Nakamoto Consensus from Multiple Resources

Mirza Ahad Baig   Christoph U. Günther   Krzysztof Pietrzak

Institute of Science and Technology Austria

Main Result

Our Result:

We characterize the design space of Nakamoto-like protocols operating in the fully-permissionless setting using physical resources that are secure against double-spending attacks in an idealized model.

Background

Degree of Permissionlessness [1]

[1] "Permissionless Consensus" (Lewis-Pye and Roughgarden 2024)

Degree of Permissionlessness [1]

1. Fully permissionless - Protocol does not know current participation, e.g., Bitcoin.

[1] "Permissionless Consensus" (Lewis-Pye and Roughgarden 2024)

Degree of Permissionlessness [1]

1. Fully permissionless - Protocol does not know current participation, e.g., Bitcoin.
2. Dynamically available - Participation from a subset of a dynamically evolving list of IDs, e.g., Ouroboros Genesis.

[1] "Permissionless Consensus" (Lewis-Pye and Roughgarden 2024)

Degree of Permissionlessness [1]

1. Fully permissionless - Protocol does not know current participation, e.g., Bitcoin.
2. Dynamically available - Participation from a subset of a dynamically evolving list of IDs, e.g., Ouroboros Genesis.
3. Quasi-permissionless - Participation of all the IDs in a dynamically evolving list above, e.g., Algorand.

[1] "Permissionless Consensus" (Lewis-Pye and Roughgarden 2024)

Degree of Permissionlessness [1]

1. Fully permissionless - Protocol does not know current participation, e.g., Bitcoin.
2. Dynamically available - Participation from a subset of a dynamically evolving list of IDs, e.g., Ouroboros Genesis.
3. Quasi-permissionless - Participation of all the IDs in a dynamically evolving list above, e.g., Algorand.
4. Permissioned - e.g., Tendermint.

[1] "Permissionless Consensus" (Lewis-Pye and Roughgarden 2024)

Bitcoin [3]

[2] "Analysis of Nakamoto Consensus"; "Everything is a Race and Nakamoto Always Wins" (Ren 2019; Dembo et al. 2020)

[3] "Bitcoin: A Peer-to-Peer Electronic Cash System" (Nakamoto 2009)

Bitcoin [3]

Bitcoin is heaviest-chain protocol that operates in a fully permissionless setting.

[2] "Analysis of Nakamoto Consensus"; "Everything is a Race and Nakamoto Always Wins" (Ren 2019; Dembo et al. 2020)

[3] "Bitcoin: A Peer-to-Peer Electronic Cash System" (Nakamoto 2009)

Bitcoin [3]

Bitcoin is heaviest-chain protocol that operates in a fully permissionless setting.

Heaviest-chain Selection Rule: Choose the chain with highest cumulative difficulty.

[2] "Analysis of Nakamoto Consensus"; "Everything is a Race and Nakamoto Always Wins" (Ren 2019; Dembo et al. 2020)

[3] "Bitcoin: A Peer-to-Peer Electronic Cash System" (Nakamoto 2009)

Bitcoin [3]

Bitcoin is heaviest-chain protocol that operates in a fully permissionless setting.

Heaviest-chain Selection Rule: Choose the chain with highest cumulative difficulty.

Secure under "honest-majority": At any point of time

$$\textbf{Honest PoW} > \textbf{Adversarial PoW}$$

[2] "Analysis of Nakamoto Consensus"; "Everything is a Race and Nakamoto Always Wins" (Ren 2019; Dembo et al. 2020)

[3] "Bitcoin: A Peer-to-Peer Electronic Cash System" (Nakamoto 2009)

Bitcoin [3]

Bitcoin is heaviest-chain protocol that operates in a fully permissionless setting.

Heaviest-chain Selection Rule: Choose the chain with highest cumulative difficulty.

Secure under "honest-majority": At any point of time

$$\textbf{Honest PoW} > \textbf{Adversarial PoW}$$

Taking network delays into account [2]

$$\textbf{Honest PoW} > \chi(\Delta) \cdot \textbf{Adversarial PoW}$$

[2] "Analysis of Nakamoto Consensus"; "Everything is a Race and Nakamoto Always Wins" (Ren 2019; Dembo et al. 2020)

[3] "Bitcoin: A Peer-to-Peer Electronic Cash System" (Nakamoto 2009)

Chia [4]

[4] "The Chia Network Blockchain" (Cohen and Pietrzak 2019)

Chia [4]

Operates in fully permissionless setting.

[4] "The Chia Network Blockchain" (Cohen and Pietrzak 2019)

Chia [4]

Operates in fully permissionless setting.

Chia is a heaviest-chain protocol that operates in fully-permissionless setting.

[4] "The Chia Network Blockchain" (Cohen and Pietrzak 2019)

Chia [4]

Operates in fully permissionless setting.

Chia is a heaviest-chain protocol that operates in fully-permissionless setting.

Uses Proof-of-Space and Verifiable Delay Functions (VDF).

[4] "The Chia Network Blockchain" (Cohen and Pietrzak 2019)

Chia [4]

Operates in fully permissionless setting.

Chia is a heaviest-chain protocol that operates in fully-permissionless setting.

Uses Proof-of-Space and Verifiable Delay Functions (VDF).

Secure under "honest-majority": At any point of time

$$\textbf{Honest Space} \cdot \textbf{VDF Speed} > \textbf{Adversarial Space} \cdot \textbf{VDF Speed}$$

[4] "The Chia Network Blockchain" (Cohen and Pietrzak 2019)

Nakamoto-Like Protocols

Nakamoto-Like Protocols

- To produce a block parties need to solve a challenge which arrive periodically.

Nakamoto-Like Protocols

- To produce a block parties need to solve a challenge which arrive periodically.
- Each block B_i indicates how much resource went into producing it (upto an approximation).

Nakamoto-Like Protocols

- To produce a block parties need to solve a challenge which arrive periodically.
- Each block B_i indicates how much resource went into producing it (upto an approximation).
- We consider three available physical resources space (POS), S, sequential work (VDF), V, and parallel work (PoW), W. There may be multiple versions of resources, say two different PoW W_1, W_2.

Nakamoto-Like Protocols

- To produce a block parties need to solve a challenge which arrive periodically.
- Each block B_i indicates how much resource went into producing it (upto an approximation).
- We consider three available physical resources space (POS), S, sequential work (VDF), V, and parallel work (PoW), W. There may be multiple versions of resources, say two different PoW W_1, W_2.
- V, W are timed resources.

Nakamoto-Like Protocols

- To produce a block parties need to solve a challenge which arrive periodically.
- Each block B_i indicates how much resource went into producing it (upto an approximation).
- We consider three available physical resources space (POS), S, sequential work (VDF), V, and parallel work (PoW), W. There may be multiple versions of resources, say two different PoW W_1, W_2.
- V, W are timed resources.
- Honestly generated blocks accurately reflect the resource honest parties had for the duration since latest challenge arrived.

Nakamoto-Like Protocols

- To produce a block parties need to solve a challenge which arrive periodically.
- Each block B_i indicates how much resource went into producing it (upto an approximation).
- We consider three available physical resources space (POS), S, sequential work (VDF), V, and parallel work (PoW), W. There may be multiple versions of resources, say two different PoW W_1, W_2.
- V, W are timed resources.
- Honestly generated blocks accurately reflect the resource honest parties had for the duration since latest challenge arrived.
- An adversary may violate these and misrepresent how much resource by spending more or less time on solving a challenge.

Nakamoto-Like Protocols

- To produce a block parties need to solve a challenge which arrive periodically.
- Each block B_i indicates how much resource went into producing it (upto an approximation).
- We consider three available physical resources space (POS), S, sequential work (VDF), V, and parallel work (PoW), W. There may be multiple versions of resources, say two different PoW W_1, W_2.
- V, W are timed resources.
- Honestly generated blocks accurately reflect the resource honest parties had for the duration since latest challenge arrived.
- An adversary may violate these and misrepresent how much resource by spending more or less time on solving a challenge.
- Chain Selection Rule: Choose the chain with higher

$$\sum_i (\Gamma(\text{Resource}(B_i)))$$

Question

Question

Question

Which weight functions, Γ, are secure?.

Question

Question

Which weight functions, Γ, are secure?.

Examples: W and SV are secure weight functions.

Question

Question

Which weight functions, Γ, are secure?.

Examples: W and SV are secure weight functions.

We first study the question in an idealized model and then make it more realistic.

Continuous Model

Resource Profile

Resource Profile

- We model time as continuous.

Resource Profile

- We model time as continuous.
- Operating under the maxim:

  "Ideal chain reflects exactly at each point of time the amount of resource that went into producing it."

Resource Profile

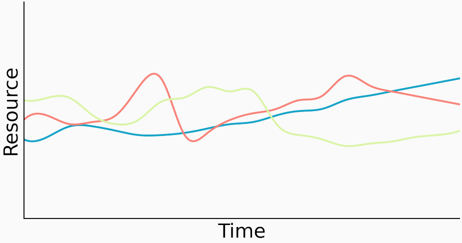
- We model time as continuous.
- Operating under the maxim:

  "Ideal chain reflects exactly at each point of time the amount of resource that went into producing it."

- **Resource Profile**: Resources which are available at any point of time.

$$S : [0, T] \to \mathbb{R}_{>0} \quad V : [0, T] \to \mathbb{R}_{>0} \quad W : [0, T] \to \mathbb{R}_{>0}$$

Collectively, $\mathcal{R} = (S, V, W)_{[0,T]}$. Honest resources are $\mathcal{R}^{\mathcal{H}}$ and adversarial resources are $\mathcal{R}^{\mathcal{A}}$.



Chain Profile

Chain Profile

- **Continuous Chain Profile**: From a given resource profile the parties create a chain which is represented as

$$\mathcal{CC} = (S, V, W)_{[0,T]}.$$

Chain Profile

- **Continuous Chain Profile**: From a given resource profile the parties create a chain which is represented as

$$\mathcal{CC} = (S, V, W)_{[0,T]}.$$

For honest parties chain profile accurately represents their resource:

$$\mathcal{CC}^{\mathcal{H}} = \mathcal{R}^{\mathcal{H}}$$

What can adversary do?

What can adversary do?

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

What can adversary do?

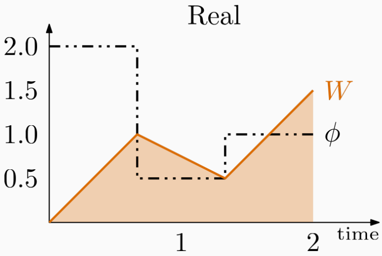
$$\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0, T_{\text{end}}]} \longrightarrow \mathcal{CC}^{\mathcal{A}} = (\widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}), \widetilde{\boldsymbol{V}}^{\mathcal{A}}(\widetilde{t}),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

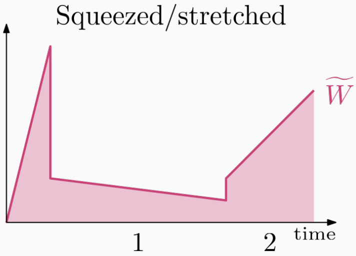
Intuition: Adversary can trade-off time with resource. It can wait longer to put more resource into the chain at one point and make it appear as if it had more by manipulating the timestamps.

What can adversary do?

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

Intuition: Adversary can trade-off time with resource. It can wait longer to put more resource into the chain at one point and make it appear as if it had more by manipulating the timestamps.





What can adversary do?

Adversarial Resource Profile $\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0, T_{\text{end}}]}$

What can adversary do?

Adversarial Resource Profile $\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0, T_{\text{end}}]}$

Choose a function $\phi(t)\colon [0, T_{\text{end}}] \to \mathbb{R}_{>0}$

What can adversary do?

Adversarial Resource Profile $\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0, T_{\text{end}}]}$

Choose a function $\phi(t) \colon [0, T_{\text{end}}] \to \mathbb{R}_{>0}$

Adversarial chain profile

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

such that

What can adversary do?

Adversarial Resource Profile $\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0, T_{\text{end}}]}$

Choose a function $\phi(t)\colon [0, T_{\text{end}}] \to \mathbb{R}_{>0}$

Adversarial chain profile

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

such that

$$0 < \widetilde{\boldsymbol{S}}^{\mathcal{A}}(\widetilde{t}) \leq \boldsymbol{S}^{\mathcal{A}}(t)$$

for all $\widetilde{t} \in [0, \widetilde{T}_{\text{end}}]$

What can adversary do?

Adversarial Resource Profile $\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0, T_{\text{end}}]}$

Choose a function $\phi(t)\colon [0, T_{\text{end}}] \to \mathbb{R}_{>0}$

Adversarial chain profile

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

such that

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

for all $\widetilde{t} \in [0, \widetilde{T}_{\text{end}}]$

Weight Function

Weight Function

Weight function is a non-constant function given by

$$\Gamma\colon \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0} \to \mathbb{R}_{>0}$$

Weight Function

Weight function is a non-constant function given by

$$\Gamma \colon \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0} \to \mathbb{R}_{>0}$$

Weight of a resource profile $\mathcal{R} = (\boldsymbol{S}'(t), \boldsymbol{V}'(t), \boldsymbol{W}'(t))_{[0,T]}$

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

Weight Function

Weight function is a non-constant function given by

$$\Gamma \colon \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0} \to \mathbb{R}_{>0}$$

Weight of a resource profile $\mathcal{R} = (\boldsymbol{S}'(t), \boldsymbol{V}'(t), \boldsymbol{W}'(t))_{[0,T]}$

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

Weight of a chain profile $\mathcal{CC} = (\boldsymbol{S}(t), \boldsymbol{V}(t), \boldsymbol{W}(t))_{[0,T]}$

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

Secure Weight Function

Secure Weight Function

A weight function Γ is *secure* in the *continuous model*

Secure Weight Function

A weight function Γ is *secure* in the *continuous model*

if for all $\mathcal{R}^{\mathcal{H}} = (\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t))_{[0, T_{\text{end}}]}$ and $\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0, T_{\text{end}}]}$ such that

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

Secure Weight Function

A weight function Γ is *secure* in the *continuous model*

if for all $\mathcal{R}^{\mathcal{H}} = (\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t))_{[0, T_{\text{end}}]}$ and $\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0, T_{\text{end}}]}$ such that

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

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

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

Secure Weight Function

A weight function Γ is *secure* in the *continuous model*

if for all $\mathcal{R}^{\mathcal{H}} = (\boldsymbol{S}^{\mathcal{H}}(t), \boldsymbol{V}^{\mathcal{H}}(t), \boldsymbol{W}^{\mathcal{H}}(t))_{[0, T_{\text{end}}]}$ and
$\mathcal{R}^{\mathcal{A}} = (\boldsymbol{S}^{\mathcal{A}}(t), \boldsymbol{V}^{\mathcal{A}}(t), \boldsymbol{W}^{\mathcal{A}}(t))_{[0, T_{\text{end}}]}$ such that

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

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

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

it holds that

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

where $\mathcal{CC}^{\mathcal{H}} := \mathcal{R}^{\mathcal{H}}$ and $\mathcal{CC}^{\mathcal{A}}$ is created from $\mathcal{R}^{\mathcal{A}}$ using some $\phi(t)$.

Main Result in Continuous Model

Main Result in Continuous Model

Theorem

A weight function Γ is secure if and only if $\Gamma(S, V, W)$ is monotonically increasing and homogeneous in V, W.

Main Result in Continuous Model

Theorem

A weight function Γ is secure if and only if $\Gamma(S, V, W)$ is monotonically increasing and homogeneous in V, W.

A function $f(x, y, z)$ is homogeneous in y, z if for all $\alpha > 0$

$$f(x, \alpha y, \alpha z) = \alpha f(x, y, z)$$

Main Result in Continuous Model

Theorem

A weight function Γ is secure if and only if $\Gamma(S, V, W)$ is monotonically increasing and homogeneous in V, W.

A function $f(x, y, z)$ is homogeneous in y, z if for all $\alpha > 0$

$$f(x, \alpha y, \alpha z) = \alpha f(x, y, z)$$

V, W are *timed* resources, while S is not a timed resource.

Main Result in Continuous Model

Theorem

A weight function Γ is secure if and only if $\Gamma(S, V, W)$ is monotonically increasing and homogeneous in V, W.

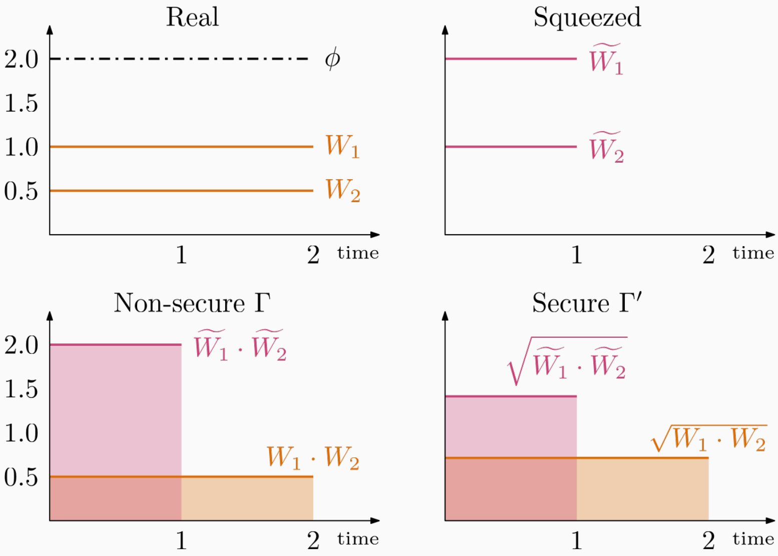
A function $f(x, y, z)$ is homogeneous in y, z if for all $\alpha > 0$

$$f(x, \alpha y, \alpha z) = \alpha f(x, y, z)$$

V, W are *timed* resources, while S is not a timed resource.

Physics intuition: we want the units to be *per second*.

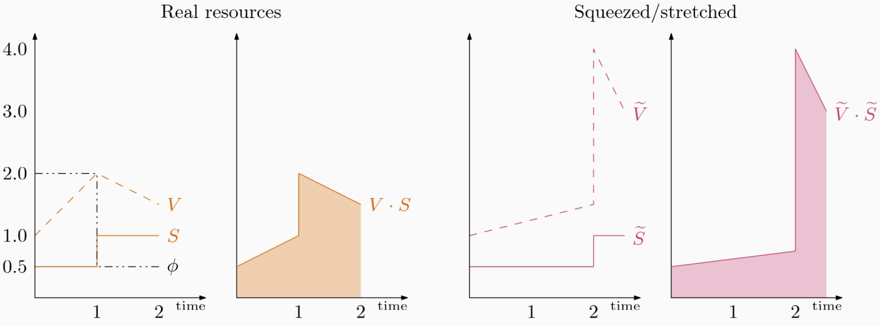
Examples



Real
2.0
1.5
1.0
0.5
ϕ
W_1
W_2
1
2
time
Squeezed
$\widetilde{W}_1$
$\widetilde{W}_2$
1
2
time
Non-secure Γ
2.0
1.5
1.0
0.5
$\widetilde{W}_1 \cdot \widetilde{W}_2$
$W_1 \cdot W_2$
1
2
time
Secure Γ'
$\sqrt{\widetilde{W}_1 \cdot \widetilde{W}_2}$
$\sqrt{W_1 \cdot W_2}$
1
2
time

Examples

Chia



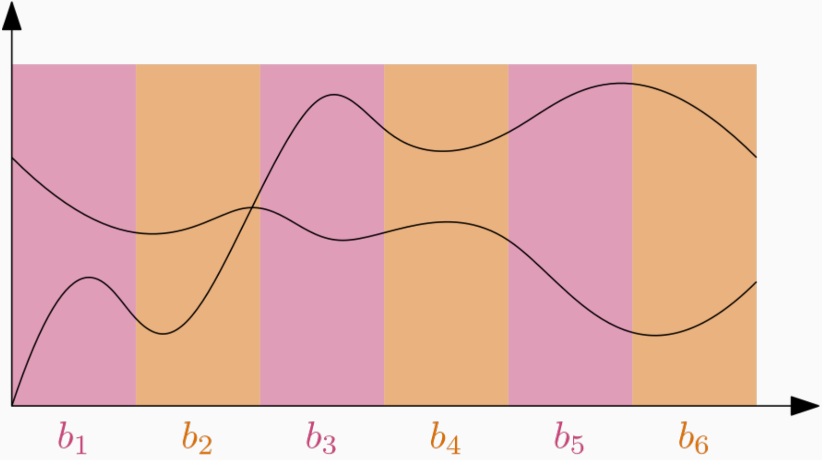
Discrete Model

Discrete Model

Towards a more realistic model.

Discrete Model

Towards a more realistic model.



Discrete Model

Discrete Model

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

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

Discrete Model

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

$$\inf_{t_i < t < t_i'} \boldsymbol{S}(t) \leq \mathbf{S}_\blacksquare(b_i) < \sup_{t_i < t < t_i'} \boldsymbol{S}(t). \tag{1}$$

Discrete Model

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

$$\inf_{t_i<t<t_i'} \boldsymbol{S}(t) \leq \mathbf{S}_{\blacksquare}(b_i) < \sup_{t_i<t<t_i'} \boldsymbol{S}(t). \tag{1}$$

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

Discrete Model

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

$$\inf_{t_i < t < t_i'} \boldsymbol{S}(t) \leq \mathbf{S}_{\blacksquare}(b_i) < \sup_{t_i < t < t_i'} \boldsymbol{S}(t). \tag{1}$$

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

A discrete blockchain $\mathcal{BC} = (b_0, \ldots b_B)$

Discrete Model

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

$$\inf_{t_i < t < t_i'} \boldsymbol{S}(t) \leq \mathbf{S}_{\blacksquare}(b_i) < \sup_{t_i < t < t_i'} \boldsymbol{S}(t). \tag{1}$$

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

A discrete blockchain $\mathcal{BC} = (b_0, \ldots b_B)$

The weight of a blockchain is

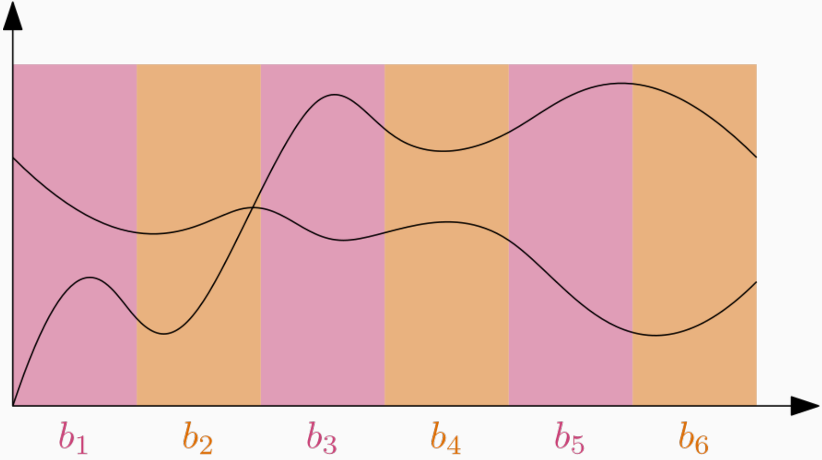
$$\overline{\Gamma}_{\blacksquare}(\mathcal{BC}) = \sum_{b_i \in \mathcal{BC}} \Gamma(\mathbf{S}_{\blacksquare}(b_i), \mathbf{V}_{\blacksquare}(b_i), \mathbf{W}_{\blacksquare}(b_i)) \tag{2}$$

What can an adversary do?

Honest parties choose uniform timestamps.

What can an adversary do?

Honest parties choose uniform timestamps.

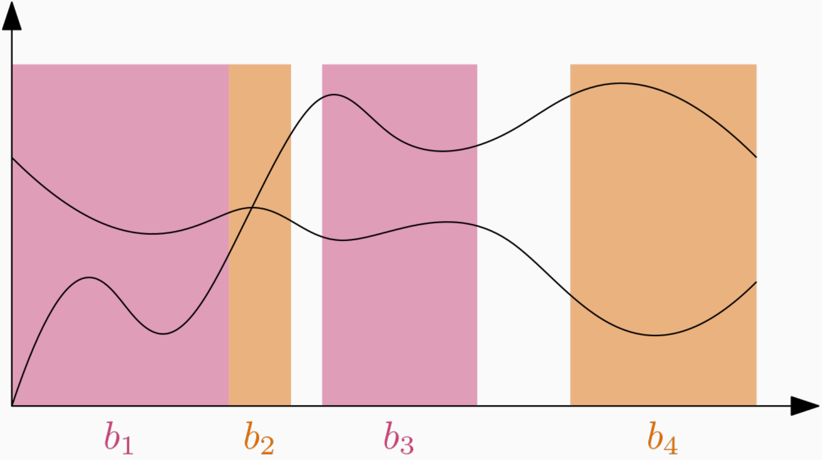


What can an adversary do?

Adversary may choose to deviate.

What can an adversary do?

Adversary may choose to deviate.



Smoothness

Adversary is now too powerful, so we need to bring additional restrictions. A natural restrictions:

Smoothness

Adversary is now too powerful, so we need to bring additional restrictions. A natural restrictions:

ξ-**smoothness**: ($\xi \geq 1$)
A blockchain $\mathcal{BC}$ created from $\mathcal{R}$ is ξ-smooth if, for all blocks b_i

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

Smoothness

Adversary is now too powerful, so we need to bring additional restrictions. A natural restrictions:

ξ-**smoothness**: ($\xi \geq 1$)
A blockchain $\mathcal{BC}$ created from $\mathcal{R}$ is ξ-smooth if, for all blocks b_i

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

This is akin to Bitcoin not allowing difficulty to change by more than a factor 4.

Smoothness

Adversary is now too powerful, so we need to bring additional restrictions. A natural restrictions:

ξ-**smoothness**: ($\xi \geq 1$)
A blockchain $\mathcal{BC}$ created from $\mathcal{R}$ is ξ-smooth if, for all blocks b_i

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

This is akin to Bitcoin not allowing difficulty to change by more than a factor 4.

This can be achieved by restricting how much total resources can go into a single block.

Security in Discrete Model

Security in Discrete Model

We additionally need to increase the gap between adversarial resources and the honest resources.

Security in Discrete Model

We additionally need to increase the gap between adversarial resources and the honest resources.

Security: A weight function Γ is (δ, ξ)-secure in the discrete model if, for any resource profiles $\mathcal{R}^{\mathcal{H}}$ and $\mathcal{R}^{\mathcal{A}}$ such that

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

the following is true:

Security in Discrete Model

We additionally need to increase the gap between adversarial resources and the honest resources.

Security: A weight function Γ is (δ, ξ)-secure in the discrete model if, for any resource profiles $\mathcal{R}^{\mathcal{H}}$ and $\mathcal{R}^{\mathcal{A}}$ such that

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

the following is true:

For any ξ-smooth blockchains $\mathcal{BC}^{\mathcal{H}}$ and $\mathcal{BC}^{\mathcal{A}}$, created from $\mathcal{R}^{\mathcal{H}}$ and $\mathcal{R}^{\mathcal{A}}$ respectively,

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

Main Result in Discrete Model

Main Result in Discrete Model

Theorem

For any $\delta \geq 1$, a weight function is $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is $(\delta, \sqrt[4]{\delta})$-secure if it is

1. monotonically increasing;
2. homogeneous in $\boldsymbol{V}$ and $\boldsymbol{W}$; and
3. sub-homogeneous in $\boldsymbol{S}$.

Main Result in Discrete Model

Theorem

For any $\delta \geq 1$, a weight function is $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is $(\delta, \sqrt[4]{\delta})$-secure if it is

1. monotonically increasing;
2. homogeneous in $\boldsymbol{V}$ and $\boldsymbol{W}$; and
3. sub-homogeneous in $\boldsymbol{S}$.

A function $f(x, y, z)$ is sub-homogeneous in x if for all $\alpha > 0$

$$f(\alpha x, y, z) \leq \alpha f(x, y, z)$$

Main Result in Discrete Model

Theorem

For any $\delta \geq 1$, a weight function is $\Gamma(\boldsymbol{S}, \boldsymbol{V}, \boldsymbol{W})$ is $(\delta, \sqrt[4]{\delta})$-secure if it is

1. monotonically increasing;
2. homogeneous in $\boldsymbol{V}$ and $\boldsymbol{W}$; and
3. sub-homogeneous in $\boldsymbol{S}$.

A function $f(x, y, z)$ is sub-homogeneous in x if for all $\alpha > 0$

$$f(\alpha x, y, z) \leq \alpha f(x, y, z)$$

Intuitively sub-homogeneity in S is required because we allowed adversary to pick max space while honest parties get the smaller one. Additionally we have a problem of replotting in space.

Discussion

Replotting

Replotting

Replotting attacks are inherent to $Proof - of - Space$ in a fully permissionless and dynamically available settings. The adversary re-initiates its space using a different key to make it appear as if it has more space.

Replotting

Replotting attacks are inherent to $Proof - of - Space$ in a fully permissionless and dynamically available settings. The adversary re-initiates its space using a different key to make it appear as if it has more space.

In quasi-permissionless setting one can stop replotting by making parties commit their space on-chain and doing frequent checks that they have not deleted their space $e.g.$ *Filecoin*.

Replotting

Replotting attacks are inherent to $Proof - of - Space$ in a fully permissionless and dynamically available settings. The adversary re-initiates its space using a different key to make it appear as if it has more space.

In quasi-permissionless setting one can stop replotting by making parties commit their space on-chain and doing frequent checks that they have not deleted their space $e.g.$ *Filecoin*.

For our secure functions in discrete model, bounding total weight that can go into a block mitigates the replotting attacks. Though a more thorough study is required.

Recording Resource On-chain, Network Delay And Other Attacks

Our model assumes that total resource gets accurately measured on-chain. While it may not be possible to fully achieve this, we can approximate it well:

Recording Resource On-chain, Network Delay And Other Attacks

Our model assumes that total resource gets accurately measured on-chain. While it may not be possible to fully achieve this, we can approximate it well:

- Collecting top k solves to the challenge, say partial solutions to PoW.
- Averaging over multiple blocks, akin to how it works for bitcoin difficulty.
- For VDF's we just require the fastest one.

Recording Resource On-chain, Network Delay And Other Attacks

Our model assumes that total resource gets accurately measured on-chain. While it may not be possible to fully achieve this, we can approximate it well:

- Collecting top k solves to the challenge, say partial solutions to PoW.
- Averaging over multiple blocks, akin to how it works for bitcoin difficulty.
- For VDF's we just require the fastest one.

We do not formally model network delays. This would bring an additional factor of $\chi(\Delta)$ for the resource gap. In case of PoW this has been extensively studied. Similar techniques should apply in our case.

Recording Resource On-chain, Network Delay And Other Attacks

Our model assumes that total resource gets accurately measured on-chain. While it may not be possible to fully achieve this, we can approximate it well:

- Collecting top k solves to the challenge, say partial solutions to PoW.
- Averaging over multiple blocks, akin to how it works for bitcoin difficulty.
- For VDF's we just require the fastest one.

We do not formally model network delays. This would bring an additional factor of $\chi(\Delta)$ for the resource gap. In case of PoW this has been extensively studied. Similar techniques should apply in our case.

We also do not model attacks like *grinding* and *double-dipping* but these are well-studied and we assume they are taken care of already.

Main Takeaway: A New Set of Weight Rules

[5] "Bitcoin: A Peer-to-Peer Electronic Cash System"; "The Chia Network Blockchain"; "Minotaur: Multi-Resource Blockchain Consensus" (Nakamoto 2009; Cohen and Pietrzak 2019; Fitzi et al. n.d.)

Main Takeaway: A New Set of Weight Rules

Weight functions like $W, SV, W_1 + \cdots + W_k$[5] we previously known.

[5] "Bitcoin: A Peer-to-Peer Electronic Cash System"; "The Chia Network Blockchain"; "Minotaur: Multi-Resource Blockchain Consensus" (Nakamoto 2009; Cohen and Pietrzak 2019; Fitzi et al. n.d.)

Main Takeaway: A New Set of Weight Rules

Weight functions like $W, SV, W_1 + \cdots + W_k$[5] we previously known.

We show a vast class of weight functions for fully-permissionless setting: $\Gamma(S, V, W)$ which is

1. monotonically increasing
2. homogeneous in V, W (the timed resources)
3. sub-homogeneous in S.

[5] "Bitcoin: A Peer-to-Peer Electronic Cash System"; "The Chia Network Blockchain"; "Minotaur: Multi-Resource Blockchain Consensus" (Nakamoto 2009; Cohen and Pietrzak 2019; Fitzi et al. n.d.)

Main Takeaway: A New Set of Weight Rules

Weight functions like $W, SV, W_1 + \cdots + W_k$[5] we previously known.

We show a vast class of weight functions for fully-permissionless setting: $\Gamma(S, V, W)$ which is

1. monotonically increasing
2. homogeneous in V, W (the timed resources)
3. sub-homogeneous in S.

Interesting examples:
$\sqrt{W_1 \cdot W_2},\ W_1^{0.3} \cdot W_2^{0.2} \cdot W_3^{0.5},\ \min\{W_1, W_2\},\ SW,\ S \cdot \sqrt{WV}$

[5] "Bitcoin: A Peer-to-Peer Electronic Cash System"; "The Chia Network Blockchain"; "Minotaur: Multi-Resource Blockchain Consensus" (Nakamoto 2009; Cohen and Pietrzak 2019; Fitzi et al. n.d.)

Main Takeaway: A New Set of Weight Rules

Weight functions like $W, SV, W_1 + \cdots + W_k$[5] we previously known.

We show a vast class of weight functions for fully-permissionless setting: $\Gamma(S, V, W)$ which is

1. monotonically increasing
2. homogeneous in V, W (the timed resources)
3. sub-homogeneous in S.

Interesting examples:
$\sqrt{W_1 \cdot W_2},\ W_1^{0.3} \cdot W_2^{0.2} \cdot W_3^{0.5},\ \min\{W_1, W_2\},\ SW,\ S \cdot \sqrt{WV}$

These provide different economic incentives and may provide additional decentralizing force.

[5] "Bitcoin: A Peer-to-Peer Electronic Cash System"; "The Chia Network Blockchain"; "Minotaur: Multi-Resource Blockchain Consensus" (Nakamoto 2009; Cohen and Pietrzak 2019; Fitzi et al. n.d.)

Future Work

Future Work

[6] "Minotaur: Multi-Resource Blockchain Consensus" (Fitzi et al. n.d.)

Future Work

- Analyze under network delays and different synchrony models.

[6] "Minotaur: Multi-Resource Blockchain Consensus" (Fitzi et al. n.d.)

Future Work

- Analyze under network delays and different synchrony models.
- Economic analysis for various rules in fully permissionless setting to derive economic security.

[6] "Minotaur: Multi-Resource Blockchain Consensus" (Fitzi et al. n.d.)

Future Work

- Analyze under network delays and different synchrony models.
- Economic analysis for various rules in fully permissionless setting to derive economic security.
- Understand the landscape of weights for heaviest-chain rules in dynamically-available setting and add Proof-of-Stake [6].

[6] "Minotaur: Multi-Resource Blockchain Consensus" (Fitzi et al. n.d.)

Conclusion

- We introduce a new idealized model to study secure weight function rules in a fully-permissionless setting.
- We characterize secure weight functions as those that are monotonically increasing, homogeneous in V, W and sub-homogeneous in S.
- Please see our paper for more details and discussion.

Conclusion

- We introduce a new idealized model to study secure weight function rules in a fully-permissionless setting.
- We characterize secure weight functions as those that are monotonically increasing, homogeneous in V, W and sub-homogeneous in S.
- Please see our paper for more details and discussion.

Thank you!