}_{\geq 0}$ to an amount $\mathbb{R}_{>0}$. This is expressive enough to capture, e.g., Bitcoin, any other PoW-based blockchain, or Chia.

In practice, cryptographic primitives are used to track these resources, usually *Proof-of-Space* (PoSpace) [DFKP15], *Verifiable Delay Functions* (VDFs) [BBBF18, Wes19, Pie19], and *Proof-of-Work* (PoW). Our modelling essentially assumes a perfect primitive, glossing over implementation details and any probabilistic nature of the resource (similar to [Ter22]).

In practice parallel work W as captured by a PoW, and sequential work V as captured by a VDF are very different. W is a quantitative resource in the sense that one can double it by investing twice as much, while V is a qualitative resource as it measures the speed of the fastest available VDF. From the perspective of our Theorem on the other hand, W and V behave the same, the only thing that matters in the (proof of the) Theorem is that W and V are "timed" resources in the sense that their unit is something "per second". W and S on the other hand are both quantitative resources, but behave very differently.

Reasons for Omitting Stake. First, as described by Roughgarden and Lewis-Pye [LPR24], stake-based blockchains usually do not operate in the fully-permissionless setting. Therefore, since our result targets this setting, modelling stake is not as important. Nevertheless, simple stake-based blockchain designs are in principle possible in the fully-permissionless setting.

Another reason is that in our modeling we assume that parties hold some resources at some given time, for an on-chain resource like stake this is not well defined as the resource is only defined with respect to some particular chain. To make issues even more tricky, with stake it's possible to obtain old keys that no longer hold any value, but still can be used in an attack [ACL+23].

The Continuous Chain Model. Towards Thm. 1, we will first consider the *Continuous Chain Model.* While it is a very abstract model, it is rich enough to already yield Conditions 1 and 2. In a nutshell, we assume that a chain *continuously* and *exactly* reflects the resources that were expended to create it.

Assume that the honest parties at time t hold resources $S(t), W(t), V(t)$, then the chain they can create in a time window $[t_0, t_1]$ will have weight $\int_{t_0}^{t_1} \Gamma(S(t), V(t), W(t))\, dt$.

This continuous model avoids some issues of actual blockchains, like their probabilistic nature or network delays. For example in Bitcoin, a so called 51% attack can actually

[6] These are three fundamental resources in computation, and also the most popular physical resources used for blockchains. Nevertheless, we believe our model could be extended to other external resources.

be conducted with less, say, 41% of the hashing power if network delays are sufficiently large. Moreover, as a Bitcoin block is found every 10 minutes *in expectation*, it frequently happens that no block is found in an hour at all. For this reason a block is only considered confirmed if it's sufficiently deep in the chain. These factors only have a *quantitative* impact on the concrete security threshold of longest-chain blockchains and are well understood [GKR20, DKT^{+}20]. The goal of this paper, however, is *qualitative* in nature. That is, we want to describe which weight functions are secure as long as honest parties have sufficiently more resources than the adversary. Precisely quantifying how much security is lost due to the fact that resources are only approximately recorded, due to network delays or other aspects like double dipping attacks is not our goal.

Private Double-Spending Attack. We analyze the security of weight functions against a specific attack, the *Private Double-Spending* (PDS) attack. As Dembo et al. [DKT^{+}20] showed, PDS is the worst attack against Bitcoin and other longest-chain protocols. This also verifies the intuition of Nakamoto, who also only considered the double-spend attack against Bitcoin [Nak09].

In a PDS attack, the adversary forks the chain at some point in time, privately extends its own fork (while honest parties continue to extend the main chain), and releases its fork later on. The attack is successful if the adversary's fork is at least as heavy as the honest chain since this would allow the adversary to double-spend a transaction.

We let the adversary choose the resources available to honest parties and itself during this time. The only condition is that we disallow the adversary to trivially perform a successful attack. That is, at every point in time the adversary resources have at most as much weight as the resources of honest parties, and, to avoid a draw, in some interval strictly less.

The honest chain directly corresponds to the resources of the honest parties. The adversary, however, may cheat since it is mining in private. In particular, it can pretend to have created the chain in a shorter or longer amount of time by stretching/squeezing time. This time manipulation affects the resources recorded on the chain. For example, consider W as the hash rate, then a chain records the total number of hashes. If the adversary now pretends to have created this chain in $1/2$ time, then its hashrate must be $2 \cdot W$ since the total amount of hashes does not change.

Given such an attack, e.g., the weight function $\Gamma(W) = W^2$ is insecure. Indeed, consider an adversary with resource $W(t) = 1$ and honest parties with $W(t) = 2$. Honest parties mining for 1 time create a chain of weight $1 \cdot 2^2 = 4$. In the same timespan, the adversary creates a chain of weight $1 \cdot 1^2 = 2$. However, if it pretends to have mined this chain privately in $1/8$ time, then its chain records the weight $1/8 \cdot (8 \cdot 1)^2 = 8$ instead, beating the honest chain.

We defer more precise definitions and figures exemplifying this time manipulation to § 3.

The Discrete Chain Model. So far, we discussed an abstract model where the chain *continuously* and *exactly* records resource expenditure.

In §4 we discuss a model closer to a real blockchain, where blocks arrive in discrete time slots. We still assume the block exactly records the resources W and V. In particular, a block produced during some timespan $[a, b)$ records $W_{\blacksquare} = \int_a^b W(t)\, dt$ and $V_{\blacksquare} = \int_a^b V(t)\, dt$. For the space we assume that the block records $S(t)$ at some point $a \leq t < b$. The reason for this difference is that W and V are resources that are measured *per second* (e.g., hashes/s or steps/s), so integration over time is well-defined. One the other hand, a proof of space gives a snapshot of the space $S(t)$ available at some point t during block creation. To be on the safe side, we simply assume that the adversary can choose the t where its space was maximal, while for the honest parties we assume t is the time where $S(t)$ is minimal.

We show (Theorem 3) that the classification of secure weight functions basically carries over to this discrete setting as long as the resources don't vary by too much within the block arrival time. But our main motivation to consider the discrete model is to discuss the issue of replotting attacks in §4.3, which only make sense in a discrete setting.

1.4 Future Work

Our work opens multiple new questions for future work. We already mentioned identifying weight functions that are not only secure, but also interesting at the end of section §1.2. At the end of section §4.3 we will discuss an open question concerning replotting attacks. Some other open questions include:

First, modelling on-chain resources, most notably, stake. While stake somewhat behaves like disk space[7], it is different and difficult to model since it is an on-chain resource. For example, one modelling challenge is capturing long range attacks in which parties sell old keys that controlled a lot of stake at some point. This is similar to a bootstrap attack for disk space, but the difference is that the adversary can perform this attack for free (after having bought the keys).

Second, considering chain-selection rules other than the heaviest-chain rule. For example, in the stake setting, Ourboros Genesis [BGK+18b] operating in in dynamically available setting achieves security against PDS using a different chain selection rule.

Third, considering different degrees of permissionless, such as the dynamically-available or quasi-permissionless setting described by [LPR24]. While our results rule out solely using disk space in the fully-permissionless setting, this impossibility does not carry over to other models. For example, Filecoin [Fil24] only uses disk space, but is secure against PDS because it operates in the quasi-permissionless model.

1.5 Related Work

Abstract Resource Models. Recently, Roughgarden and Lewis-Pye [LPR24] (an updated version of [LPR23]) presented many (im)possibility results about permissionless consensus. They consider a resource-restricted adversary where resources can be external or on-chain resources. External resources are modelled by so-called *permitter oracles*, whose outputs depend on the amount of resources the querying party has at the time of the query. An important part of their work is a classification of the permissionlessness of consensus protocols:

- *Fully-permissionless* protocols are oblivious to its participants (e.g., Bitcoin).
- *Dynamically-available* protocols know a dynamic list of parties, which may be a function of the past protocol execution (e.g., parties who staked coins), the participants are a subset of this list, and *at least one* honest member of this list participates.

[7] There exist proposals similar to Chia that use stake instead of disk space, i.e., where the weight is $\text{Stake} \cdot V$ [DKT21].

- *Quasi-permissionless* protocols are similar to dynamically-available protocols, but make the stronger requirement that *all* honest members of the list participate. Note that such protocols differ from *permissioned* ones, which also have a list of parties, but where the list *cannot* depend on the past protocol execution.

An exemplary result of theirs states, e.g., no deterministic protocol solves Byzantine agreement in the fully-permissionless setting, even with resource restrictions.

Two preceding works modelling abstract resources are Terner [Ter22] and Azouvi et al. [ACL+23]. Terner [Ter22] considers an abstract resource that essentially is a black-box governing participant selection. They give a consensus protocol that can be instantiated with any such resource satisfying certain properties (e.g., resource generation must be rate-limited relative to the maximum message delay).

Azouvi et al. [ACL+23] use the abstraction of resource allocators (similar to permitter oracles in [LPR24]) to build a total-ordered broadcast protocol. They describe the properties a resource allocator must fulfill (e.g., honest majority), and construct resource allocators for the resources stake, space[8], and work. As part of this, they classify resources as external vs. on-chain (they call it virtual), and burnable vs. reusable (space and stake are reusable whereas work is not) and discuss trade-offs between different types of resources, e.g., on-chain resources are susceptible to long-range attacks.

Blockchain Designs. We give a selection of well-known permissionless blockchain designs, describing the weight function or—for non-Nakamoto-like protocols—resource (S, V and W as before, and stake St) and degree of permissionlessness (**f**ully-**p**ermissionless, **d**ynamically-**a**vailable, or **q**uasi-**p**ermissionless): Bitcoin [Nak09] (fp, W), Chia [CP19] (fp, $S \cdot V$), Filecoin [Fil24] (qp, S), Ethereum [Woo] (da/qp,[9] St), Algorand [GHM+17] (qp, St), Ouroboros [KRDO17, DGKR18, BGK+18a] (da/qp, St), Snow White [DPS19] (da, St).

Multiple combinations of proof-of-stake (PoStake) and PoW, e.g., [KN12, BLMR14, FWK+22] exist (all either da or qp). [FWK+22] is the only *fungible* protocol, i.e., it is secure as long as the adversary controls less than half of all stake and work *cumulatively* (essentially mapping to $\Gamma(St, W) = St + W$). Their protocol handles multiple PoStake and multiple PoW resources, thus capturing $\Gamma(W_1, W_2) = W_1 + W_2$, which we discussed in § 1.2.

[DKT21] combine PoStake with sequential computation to create a dynamically-available protocol.

Ignoring difficulty, Bitcoin assigns unit weight to every block whose hash surpasses a threshold. [KMM+21] analyze other functions assigning weight to block hashes. They suggest using a function that grows exponentially, but is capped at a certain value, which depends on the maximum network delay.

Analyses of Blockchain Protocols. Various works analyze specific blockchains, mostly Bitcoin (or similar PoW chains) [GKL15, PSs17, PS17, Ren19, GKR20], but also longest-chain protocols in general [DKT+20]. These generally give quantitative security thresholds (i.e., what fraction of adversarial resources is tolerable) depending on, e.g., maximum message delay. We again remark that our work has a different aim, namely a qualitative description of weight functions, disregarding precise security thresholds.

[8] Their space allocator lies in the quasi-permissionless model, thus not conflicting with our results.

[9] Depending on whether the network is synchronous/partially-synchronous [LPR24].

2 Preliminaries

Let $[n] = \{1, \dots, n\}$. 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 $(x_1, \dots, x_n), (x'_1, \dots, x'_n) \in \mathbb{R}^n_{>0}$ we say $(x_1, \dots, x_n) \leq (x'_1, \dots, x'_n)$ 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 the time by $t \in \mathbb{R}_{\geq 0}$. 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. The open interval $(T_0, T_1]$ denotes the time interval $[T_0, T_1]$ excluding T_0. $[T_0, T_1)$ is defined analogously.

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

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

Definition 2 (Homogeneity)**.** A function $f \colon \mathbb{R}^n_{>0} \to \mathbb{R}_{>0}$ is homogeneous[10] in $x_j, \dots, x_n$ with $0 \leq j \leq n$ if, for all $(x_1, \dots, x_n) \in \mathbb{R}^n_{>0}$ and $\alpha > 0$,

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

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

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

3 Continuous Chain Model

To characterize which weight functions provide security against private double-spending (PDS) attacks, we will first introduce the *Continuous Chain Model.* It models physical resources, how resources are turned into an idealized blockchain, and private double spending (PDS) attacks. As the name suggests, the continuous model views the blockchain as one continuous object, instead of consisting of multiple discrete blocks.

3.1 Modelling Resources

The model captures physical resources, which are external to the chain, and allows for multiple resources per type. The resources are disk space $\boldsymbol{S} \coloneqq (S_1, \dots, S_{k_1})$, sequential work $\boldsymbol{V} \coloneqq (V_1, \dots, V_{k_2})$, and parallel work $\boldsymbol{W} \coloneqq (W_1, \dots, W_{k_3})$.[11] It allows for multiple resources per type. We will omit k_1, k_2, and k_3 unless needed for clarity.

A *resource profile* records the amount of each resource available at any point in time. Time is modelled as a continuous variable $t \in \mathbb{R}_{\geq 0}$, and we restrict our attention to the time interval $[0, T]$ for some $T > 0$. We take each resource to be a function mapping this interval to $\mathbb{R}_{>0}$, e.g., $W_1 \colon [0, T] \to \mathbb{R}_{>0}$.

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

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

where each tuples of functions is composed of functions with domain $t \in [0, T]$ with $T > 0$ and range $\mathbb{R}_{>0}$, and where each function is Lebesgue integrable.

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

[11] These are three fundamental resources in computation and also the most popular physical resources used for blockchains. Nevertheless, we believe our model could be extended to other external resources.

Remark 1. The requirement that each resource is non-zero at every point in time is a minor technical condition. Note that it is always fulfilled in practice since interaction with the blockchain requires a general-purpose computer, and even a low-powered one provides a non-zero amount of S, V, and W.

3.2 Idealized Chain

Ideally, a blockchain should record the amount of resources that were expended to create it, and blockchain protocols are generally designed to approximate this as closely as possible. In our idealized model, we assume that a blockchain *continuously* and *exactly* reflects the resources expended to create it.

Definition 5 (Continuous Chain Profile)**.** A continuous chain profile $\mathcal{CC}$ is a 3-tuple of tuple of functions

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

where each tuples of functions is composed of functions with domain $t \in [0, T]$ where $T > 0$ and range $\mathbb{R}_{>0}$, and where each function is Lebesgue integrable.

Remark 2. Resource and chain profiles are syntactically identical. The difference lies in semantics: A resource profile describes the resources available to a party (or a set of parties). Meanwhile, a chain profile describes the resources that the chain reflects.

Remark 3. In practice, blockchains do not *exactly* record the amount of resources, but only approximate them. For example, in Bitcoin, finding blocks is a probabilistic process, so blocks do not record the actual work invested to create them, but only the *expected* amount of work. Additionally, network delays cause miners to waste time (and thereby work) trying to extend an out-of-date block, in the worst case leading to orphaned blocks. In spite of these issues, the ideal model is still meaningful because these issues introduce *quantitative* gaps (e.g., [DKT+20, GKL15, GKR20]).

To capture the heaviest-chain rule, the model assigns each chain a weight. To this end, we first introduce the *weight function* Γ, which assigns a weight to a triple of resources. In other words, it assigns a weight to one point in time.

Definition 6 (Weight Function)**.** A weight function is a non-constant function given by

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

Remark 4. The requirement that Γ is non-constant is natural in practice. We explicitly require it because such functions would be vacuously secure. Looking ahead, the security definition only considers adversaries with a resource disadvantage, which is measured using weight. But if the weight is constant, no such adversary exists, so the function would always be secure.

As said, Γ takes in three resources and outputs the weight of a specific point in time. In the next step, we extend Γ to compute the weight of a whole chain. We denote this function by $\overline{\Gamma}$. It takes a continuous chain profile as input and outputs the weight of it. Overloading notation slightly, we also allow inputting a resource profile to $\overline{\Gamma}$ since it is syntactically identical to a chain profile.

Definition 7 (Weight of a Chain or Resource Profile)**.** Consider a weight function Γ and a continuous chain $\mathcal{CC} = (\boldsymbol{S}(t), \boldsymbol{V}(t), \boldsymbol{W}(t))_{[0,T]}$ or resource profile $\mathcal{R} = (\boldsymbol{S}'(t), \boldsymbol{V}'(t), \boldsymbol{W}'(t))_{[0,T]}$. The chain weight function $\overline{\Gamma}$ is defined as

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

and

$$\overline{\Gamma}(\mathcal{R}) \coloneqq \int_0^T \Gamma(\boldsymbol{S}'(t), \boldsymbol{V}'(t), \boldsymbol{W}'(t))\, dt.$$

3.3 The Private Double-Spending Attack

In a private double-spending (PDS) attack, the adversary forks the chain at some point in time, extends this fork in private, before releasing the private fork to the public. The attack is successful if the adversary's fork is heavier than the honest chain, because the adversarial fork replaces the honest chain, effectively reverting past transactions.

We focus on the PDS attack because it is the prototypical attack against blockchains. This is for a good reason. As Dembo et al. [DKT+20] showed, PDS is the worst attack against Bitcoin and other longest-chain protocols. This also verifies the intuition of Nakamoto, who also only considered the double-spend attack against Bitcoin [Nak09].

Modelling the Attack. To model this attack, we first consider the time frame of the attack. The attack starts (i.e., the adversary forks the chain) at time 0, and the adversary publicly publishes its private chain at time T_{end}. So the attack spans the time interval $[0, T_{\mathsf{end}}]$. During this time, the resources available to the honest parties are given by the resource profile $\mathcal{R}^{\mathcal{H}}$, and they use them to build the honest chain profile $\mathcal{CC}^{\mathcal{H}}$ in the following way:

Definition 8 (Honest Chain Profile)**.** Consider a resource profile $\mathcal{R}^{\mathcal{H}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0,T_{\mathsf{end}}]}$ The corresponding honest chain profile is $\mathcal{CC}^{\mathcal{H}} \coloneqq \mathcal{R}^{\mathcal{H}}$.

That is, the honest chain $\mathcal{CC}^{\mathcal{H}}$ correctly reflects the resources available to honest parties, and also precisely keeps track at which point in the resources were available.[12]

The adversary's resources are $\mathcal{R}^{\mathcal{A}}$, and they use them to build the chain profile $\mathcal{CC}^{\mathcal{A}}$.[13] In contrast to the honest parties, the adversary may deviate from the protocol and cheat. First, they may simply not use some of the resources available to them. Second, and more importantly, since the adversary creates the fork in private, the chain $\mathcal{CC}^{\mathcal{A}}$ may not correctly reflect *at what time* the resources were available. In essence, the adversary can *alter* the time by *stretching* and *squeezeing* it. For example, in Bitcoin the adversary may mine a block in 100 minutes but pretend to have mined it within 10 minutes.

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 alteration. Altering the time affects, e.g., the length of the interval $[0, T_{\mathsf{end}}]$. To this end, we introduce the *altered time* function AT (and its inverse AT^{-1})[14] to translate

[12] While this is by construction in our idealized model, timestamps are generally accurate in longest-chain blockchains—even if a not-too-powerful adversary tries to disrupt them [TSZ23].

[13] In general, we use the superscripts $\mathcal{H}$ and $\mathcal{A}$ to denote the honest parties and the adversary, respectively.

[14] 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 $[0, \mathsf{AT}(T_{\mathsf{end}})]$.

between time before and after squeezing/stretching. For example, $\widetilde{T}_{\mathsf{end}} = \mathsf{AT}(T_{\mathsf{end}})$, resulting in the interval $[0, \widetilde{T}_{\mathsf{end}}]$.[15]

Altering time affects how $\mathcal{CC}^{\mathcal{A}}$, which covers the time interval $[0, \widetilde{T}_{\mathsf{end}}]$, reflects resources. For the resources V and W, altering time cannot change the cumulative amount (e.g., in Bitcoin it cannot change the number of found blocks and thus work performed). Therefore, V and W must be multiplied by ϕ. That is, at altered time $\tilde{t} \in [0, \widetilde{T}_{\mathsf{end}}]$, $\mathcal{CC}^{\mathcal{A}}$ records the resource $\phi(t)V^{\mathcal{A}}(t)$ and $\phi(t)W^{\mathcal{A}}(t)$. The disk space S behaves differently. As long as it is available, it can be reused [ACL+23], so it does not accumulate (unlike V and W). As a consequence, altering time just changes when space was available. Thus, at altered time $\tilde{t} \in [0, \widetilde{T}_{\mathsf{end}}]$, $\mathcal{CC}^{\mathcal{A}}$ records $S(\mathsf{AT}^{-1}(\tilde{t}))$.

Definition 9 (Adversarial Chain Profile)**.** 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 $\mathsf{AT}(t) := \int_0^t \frac{1}{\phi(u)}\, du$ and its inverse $\mathsf{AT}^{-1}(\cdot)$.

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

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

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

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

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

Let us illustrate the stretching and squeezing from Def. 9 by the example of Bitcoin and Chia in Figs. 1 and 2.

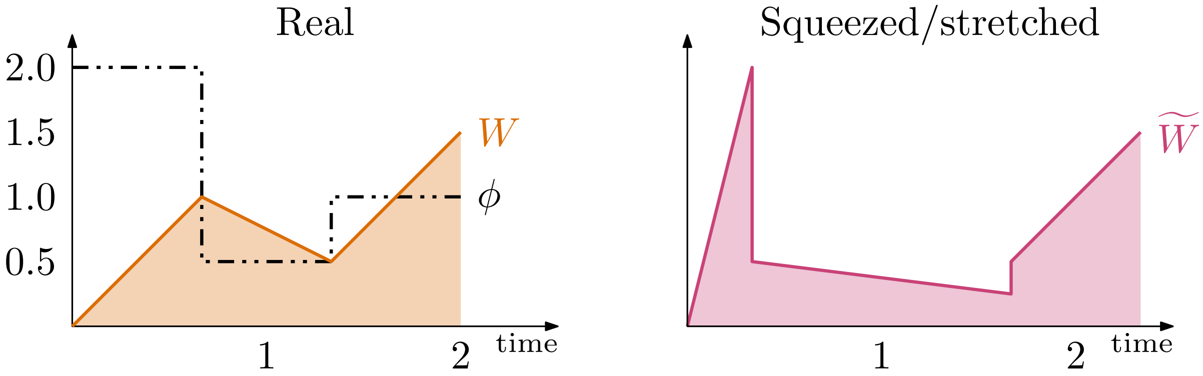


Figure 1: Bitcoin's weight function W and how it reacts to stretching and squeezing. The shaded area is the weight.

We now have all ingredients to define when a weight function is secure against PDS attacks. On a high level, the definition states that an adversary having resources of less weight than the honest parties[16] cannot create a private chain that is heavier than the honest parties one—even by manipulating time. In more detail, "less weight" means that the adversary has at most equal weight at every point in time (Eq. (1)), and in some interval it has strictly less (Eq. (2)).

[15] We use $\tilde{}$ to denote time after squeezing/stretching.

[16] Clearly, a PDS attack always works when the adversary has a resource advantage.

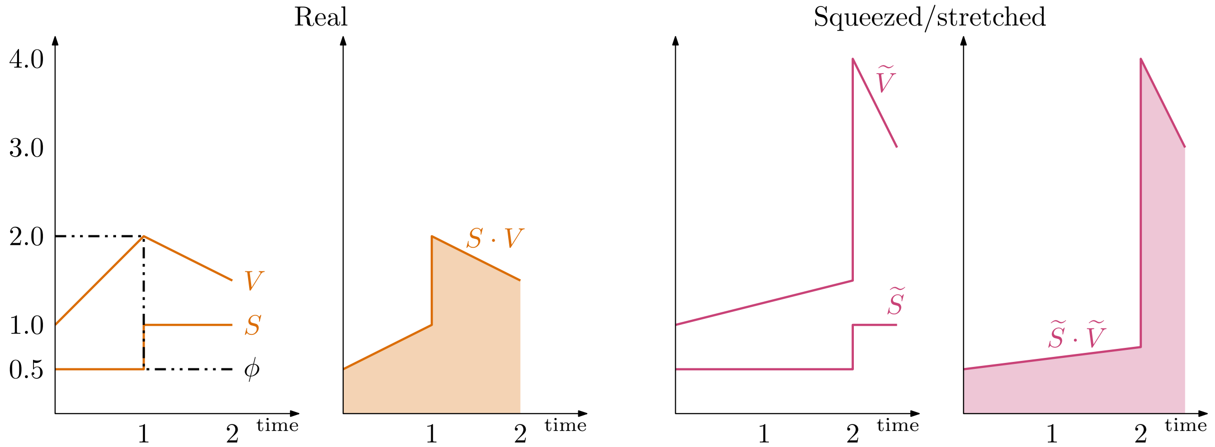


Figure 2: Chia's weight function $S \cdot V$ and how it reacts to stretching and squeezing. The shaded are is the weight.

Definition 10 (Secure Weight Function, Continuous Model)**.** A weight function Γ is *secure against private double-spending attack* in the *continuous model* 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

$$\Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t)) \leq \Gamma(\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t)) \; \forall t \in [0, T_{\mathsf{end}}] \tag{1}$$

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

$$\Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t)) < \Gamma(\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t)) \; \forall t \in [T_0, T_1] \tag{2}$$

it holds that

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

where $\mathcal{CC}^{\mathcal{H}} := \mathcal{R}^{\mathcal{H}}$ and $\mathcal{CC}^{\mathcal{A}}$ satisfies Def. 9 for $\mathcal{R}^{\mathcal{A}}$ and any $\phi(t)$.

Remark 5. An alternative to the precondition (Eqs. (1) and (2)) on resource profiles in Def. 10 is that adversarial resources must be strictly smaller than the honest ones at every point in time (instead of just for an interval). Looking ahead, Thm. 2 would be true in the if-direction (monotonically increasing and homogeneous implies secure), but not in the only-if direction. The reason is that not every non-homogeneous function can be attacked (e.g., a function that is $S \cdot \max(V, W)$ when each resource is below some constant threshold c and that is constant c^2 after that). If we additionally put the natural constraint that a 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 only-if direction also holds (by an adaption of our proof). Either way, the main takeaway is that monotonically increasing and homogeneous are the secure functions, and they are the only ones that should be used to construct Nakamoto-like blockchains using multiple resources.

3.4 Main Theorem in the Continuous Model

There are many possible choices for Γ, but not all are secure against PDS, i.e., a bad choice for a blockchain. For example, Fig. 3 shows that $W_1 \cdot W_2$ is insecure, but that $\sqrt{W_1} \cdot \sqrt{W_2}$ seems secure—at least in the example. The following theorem shows that it is secure in general, because it is monotone (cf. Def. 1) and homogeneous in $\boldsymbol{W}$ and $\boldsymbol{V}$

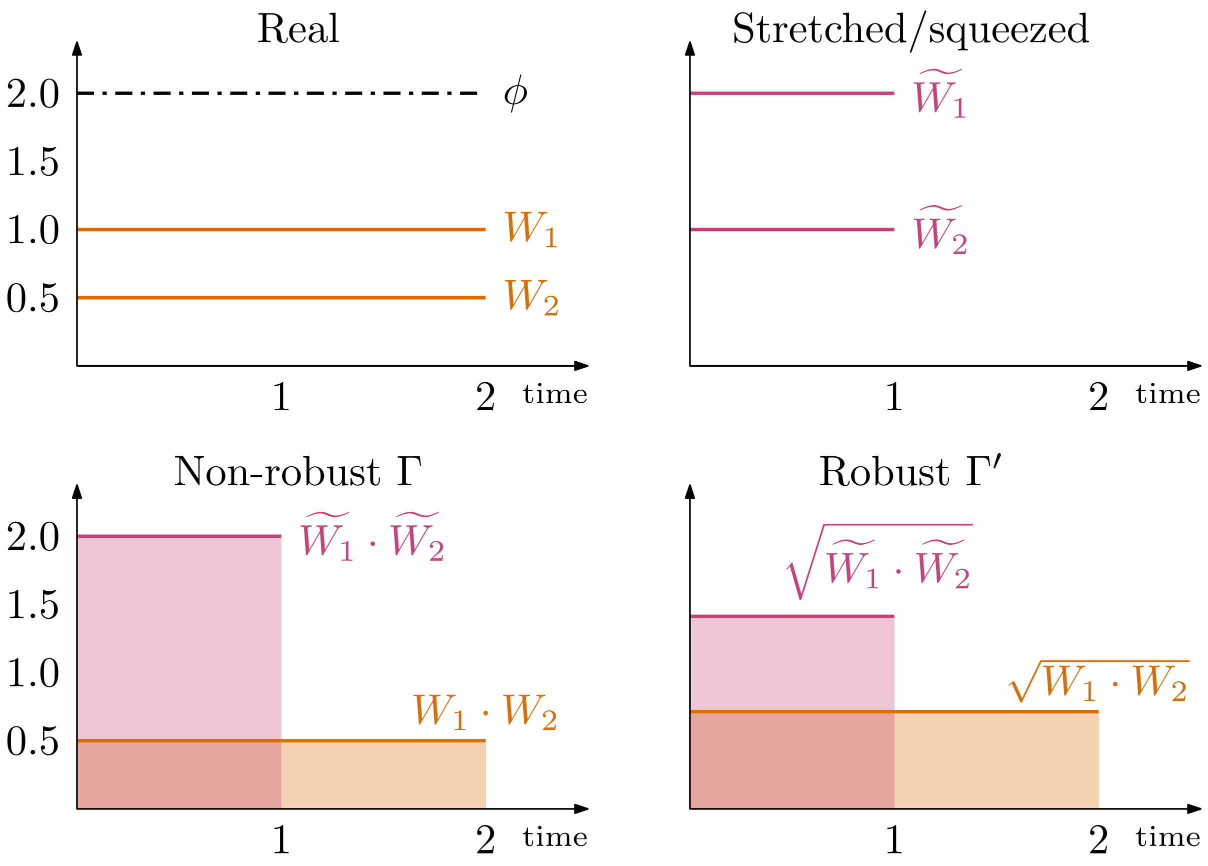


Figure 3: Consider two PoWs W_1, W_2, and two weight functions $\Gamma(W_1, W_2) = W_1 \cdot W_2$ and $\Gamma'(W_1, W_2) = \sqrt{W_1 \cdot W_2}$. The top row show the real resources W_1, W_2 (left) and how squeezing them by $\phi(\cdot) = 2$ (left) results in $\widetilde{W}_1, \widetilde{W}_2$ (right). The bottom row shows that Γ is not secure 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 not affected by the squeezing.

(i.e., $\alpha\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W}) = \Gamma(\boldsymbol{S}, \alpha\boldsymbol{V}, \alpha\boldsymbol{W})$, cf. Def. 2). These are sufficient, but also necessary conditions for security against PDS in the continuous chain model.

Theorem 2 (Secure Weight Functions)**.** *A weight function* Γ *is* secure against private double-spending attacks *in the* continuous model *if and only if* $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ *is monotonically increasing (Def. 1) and homogeneous in* $\boldsymbol{V}, \boldsymbol{W}$ *(Def. 2).*

We will prove the theorem in three parts using Lems. 1 to 3.

Lemma 1 (If-Direction of Thm. 3)**.** *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* $\overline{\Gamma}(\mathcal{R}^{\mathcal{A}}) \geq \overline{\Gamma}(\mathcal{CC}^{\mathcal{A}})$ *where* $\mathcal{CC}^{\mathcal{A}}$ *satisfies Def. 9 for* $\mathcal{R}^{\mathcal{A}}$ *and any* $\phi(t)$*.*

As a consequence, Γ *is secure 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 $\mathcal{CC}^{\mathcal{A}}$ from Def. 9. For any $\tilde{t} \in [0, \widetilde{T}_{\mathsf{end}}]$, $\mathcal{A}$ could create a chain profile such that

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

Since Γ is monotonic,

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

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

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

so we can conclude that

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

Now, we integrate by substituting[17] $t = \mathsf{AT}^{-1}(\tilde{t})$. Here, note that $\frac{d}{d\tilde{t}}\mathsf{AT}^{-1}(\tilde{t}) = \phi(\mathsf{AT}^{-1}(\tilde{t}))$ by the inverse function rule.[18] This leads to

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

This proves the first part of the lemma.

For the second part, note that the preconditions on resources in Def. 10 imply that

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

By the first part of this lemma and since $\mathcal{R}^{\mathcal{H}} = \mathcal{CC}^{\mathcal{H}}$ by Def. 10, the second part follows. □

Lemma 2 (Only-If-Direction of Thm. 2, Part I)**.** Γ *is not secure 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 $T_{\mathsf{end}} > 0$, consider the resource profiles

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

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

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

Clearly, the weight of adversarial resources is strictly less than honest resources at every point of time. Now for adversarial chain (Def. 9) $\mathcal{A}$ 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

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

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

Thus,

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

Therefore, Γ is not secure. □

Lemma 3 (Only-If-Direction of Thm. 2, Part II)**.** *Γ is not secure if $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is not homogeneous in $\boldsymbol{V}, \boldsymbol{W}$.*

Proof. Due to Def. 10, if Γ is constant then it is not secure 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 cases:

- **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

$$\Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) = \alpha\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \beta \tag{3}$$

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. (3), 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:

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

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. 10 are satisfied.

The weight of the honest chain profile is

$$\begin{aligned}
\overline{\Gamma}(\mathcal{CC}^{\mathcal{H}}) &= T_0 \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1 \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})\\
&= T_0 \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1 \cdot \alpha \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1 \cdot \beta \qquad \text{(by } \textit{Eq.} \text{ (3))}
\end{aligned}$$

$\mathcal{A}$ 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

$$\begin{aligned}
\overline{\Gamma}(\mathcal{CC}^{\mathcal{A}}) &= \int_0^{\widetilde{T}_{\mathsf{end}}} \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})\, dt = \widetilde{T}_{\mathsf{end}} \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})\\
&= \frac{T_{\mathsf{end}}}{\alpha} \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) = \frac{T_0 + T_1}{\alpha} \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})\\
&= \frac{T_0 + T_1}{\alpha} \cdot (\alpha\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \beta) \qquad \text{by } \textit{Eq.} \text{ (3)}\\
&\text{Since } T_0 \geq \frac{\alpha - 1}{\beta} \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \text{ and } T_1 = 1,\\
&\text{by simplifying, we get}\\
&\geq T_0\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + T_1(\alpha\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \beta)\\
&= \overline{\Gamma}(\mathcal{CC}^{\mathcal{H}}).
\end{aligned}$$

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

Case 1b and 1c: These cases are similar in style to Case 1a. We defer them to App. A.1.

This completes the proof. □

4 Discrete Chain Model

The continuous chain model is rather abstract, so we also consider a discretized version using blocks. A block reflects the *total* amount of resources that were expended to create it. Honest users create blocks according to some prescribed rule, e.g., in fixed time intervals, but the adversary may not adhere to this rule.

Like in the continuous model, we will describe which weight function Γ leads to a secure discrete blockchain. Compared to the continuous model the security statement introduces quantitative factors. The reason is that resources can fluctuate within a block. The quantitative parameters essentially state: The higher the magnitude of fluctuations within blocks, the larger the resource disadvantage of the adversary must be.

In principle, this statement also needs to quantitatively depend on how Γ depends on $\boldsymbol{S}$. The reason is that in our modelling a block reflects $\boldsymbol{S}$ available *at some point in time* during the block creation. Since $\boldsymbol{S}$ can fluctuate within a block, we pessimistically assume that the adversary always gets the maximum and the honest parties only the minimum. So, e.g., the function $S^2 \cdot V$ requires a larger disadvantage than $S \cdot V$. To not carry around another parameter, we restrict our attention to natural choices for Γ, namely, Γ that are subhomogeneous in $\boldsymbol{S}$ (cf. Def. 3)—think of this as "at most linear in $\boldsymbol{S}$".

4.1 Definitions

We define a blockchain $\mathcal{BC}$ as the discretization of a resource profile $\mathcal{R}$. Let us first define a *block*.

Definition 11 (Blocks)**.** Let $\mathcal{R} = (\boldsymbol{S}(t), \boldsymbol{V}(t), \boldsymbol{W}(t))_{[0,T]}$ be a resource 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

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

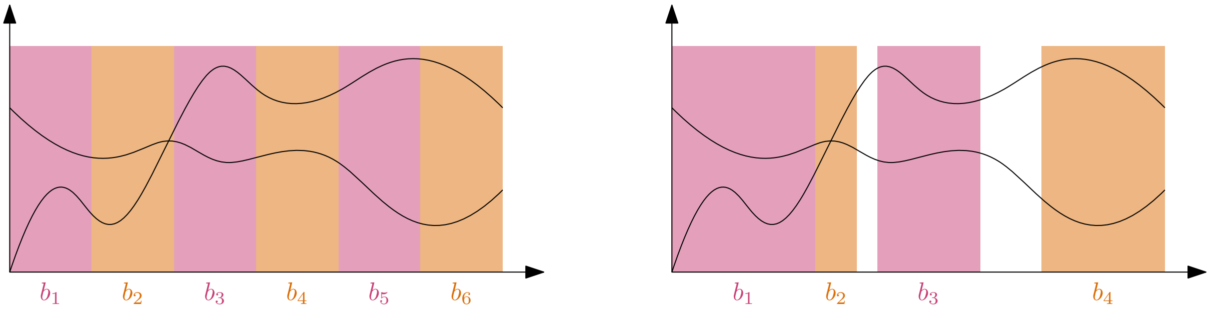
The constraint on $\mathbf{S}_{\blacksquare}$ is that

$$\inf_{t_i < t < t'_i} \boldsymbol{S}(t) \leq \mathbf{S}_{\blacksquare}(b_i) < \sup_{t_i < t < t'_i} \boldsymbol{S}(t). \tag{4}$$

The weight of a block b is $\Gamma(\mathbf{S}_{\blacksquare}(b), \mathbf{V}_{\blacksquare}(b), \mathbf{W}_{\blacksquare}(b))$.

The resources $\boldsymbol{V}$ and $\boldsymbol{W}$ accumulate within a block (e.g., a Bitcoin block reflects the expected number of hashes). As mentioned before, $\boldsymbol{S}$ is reusable, so a block can only reflect *some* amount of space that was available within the block's timespan.

In the sequel, we will often make use of minima and maxima of resources within a block. For technical reasons, they are defined via infimum and supremum, but think of them as minimum and maximum.



(a) Honest parties.

(b) Adversary (Example).

Figure 4: Discretization of parties. Here, honest parties discretize in fixed time intervals, while the adversary may construct blocks in any non-overlapping fashion.

Definition 12 (Minima and Maxima of Resources)**.** For a resource profile $\mathcal{R} = (\boldsymbol{S}(t), \boldsymbol{V}(t), \boldsymbol{W}(t))_{[0,T]}$ we denote the minima/maxima of resources in a block b with timespan (t', t'') by

$$\begin{aligned} \mathbf{S}_{\min}(b) &= \inf_{t'<t<t''} \boldsymbol{S}(t) & \mathbf{S}_{\max}(b) &= \sup_{t'<t<t''} \boldsymbol{S}(t) \\ \mathbf{V}_{\min}(b) &= \inf_{t'<t<t''} \boldsymbol{V}(t) & \mathbf{V}_{\max}(b) &= \sup_{t'<t<t''} \boldsymbol{V}(t) \\ \mathbf{W}_{\min}(b) &= \inf_{t'<t<t''} \boldsymbol{W}(t) & \mathbf{W}_{\max}(b) &= \sup_{t'<t<t''} \boldsymbol{W}(t) \end{aligned}$$

where inf, sup is applied element-wise over whole vector.

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

Definition 13 (Discrete Blockchain)**.** A discrete blockchain is a sequence of blocks $\mathcal{BC} = (b_0, \ldots b_B)$ whose timespans do not overlap. The weight of a blockchain is

$$\overline{\Gamma}_{\blacksquare}(\mathcal{BC}) = \sum_{b_i \in \mathcal{BC}} \Gamma(\mathbf{S}_{\blacksquare}(b_i), \mathbf{V}_{\blacksquare}(b_i), \mathbf{W}_{\blacksquare}(b_i)) \tag{5}$$

For honest parties, chain profile and resource profile are identical. Honest parties discretize their resource profile $\mathcal{R}^{\mathcal{H}}$ by following some prescribed rules to create blocks (e.g., in fixed one unit time intervals as depicted in Fig. 4a). The resulting blocks are non-overlapping and cover the whole timespan without gaps. The latter requirement is reasonable since honest parties generally do not waste resources. Without loss of generality, we assume the time interval to create a block is 1.

Definition 14 (Honest Discretization)**.** Let the honest parties' resource profile be $\mathcal{R}^{\mathcal{H}} = (\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t))_{[0,T]}$. Consider a blockchain $\mathcal{BC}^{\mathcal{H}} = (b_0^{\mathcal{H}}, \ldots, b_T^{\mathcal{H}})$ where each block $b_i^{\mathcal{H}}$ spans the time (t_i, t_i'). $\mathcal{BC}^{\mathcal{H}}$ is an honest blockchain arising from $\mathcal{R}^{\mathcal{H}}$ if $t_0 = 0$, $t_T' = T$, and $t_i' = i + 1$ for all $i \in [T-1]$.

The adversary also starts from a resource profile $\mathcal{R}^{\mathcal{A}}$, but they may cheat when deriving the blockchain from the resources. 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 (as shown in Fig. 4b). We capture varying timespans by continuous chain profile (Def. 9).

Definition 15 (Adversarial Discretization)**.** Let the adversary's resource profile $\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0,T]}$. Consider a blockchain $\mathcal{BC}^{\mathcal{A}} = (b_0^{\mathcal{A}}, \dots, b_B^{\mathcal{A}})$ where each block $b_i^{\mathcal{A}}$ spans the time (t_i, t_i'). $\mathcal{BC}^{\mathcal{A}}$ is an adversarial blockchain arising from $\mathcal{R}^{\mathcal{A}}$ if $0 \leq t_0$, $t_B' \leq T$ and $t_i' \leq t_{i+1}$ for all $i \in [B-1]$.

Looking ahead, the security of the discrete blockchain quantitatively depends on the maximum fluctuation of resources within blocks. We quantify this fluctuation by the ξ-*Smoothness* of resources. Essentially, $\xi \geq 1$ bounds the absolute change of resources within a block.

Definition 16 (ξ-Smoothness)**.** Let $\xi \geq 1$. A blockchain $\mathcal{BC}$ arising from $\mathcal{R}$ satisfies ε-smoothness if, for all blocks $0 \leq i \leq B$, it holds that

$$\begin{aligned} \mathbf{S}_{\max}(b_i) &\leq \xi \cdot \mathbf{S}_{\min}(b_i) \\ \mathbf{V}_{\max}(b_i) &\leq \xi \cdot \mathbf{V}_{\min}(b_i) \\ \mathbf{W}_{\max}(b_i) &\leq \xi \cdot \mathbf{W}_{\min}(b_i). \end{aligned}$$

Remark 6. In practice, blockchains generally ensure that resources within a block are relatively smooth, i.e, ξ is small. They do so by imposing an upper bound on the resources within a block. For example, Bitcoin's difficulty mechanism essentially puts an upper and lower bound on the work within a block ([KMM+21] proposes an alternative way to record work; it has no lower bound, yet still an upper bound). This effectively limits the time span a block takes. Since physical resources are not very elastic—especially at the quantities that are consumed by blockchains—, fluctuation is effectively limited.

4.2 Security Statement and Proof

Intuitively, the security statement in the continuous model was: If the adversary starts out with fewer resources than the honest parties, then the weight of the adversarial chain is lower than that of the honest one. In the discrete model, the result is a bit weaker because we require a quantitative gap, denoted by $\delta \geq 1$, between honest and adversarial resources. Together with ξ-Smoothness, this leads to the definition of (δ, ξ)-security below.

Definition 17 (Secure Weight Function for Discrete Chains)**.** A weight function Γ is (δ, ξ)-secure against private-double spending in the discrete model if, for all honest and adversarial resource profiles $\mathcal{R}^{\mathcal{H}}$ and $\mathcal{R}^{\mathcal{A}}$ such that

$$\delta \cdot \Gamma(\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t)) < \Gamma(\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t)) \; \forall t \in [0, T] \tag{6}$$

the following holds:

For any ξ-smooth (Def. 16) blockchains $\mathcal{BC}^{\mathcal{H}}$ and $\mathcal{BC}^{\mathcal{A}}$, respectively arising from $\mathcal{R}^{\mathcal{H}}$ and $\mathcal{R}^{\mathcal{A}}$ according to Defs. 14 and 15, it holds that

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

We remark that the adversary is more powerful if δ is small (i.e., the gap is small) and ξ is large (i.e., resources may fluctuate by a large magnitude). The following theorem expresses ξ as a function of δ, namely $\xi = \sqrt[4]{\delta}$. This means that if the gap δ is small, then only small fluctuations of resources within blocks may be tolerated.

Theorem 3. *For any $\delta \geq 1$, a weight function is $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is $(\delta, \sqrt[4]{\delta})$-secure against private-double spending (Def. 17) if it is*

1. *monotonically increasing;*

2. *homogeneous in $\boldsymbol{V}$ and $\boldsymbol{W}$; and*

3. *subhomogeneous in $\boldsymbol{S}$.*

Proof. Consider resource profiles $\mathcal{R}^{\mathcal{H}} = (\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t))_{[0,T]}$ and $\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0,T]}$ with $\overline{\Gamma}(\mathcal{R}^{\mathcal{H}}) > \delta \cdot \overline{\Gamma}(\mathcal{R}^{\mathcal{A}})$. Let $\xi = \sqrt[4]{\delta}$ and consider the blockchains $\mathcal{BC}^{\mathcal{H}} = (b_0^{\mathcal{H}}, \dots, b_{T-1}^{\mathcal{H}})$ and $\mathcal{BC}^{\mathcal{A}} = (b_0^{\mathcal{A}}, \dots, b_{B-1}^{\mathcal{A}})$ which arise from the resource profiles and are ξ-smooth.

We will now prove the sequence of inequalities

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

which implies the theorem since $\overline{\Gamma}(\mathcal{R}^{\mathcal{A}}) \cdot \xi^4 < \overline{\Gamma}(\mathcal{R}^{\mathcal{H}})$ due to Eq. (6). We will prove the left and right inequality separately, using one and two lemmas, respectively.

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

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

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

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

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

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. 16 to switch inf to sup, that is,

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

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

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

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

Case $\overline{\Gamma}_{\blacksquare}(\mathcal{BC}^{\mathcal{A}}) \leq \xi^2 \overline{\Gamma}(\mathcal{R}^{\mathcal{A}})$: The proof is symmetrical to the previous case. Essentially, "$\geq$" is swapped with "$\leq$" and "inf" with "sup". For completeness, this case is stated in App. A.2. □

4.3 Replotting Attacks

In the discrete model, we also have to consider *replotting attacks*. Such attacks were first discussed in the Spacemint [PKF+18] paper under the term "space reuse". The Chia green paper[19] discusses them in more detail.

Replotting attacks are easiest understood when one assumes that disk space is bound to some public key of a wallet. Then, in a replotting attack, the adversary repeatedly *replots* (i.e., re-initializes) its space using different keys within the time span of a block. This effectively increases the adversary's space within a block at the cost of extra computation to perform the replotting. Concretely, we assume replotting takes $\rho > 1$ time (usually, blockchains ensure that this ρ large), and an adversary with N space that replots m times appears to have $(m + 1) \cdot N$ space.

Remark 7. No matter the concrete cryptographic primitive used to track space, such attacks seem unavoidable in the fully-permissionless model. In other settings, e.g., the quasi-permissionless model, replotting can be disincentivized. For example, Filecoin [Fil24] requires parties to commit to space, and then the parties must continuously prove that they are storing the committed space. This prevents replotting if the gap between the required proofs is smaller than the replotting time.

Extra Assumptions are Necessary. Without any extra assumptions, replotting leads to attacks in the discrete model. For the sake of example, consider Chia's weight function $S \cdot V$, which is secure according to Thm. 3. Assume replotting takes time $\rho = 2$, $S^{\mathcal{A}} = V^{\mathcal{A}} = 1$ and $S^{\mathcal{H}} = V^{\mathcal{H}} = 1.1$ (this gap suffices since both profiles are 1-smooth), and consider the timespan $[0, 6]$. The honest parties create 5 blocks with a cumulative weight of $6 \cdot (1.1 \cdot 1.1) \approx 7.3$. The adversary creates one block in which it replots once. Assuming that the adversary cannot do anything else while replotting (i.e., it can only gain V for 4 time), the weight of the block is $4 \cdot (2 \cdot 1) = 8$. Note that this attack generalizes to other weight functions Γ and other values of ρ.

[19] https://docs.chia.net/green-paper-abstract/

A Solution using Difficulty. One way to disincentive replotting in the discrete model is bounding the total weight of a block b. Consider that the protocol keeps track of a difficulty D that is periodically adjusted so that roughly one block is created per time unit (e.g., in Bitcoin the difficulty is reset once every two weeks so blocks arrive roughly every 10 minutes). Further, let $\eta \geq 1$ be a protocol parameter (to be set later). Then, the protocol bounds the weight of blocks as $D \leq \Gamma(b) \leq \eta \cdot D$ (abusing notation of Γ slightly).

If we now set $\eta < \rho$, it is not hard to see that replotting does not help: Replotting even once requires ρ time, and the resulting adversarial block has at most $\eta \cdot D < \rho \cdot D$ weight. Meanwhile, the honest parties produce around ρ blocks, each of weight at least D; so in total $\rho \cdot D$.

Note that this argument requires D to stay fixed, an attacker might still be able to create a heavier chain over a long period of time that spans several epochs (where the difficulty is reset once every epoch).

Chia [chi19] with its weight function $\Gamma(S, V) = S \cdot V$ uses a similar idea, but it tracks the difficulty of the space and VDF separately. The block arrival time is only determined by the VDF difficulty, which is nice as VDF speed hardly fluctuates at all over time.

One can generalize this idea to any weight function that can be written as $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W}) = \Gamma_1(\boldsymbol{S}) \cdot \Gamma_2(\boldsymbol{V}, \boldsymbol{W})$. Now one would require that each block that records resources $\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{s}$ must satisfy $\Gamma_2(\boldsymbol{v}, \boldsymbol{w}) = D$. One doesn't need to put an additional upper bound on the space $\Gamma_1(\boldsymbol{s})$ if Γ_1 is subhomogenous, i.e., for any $\alpha > 1$ we have $\Gamma_1(\alpha\boldsymbol{s}) \leq \alpha\Gamma_1(\boldsymbol{s})$ (Chia does both, it is (sub)homogenous and has an upper bound).

Future Work on Replotting. As mentioned, the above solutions don't formally prevent replotting attacks that range over many epochs. In practice that might not be such a big issue, as extremely long range attacks are not really practical: they require a lot of resources for a long period of time, and thus are expensive to launch. Moreover it might be difficult to convince honest parties to accept a very long fork as it's such an obvious attack. It still would be interesting to understand whether it's possible to formally achieve security against replotting attacks, we leave this for future work.

References

[ACL+23] Sarah Azouvi, Christian Cachin, Duc V. Le, Marko Vukolić, and Luca Zanolini. Modeling Resources in Permissionless Longest-Chain Total-Order Broadcast. In Eshcar Hillel, Roberto Palmieri, and Etienne Rivière, editors, *26th International Conference on Principles of Distributed Systems (OPODIS 2022)*, volume 253 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:23, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[ADH+23] Steven Allen, Masih H. Derkani, Jie Hou, Henrique Moniz, Alex North, Matej Pavlovic, Aayush Rajasekaran, Alejandro Ranchal-Pedrosa, Jorge M. Soares, Jakub Sztandera, Marko Vukolic, and Jennifer Wang. Fast Finality in Filecoin (F3). https://github.com/filecoin-project/FIPs/blob/master/FIPS/fip-0086.md, 2023.

[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, Cham, 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.

[BGK+18a] Christian Badertscher, Peter Gazi, Aggelos Kiayias, Alexander Russell, and Vassilis Zikas. Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 913–930. ACM Press, October 2018.

[BGK+18b] Christian Badertscher, Peter Gazi, Aggelos Kiayias, Alexander Russell, and Vassilis Zikas. Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 913–930. ACM, 2018.

[BLMR14] Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract]y. *SIGMETRICS Perform. Eval. Rev.*, 42(3):34–37, December 2014.

[chi19] The chia network blockchain. https://docs.chia.net/green-paper-abstract/, 2019.

[CP19] Bram Cohen and Krzysztof Pietrzak. The chia network blockchain. https://docs.chia.net/files/Precursor-ChiaGreenPaper.pdf, 2019. This is an early proposal and differs significantly from the implemented version [chi19].

[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, Berlin, Heidelberg, August 2015.

[DGKR18] Bernardo David, Peter Gazi, Aggelos Kiayias, and Alexander Russell. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 66–98. Springer, Cham, April / May 2018.

[DKT+20] Amir Dembo, Sreeram Kannan, Ertem Nusret Tas, David Tse, Pramod Viswanath, Xuechao Wang, and Ofer Zeitouni. Everything is a race and nakamoto always wins. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 859–878. ACM Press, November 2020.

[DKT21] Soubhik Deb, Sreeram Kannan, and David Tse. PoSAT: Proof-of-work availability and unpredictability, without the work. In Nikita Borisov and Claudia Díaz, editors, *FC 2021, Part II*, volume 12675 of *LNCS*, pages 104–128. Springer, Berlin, Heidelberg, March 2021.

[DPS19] Phil Daian, Rafael Pass, and Elaine Shi. Snow white: Robustly reconfigurable consensus and applications to provably secure proof of stake. In Ian Goldberg and Tyler Moore, editors, *FC 2019*, volume 11598 of *LNCS*, pages 23–41. Springer, Cham, February 2019.

[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, Berlin, Heidelberg, March 2014.

[Fil24] Filecoin. Filecoin. https://filecoin.io, 2024.

[FWK$^+$22] Matthias Fitzi, Xuechao Wang, Sreeram Kannan, Aggelos Kiayias, Nikos Leonardos, Pramod Viswanath, and Gerui Wang. Minotaur: Multi-resource blockchain consensus. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 1095–1108, New York, NY, USA, 2022. Association for Computing Machinery.

[GHM$^+$17] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, page 51–68, New York, NY, USA, 2017. Association for Computing Machinery.

[GKL15] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 281–310. Springer, Berlin, Heidelberg, April 2015.

[GKR20] Peter Gazi, Aggelos Kiayias, and Alexander Russell. Tight consistency bounds for bitcoin. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 819–838. ACM Press, November 2020.

[KMM$^+$21] Simon Holmgaard Kamp, Bernardo Magri, Christian Matt, Jesper Buus Nielsen, Søren Eller Thomsen, and Daniel Tschudi. Weight-based nakamoto-style blockchains. In Patrick Longa and Carla Ràfols, editors, *Progress in Cryptology – LATINCRYPT 2021*, pages 299–319, Cham, 2021. Springer International Publishing.

[KN12] Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. 2012.

[KRDO17] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 357–388. Springer, Cham, August 2017.

[LPR23] Andrew Lewis-Pye and Tim Roughgarden. Byzantine generals in the permissionless setting. In Foteini Baldimtsi and Christian Cachin, editors, *FC 2023, Part I*, volume 13950 of *LNCS*, pages 21–37. Springer, Cham, May 2023.

[LPR24] Andrew Lewis-Pye and Tim Roughgarden. Permissionless consensus. https://arxiv.org/abs/2304.14701, 2024.

[Nak09] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf, 2009.

[Pie19] Krzysztof Pietrzak. Simple verifiable delay functions. In Avrim Blum, editor, *ITCS 2019*, volume 124, pages 60:1–60:15. LIPIcs, January 2019.

[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, Berlin, Heidelberg, February / March 2018.

[PS17] Rafael Pass and Elaine Shi. Rethinking large-scale consensus. In Boris Köpf and Steve Chong, editors, *CSF 2017 Computer Security Foundations Symposium*, pages 115–129. IEEE Computer Society Press, 2017.

[PSs17] Rafael Pass, Lior Seeman, and abhi shelat. Analysis of the blockchain protocol in asynchronous networks. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 643–673. Springer, Cham, April / May 2017.

[Ren19] Ling Ren. Analysis of Nakamoto consensus. Cryptology ePrint Archive, Report 2019/943, 2019.

[SZ13] Yonatan Sompolinsky and Aviv Zohar. Accelerating Bitcoin's transaction processing. Fast money grows on trees, not chains. Cryptology ePrint Archive, Report 2013/881, 2013.

[Ter22] Benjamin Terner. Permissionless consensus in the resource model. In Ittay Eyal and Juan Garay, editors, *Financial Cryptography and Data Security*, pages 577–593, Cham, 2022. Springer International Publishing.

[TSZ23] Apostolos Tzinas, Srivatsan Sridhar, and Dionysis Zindros. On-chain timestamps are accurate. Cryptology ePrint Archive, Report 2023/1648, 2023.

[Wes19] Benjamin Wesolowski. Efficient verifiable delay functions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 379–407. Springer, Cham, May 2019.

[Woo] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger.

A Missing Parts of Proofs

A.1 Rest of Lem. 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. 10 are met.

The weight of the honest chain profile is

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

$\mathcal{A}$ 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

$$\begin{aligned}
\overline{\Gamma}(\mathcal{CC}^{\mathcal{A}}) &= \int_0^{\mathsf{AT}(T_0)} \Gamma(\widetilde{\boldsymbol{S}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{V}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{W}}^{\mathcal{A}}(t))\, dt \\
&\quad + \int_{\mathsf{AT}(T_0)}^{\widetilde{T}_{\mathsf{end}}} \Gamma(\widetilde{\boldsymbol{S}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{V}}^{\mathcal{A}}(t), \widetilde{\boldsymbol{W}}^{\mathcal{A}}(t))\, dt \\
&= \int_0^{\frac{T_0}{\alpha}} \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})\, dt \\
&\quad + \int_{\frac{T_0}{\alpha}}^{\frac{T_0}{\alpha}+T_1} \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w})\, dt \\
&= \frac{T_0}{\alpha} \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) + T_1 \cdot \Gamma(\boldsymbol{s}, \alpha\boldsymbol{v}, \alpha\boldsymbol{w}) \\
&= \left(\frac{T_0}{\alpha} + T_1\right) (\alpha\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) + \beta) && \text{by } \textit{Eq. } (3) \\
&\text{Since } T_0 \geq \frac{\alpha}{\beta}((1-\alpha)\Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) - \beta) \text{ and } T_1 = 1, \\
&\text{by simplifying, we get} \\
&\geq (T_0 + T_1) \cdot \Gamma(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{w}) \\
&= \overline{\Gamma}(\mathcal{CC}^{\mathcal{H}}).
\end{aligned}$$

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

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

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. 10 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:

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

Case A: :

$$\begin{array}{llll}
\boldsymbol{S}^{\mathcal{H}}(t) = \boldsymbol{s}', & \boldsymbol{V}^{\mathcal{H}}(t) = \boldsymbol{v}', & \boldsymbol{W}^{\mathcal{H}}(t) = \boldsymbol{w}' & \text{for } t \in [T_0, T_{\mathsf{end}}] \\
\boldsymbol{S}^{\mathcal{A}}(t) = \boldsymbol{s}, & \boldsymbol{V}^{\mathcal{A}}(t) = \boldsymbol{v}, & \boldsymbol{W}^{\mathcal{A}}(t) = \boldsymbol{w} & \text{for } t \in [T_0, T_{\mathsf{end}}]
\end{array}$$

Case B: :

$$\begin{array}{llll}
\boldsymbol{S}^{\mathcal{H}}(t) = \boldsymbol{s}, & \boldsymbol{V}^{\mathcal{H}}(t) = \boldsymbol{v}, & \boldsymbol{W}^{\mathcal{H}}(t) = \boldsymbol{w} & \text{for } t \in [T_0, T_{\mathsf{end}}] \\
\boldsymbol{S}^{\mathcal{A}}(t) = \boldsymbol{s}', & \boldsymbol{V}^{\mathcal{A}}(t) = \boldsymbol{v}', & \boldsymbol{W}^{\mathcal{A}}(t) = \boldsymbol{w}' & \text{for } t \in [T_0, T_{\mathsf{end}}]
\end{array}$$

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. 10.

The weight of the honest chain profile is:

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

which, by definition of δ, is same as:

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

$\mathcal{A}$ 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

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

For **Case A:**,

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

For **Case B:**,

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

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

A.2 Rest of Thm. 3

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

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

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

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

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

where the last line follows from monotonicity.

Now we invoke Def. 16 to switch max to min, that is,

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

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

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

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