The Cost of Restaking vs. Proof-of-Stake*

Akaki Mamageishvili[1] and Benny Sudakov[2]

[1]Offchain Labs
[2]ETH Zurich

May 2025

Abstract

We compare the efficiency of restaking and Proof-of-Stake (PoS) protocols in terms of stake requirements. First, we consider the sufficient condition for the restaking graph to be secure. We show that the condition implies that it is always possible to transform such a restaking graph into secure PoS protocols. Next, we derive two main results, giving upper and lower bounds on required extra stakes that one needs to add to validators of the secure restaking graph to be able to transform it into secure PoS protocols. In particular, we show that the restaking savings compared to PoS protocols can be very large and can asymptotically grow in the worst case as a square root of the number of validators. We also study a complementary question of transforming secure PoS protocols into an aggregate secure restaking graph and provide lower and upper bounds on the PoS savings compared to restaking.

1 Introduction

Restaking has recently been proposed as a cryptoeconomically more efficient alternative to the Proof-of-Stake solution concept for blockchains. The latter was proposed on its own as an energy-efficient alternative to the Proof-of-Work concept [1]. In restaking, validators that are staked for one protocol can reuse their stake to secure other projects or protocols of their choice. This gives a boost to newer projects in attracting capital for their *cryptoeconomic security*. Cryptoeconomic security refers to a security derived from staked parties being slashed their stakes in case of misbehavior. Restaking seems to be attractive for the validators of the original protocol as well. In particular, restaking allows them to reuse their computational power and storage allocation for additional

*We thank Ed Felten, Manvir Schneider, Paolo Penna, Pranay Anchuri, Christoph Schlegel and Bruno Mazorra for useful discussions. No AI was consulted when working on the paper. All errors are our own.

[1]See [6] for a more detailed discussion on Proof-of-Work vs. Proof-of-Stake concepts.

payoffs coming from other projects' rewards. The original protocol, for which the validators are staked, may have requirements that do not exhaust all the resources of the validators, usually for the purpose of decentralization[2]. Another potential reason for such availability might be economies of scale for establishing further validation services.

The idea of restaking was originally proposed in the whitepaper of the Eigenlayer project, [7]. A similar concept existed in traditional finance literature under the name of *rehypothecation*, [4]. The interdependencies between protocols and validators introduced by restaking create new risks, which have been modeled and analyzed in the original whitepaper, as well as in the series of papers [3], [1], [8], [5]. The simplest way to verify the security of the restaking uses a cost-benefit analysis. In this analysis, the total amount of lost stakes of attackers is compared to the total value obtained from the attack of a set of validators. The same cost-benefit analysis can be used to check the security of a Proof-of-Stake protocol, in which a single project/service is secured by multiple validators. This allows a reasonable comparison of restaking and multiple PoS protocols. [2] discusses ways to estimate the value gained from attacking a protocol and cryptoeconomic security more in detail.

Since the Proof-of-Stake protocol is simpler and hence safer solution concept than restaking, we are interested in checking comparative benefits of restaking protocols. One particular attractive point of restaking protocols can be the lower total staking requirements than that of Proof-of-Stake protocols, since there are opportunity costs associated with locking up stake for security reasons. Most PoS protocols pay for this cost by rewarding stakers with newly issued tokens, which leads to inflation. Hence, minimizing the total stake amount while maintaining security can be a reasonable goal in both restaking or PoS protocols. Starting with a secure restaking protocol and initial stake endowments for all validators, we measure how much extra stake would be required if validators decided to form separate Proof-of-Stake protocols with their stake amounts, dividing them across the same set of projects they secure in the original restaking graph. We only allow all validator stakes to (weakly) increase. This captures a real-life consideration in which validators' stakes can not be taken away, but their stakes can be increased through financing when needed. If both increasing and decreasing of validator stakes were allowed, it would be trivial to construct a restaking graph with minimum total stake. This amount is equal to the sum of all project values, and secure PoS protocols for all projects can be constructed by dividing the stakes of the validators.

Our Contributions First, we check the sufficient condition for a secure restaking graph, specified by the EigenLayer project [7]. Although this sufficient condition is the only currently available (generic) condition that can be checked in polynomial time in the size of the restaking graph, it does not save

[2]Check out Ethereum network plans to resolve this asymmetry between validator resources using rainbow staking `https://ethresear.ch/t/unbundling-staking-towards-rainbow-staking/18683`.

on total required stake. In particular, it is possible to divide the initial stakes to obtain secure PoS protocols across all projects. To further extend the analysis, we define *restaking savings* of a secure restaking graph. It is equal to the extra stake required, relative to the original total available stake in the graph, to be able to obtain secure PoS protocols through the division of stake described above. Higher restaking savings indicate an advantage of restaking to PoS. To upper bound this value, we use simple secure PoS constructions from the original restaking graph. In particular, we upper bound the value of restaking savings with the highest degree among the projects. Another upper bound is the minimum incidence of any validator in the cover of a validator set by project neighborhoods. The third upper bound is equal to second largest multiplicative inverse of the security parameter, which is a small constant value in many practical cases. The last upper bound is equal to the square root of the number of validators. Next, we give a construction to lower bound this value. The lower bound example asymptotically matches all upper bounds obtained.

We study a complementary question of aggregating given secure PoS protocols into a secure restaking graph. When aggregating, each validator adds stakes across projects that they secure in separate PoS protocols, resulting in a restaking graph. Similarly to the restaking savings, we define *PoS savings* as the total additional stake required to add validators in the aggregate restaking graph to make it secure. We first construct an example showing that the resulting restaking graph is not secure, in fact, at least an additional stake equal to the original total stake is required to make it secure, thus showing a lower bound of 1 on PoS savings. We also provide an upper bound on PoS savings as a function of the underlying restaking graph.

2 Model

Our notation closely follows [3]. A restaking graph $G = (S, \pi, \alpha, V, \sigma, E)$ consists of the following:

- S denotes the set of m services $\{1, ..., m\}$,
- $\pi \in \mathbb{R}_+^m$ denotes values of services (we will refer projects as services from now on throughout the paper),
- $\alpha \in \mathbb{R}_+^m$ denotes security parameters of services, where $\alpha_s \in [0, 1]$,
- V denotes the set of n validators $\{1, ..., n\}$,
- $\sigma \in \mathbb{R}_+^n$ denotes stakes of validators,
- The set of edges E connecting S nodes (services) to V nodes (validators).

An edge $(s, v) \in E$ indicates that a service s is secured by a validator v. Hence, a restaking graph is represented as a bipartite graph with two parts made up of services and validators.

Let $N_G(s)$ denote the neighborhood of the service $s \in S$ in the restaking graph G, i.e. $N_G(s) := \{v \in V : (s, v) \in E\}$, is the set of validators staked on service s. Similarly, $N_G(v)$ denotes the neighborhood of the validator $v \in V$, i.e., $N_G(v) := \{s \in S : (s, v) \in E\}$, is the set of services validator v is staked on.

A service s has a security parameter α_s if only subsets of validators staked on this server that have at least α_s fraction of the total stake of validators staked on s can attack it. More formally:

Definition 1. *A subset of validators $W \subseteq V$ can attack the service s if and only if*

$$\frac{\sum_{k \in W \cap N_G(s)} \sigma_k}{\sum_{v \in N_G(s)} \sigma_v} > \alpha_s.$$

For each subset of validators A, we can define the maximal set of services that they can collectively attack. Let this set be denoted by $M(A)$. That is $M(A) = \{s \in S : A \text{ can attack } s\}$. The attack carried out by the validators in A is profitable when $\sum_{s \in M(A)} \pi_s > \sum_{v \in A} \sigma_v$, that is, validators lose less total stake than the total value they obtain from the attack.

Definition 2. *A restaking graph G is secure if there exists no subset of validators A that can profitably attack its corresponding set of services $M(A)$.*

We are interested in the following question:

Question 1. *Is it possible to divide the stakes of all validators between all services they secure in the graph G so that all services are secure in their corresponding PoS protocols?*

A division of stake σ_v of validator v across services it secures in G induces a vector $c^v \in \mathbb{R}_+^{|N_G(v)|}$, such that, $\sum_{s \in N_G(v)} c_s^v = \sigma_v$. We say that division of stake is secure if in the corresponding Proof-of-Stake protocols, that represent star graphs centered at the service, we have guaranteed security. That is, for each service s with security parameter α_s, a vector of stakes $\{k \in N_G(s) : c_s^k\}$ is secure, which means that there is no profitable attack by validators having stakes $\{k \in N_G(s) : c_s^k\}$. Let $\mathcal{W}(G)$ denote all initial stake vectors σ for which such division is possible. To simplify notation, here we implicitly assume that the stake vector is not part of a restaking graph G, and only keep its combinatorial nature together with the π and α vectors.

We consider stake vectors σ' that (weakly) dominate stake vector σ, that is, $\sigma'_s \geq \sigma_s$ for any s. The set of all stake vectors that dominate σ is denoted by $D(\sigma)$. Let $T(\sigma) := \sum_{v \in V} \sigma_v$. Then, the total extra value to reach from the stake vector σ to the stake vector σ' is $T(\sigma') - T(\sigma)$. The rationale behind considering (weakly) dominant vectors of stakes is that we assume that initial stakes cannot be taken away from the validators, but they can be increased if needed.

The Eigenlayer project, [7], specifies a sufficient condition when a restaking graph is secure: for any $v \in V$, the following inequality holds:

$$\sigma_v \geq \sum_{s \in N_G(v)} \frac{\sigma_v}{\sum_{k \in N_G(s)} \sigma_k} \frac{\pi_s}{\alpha_s}. \tag{1}$$

This condition allows to check whether a given restaking graph is secure in polynomial time in the input graph size. In fact, the time is even linear in the number of edges $|E|$ [3].

3 Results

In this section, we first check the sufficient condition (1), then define restaking and Proof-of-Stake savings, and last, we obtain lower- and upper-bound results for them.

3.1 Restaking Savings

If the sufficient condition on validator stakes (1) is satisfied, there is a division of stakes such that all services are secure in their corresponding PoS protocols. More formally, we obtain the following result:

Proposition 1. *When the stake vector σ satisfies* (1)*, then $\sigma \in \mathcal{W}(G)$.*

Proof. Consider the following division of stakes: the validator v allocates at least

$$\frac{\sigma_v}{\sum_{k \in N_G(s)} \sigma_k} \frac{\pi_s}{\alpha_s}$$

stake to a service s it secures in G. Such allocation is possible because of (1). Each service s is assigned a total stake amount of at least $\frac{\pi_s}{\alpha_s}$, since

$$\sum_{v \in N_G(s)} \frac{\sigma_v}{\sum_{k \in N_G(s)} \sigma_k} \frac{\pi_s}{\alpha_s} = \frac{\pi_s}{\alpha_s},$$

by bringing the outer summation operator into the nominator. With $\frac{\pi_s}{\alpha_s}$ total stake amount in PoS protocol for service s, there is no profitable attack, since for the successful attack the attackers should control at least $\frac{\pi_s}{\alpha_s}\alpha_s = \pi_s$ stake, hence, it cannot be profitable. □

Note that the value $\frac{\pi_s}{\alpha_s}$ is the minimum required total stake in a PoS protocol for a service s with value π_s and security parameter α_s that guarantees that there is no profitable attack. For any smaller amount $t < \frac{\pi_s}{\alpha_s}$, there exists a distribution of the total stake t among any number of validators such that there is a profitable attack.

There is a minimum total stake requirement to ensure that there is a division in secure PoS protocols. The amount is equal to $\sum_{s \in S} \pi_s$. It can be distributed arbitrarily among validators, with the only guarantee that each service can

[3]The general decision problem whether a given restaking graph is secure is difficult, see [3].

get its own value staked by one of the validators securing it. However, such a distribution does not take into account that validators already have stakes, which cannot be taken away from them. Another consideration is that the restaking graph with the minimum total stake in $\mathcal{W}(G)$ is not necessarily secure, while we will focus on secure restaking graphs from now on.

Next, we define the restaking savings for a given secure restaking graph:

Definition 3. *For a given secure restaking graph G, restaking savings, $RS(G)$, denotes a minimum extra total value to add to the stakes in σ vector, in relative terms, so that secure subdivision of the new stake vector is possible. That is,*

$$RS(G) = \min_{\sigma' \in D(\sigma) \cap \mathcal{W}(G)} \frac{T(\sigma') - T(\sigma)}{T(\sigma)}.$$

Let $d_G(u)$ denote the degree of vertex $u \in S \cup V$ in graph G. Note that $d_G(u) = |N_G(u)|$. Consider a subset of services whose neighborhoods cover the entire set of validators V:

$$\mathcal{R} := \{R : R \subseteq S \& \cup_{s \in R} N_G(s) = V\}.$$

For each of these covers, calculate the incidences of all validators in it, and take the largest incidence number. Let K be minimum such number over all elements of $\mathcal{R}$. Formally,

$$K = \min_{R \in \mathcal{R}} \{k | k = \max_{v \in V} \sum_{s \in R} \mathbb{I}(v \in N_G(s))\},$$

where $\mathbb{I}(P) = 1$ whenever the statement P is true.

We prove several upper bounds on the restaking savings, as a function of different parameters of the graph G. Note that since the original graph is secure, we have the following inequality:

$$T(\sigma) = \sum_{v \in V} \sigma_v \geq \sum_{s \in S} \pi_s = T(\pi).$$

Otherwise, there is a profitable attack that involves all validators V.

Proposition 2. *$RS(G)$ is upper bounded by the following values:* $\max_{s \in S}(d_G(s))$ *and K.*

Proof. First, we list three types of secure PoS constructions for a service $s \in S$.

1. the vector of stakes in a corresponding PoS protocol $(c_1, ..., c_{d_G(s)})$ is equal to $(0, ..., 0, \pi_s, 0, ...0)$. This construction is secure for any α_s since the only validator that can attack the service is losing at least the value it is gaining.

2. the vector of stakes $(c_1, ..., c_{d_G(s)})$ is equal to $(0, ..., 0, \pi_s + \sigma_v, 0, ...0)$ for any v. This construction is secure for a similar reason to above.

3. the vector of stakes $(c_1, ..., c_{d_G(s)})$ is equal to $(\sigma_{v_1}, ..., \sigma_{v_{d_G(s)}})$, where $N_G(s) = \{v_1, ..., v_{d_G(s)}\}$. This construction is secure for any α_s since otherwise the original restaking graph G would not be secure: validators that secure the service s have a profitable attack on the service s.

We can add the value π_s to all stakes in $N_G(s)$, i.e., add it $d_G(s)$ times, and use subdivision for each validator v that allocates $\sigma_v + \pi_s$ once to some service s in its neighborhood $N_G(v)$ and π_k to all other corresponding services $k \in N_G(v) \setminus \{s\}$. This way, all services are secure. Also all original stake sizes are used, so that the resulting stake vector is in $D(\sigma)$. At the same time, extra value added is upper bounded by

$$\sum_{s \in S} d_G(s)\pi_s \leq \max_{s \in S}(d_G(s)) \sum_{s \in S} \pi_s \leq \max_{s \in S}(d_G(s)) \sum_{v \in V} \sigma_v.$$

Next, we show $RS(G) \leq K$. Consider a cover of vertices in V by neighborhoods of services $s \in R \subseteq S$ that has the minimum maximum incidence number K. For each neighborhood $N_G(s)$, where $s \in R$, use construction 3: original $(\sigma_{v_1}, ..., \sigma_{v_{d_G(s)}})$ stakes from the restaking graph to secure service s in the PoS protocol. To be able to do that, we need to add at most $K - 1$ times all σ_v stakes, for any $v \in V$. In this way, we utilize all σ_v stakes and secure services in the cover set R. For service s in $S \setminus R$, we add π_s stake once to any of its neighbors in $N_G(s)$ and use construction $(0, , , 0, \pi_s, 0, ..., 0)$ to secure service s in a corresponding PoS protocol. This makes sure we have secured all services and utilized all stakes of all validators. Then, $RS(G) \leq (K - 1) + 1 = K$. □

Note that $RS(G) \leq K$ in particular implies that $RS(G) \leq \max_{s \in S}(d_G(s))$, as K is upper bounded by $\max_{s \in S}(d_G(s))$. There are examples where $K = \max_{s \in S}(d_G(s))$, implying that the upper bound can be as bad as n. However, next, we show an upper bound as a function of the number of validators, which is asymptotically much lower than n.

Theorem 1. *For any secure restaking graph G, the following inequality holds $RS(G) \leq 2\sqrt{n} - 1$.*

Proof. The proof combines constructions from the proof of Proposition 2. Initialize the set $S_L := \emptyset$ and $V_L := \emptyset$. The following procedure is repeated until it is no longer possible: At each step, find a service s in $S \setminus S_L$ such that its degree in the set $V \setminus V_L$ is at least $\sqrt{n}$. Update the set $S_L := S_L \cup s$ and $V_L := V_L \cup N_G(s)$. Note that after at most $\sqrt{n}$ steps, it is not possible to find a new service s and, hence, the procedure stops. That is, for the resulting set of services S_L, we have $|S_L| \leq \sqrt{n}$. This, in particular, implies that the maximum outdegree of a validator v in V_L to S_L, denoted by $d_{S_L}(v)$, is upper bounded by $\sqrt{n}$.

Add to each validator $v \in V_L$ a stake amount $(d_{S_L}(v) - 1)\sigma_v$. This allows to secure any service $s \in S_L$ by using a construction $(\sigma_{v_1}, ..., \sigma_{v_{d_G(s)}})$ from the proof of Proposition 2. Together with securing all services in S_L, we also utilize validator stakes in V_L, and this is done by adding at most $(\sqrt{n}-1)T(\sigma)$ extra stake.

We can secure any service $s \in S \setminus S_L$ by adding stakes π_s for to all its neighbors in $V \setminus V_L$, that is, by adding $d_{V \setminus V_L}(s)\pi_s$ extra stakes. By the construction of the set S_L we know that $d_{V \setminus V_L}(s)$ is upper bounded by $\sqrt{n}$ for any $s \in S \setminus S_L$. We can, at the same time, utilize the remaining stakes in $V \setminus V_L$, by using a construction of type $(0, .., 0, \pi_s + \sigma_v, ..0)$ for $v \in V \setminus V_L$. The last step is possible, because for each validator in $V \setminus V_L$, there must be at least one service in $S \setminus S_L$ it secures. The total extra stake is upper bounded by $\sqrt{n}T(\pi_{S \setminus S_L}) \leq \sqrt{n}T(\sigma)$, where $T(\pi_{S \setminus S_L})$ denotes the sum $\sum_{s \in S \setminus S_L} \pi_s$. By summing up two values of extra stakes used to secure S_L and $S \setminus S_L$ sets, and also utilizing all stakes in V, we get

$$RS(G) \leq \frac{(\sqrt{n}-1)T(\sigma) + \sqrt{n}T(\sigma)}{T(\sigma)} = 2\sqrt{n} - 1,$$

a required bound of the Theorem claim. □

Note that in the proofs of Proposition 2 and Theorem 1, we did not use any property of α values. Both upper bounds use combinatorial features of the underlying restaking graph G. Next, we show an upper bound that is a function of the security parameters α_s. The upper bound is interesting in practical cases, since the security parameters of most protocols are usually from the set of fractions $\{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}\}$. These values come from Byzantine fault-tolerant mechanisms, used to achieve consensus on the state of the chain. We show the following upper bound on $RS(G)$:

Proposition 3. *Assume that services are labeled in the way that the security parameters are sorted in increasing order $\alpha_1 \leq \alpha_2 \leq ... \leq \alpha_m$. Then, the following inequality $RS(G) \leq \frac{1}{\alpha_2}$ holds.*

Proof. For the service $s = 1$, allocate the original stakes of the restaking graph in the PoS protocol: $(c_1, ..., c_{d_G(1)}) = (\sigma_{v_1}, ..., \sigma_{v_{d_G(1)}})$. This PoS protocol is secure. Repeat the following procedure for all services $s > 1$:

1. add stake $t_s = \frac{\pi_s}{\alpha_s}$ to any validator $v \in N_G(s)$,
2. use construction $(\sigma_{v_1}, ..., \sigma_{v_{d_G(s)}})$ to secure service s in PoS protocol,
3. set all values in the vector $(\sigma_{v_1}, ..., \sigma_{v_{d_G(s)}})$ to 0.

Each PoS derived at step 2. is secure, as the total stake allocated for a service $s > 1$ is at least $\frac{\pi_s}{\alpha_s}$.

In this way, we add $\sum_{s \in S, s>1} \frac{\pi_s}{\alpha_s}$ total extra stake to the original stakes of validators. This implies the following chain of inequalities:

$$RS(G) \leq \frac{\sum_{s \in S, s>1} \frac{\pi_s}{\alpha_s}}{T(\sigma)} \leq \frac{1}{\alpha_2} \frac{\sum_{s \in S, s>1} \pi_s}{T(\pi)} \leq \frac{1}{\alpha_2}.$$

□

Next, we construct an example of a secure restaking graph that derives a lower bound on the restaking savings, that is linear in the number of services and asymptotically matches to all upper bounds derived so far.

Theorem 2. *For any $m \in \mathbb{N}$, there are instances of a secure restaking graph G in which the restaking savings $RS(G) \in \Theta(m)$.*

Proof. Assume the number of validators is $n = m^2 + 1$ for some $m \in \mathbb{N}$. The edge set E consists of m edges $(s, m^2 + 1)$, for any $1 \leq s \leq m$ and m^2 edges of type $(s, (s-1)m + j)$, for any $1 \leq s \leq m$, $1 \leq j \leq m$.

The value of service s is defined as $\pi_s = 2$ for any $1 \leq s \leq m$. Any validator with index less than $v < m^2 + 1$ has stake $\sigma_v = \frac{1}{m}$. The last validator has stake $\sigma_{m^2+1} = 2m$. Security parameter of service s is defined as $\alpha_s = \frac{1}{2m+1}$.

Intuitively, there is one validator with large stake and many validators with equal small stakes. Each service is easy to attack, but the large staked validator does not find it profitable to attack. On the other hand, each service is hard enough to attack by all small staked validators that secure it. Each service is secured by m small staked validators. See the Figure 1 for $m = 3$.

First, we show that G is a secure restaking graph. Note that the large validator, indexed with $m^2 + 1$ and with stake $\sigma_{m^2+1} = 2m$ does not want to participate in any attack, as she is losing at least the total value available across all services. No service $s \in \{1, 2, ..., m\}$ can be attacked by all validators securing it other than the large validator, since the service security parameter is $\alpha_s = \frac{1}{2m+1}$, while all small validators indexed $((s-1)m+1, (s-1)k+2, ..., sm)$ make up stake 1 in total. The total available stakes securing a service s is $2m+1$.

To show a lower bound on $RS(G)$, first we show that when any service with value 2 and security parameter $\frac{1}{2m+1}$ gets initial allocation of stakes in PoS protocol equal to a vector $(\frac{1}{m}, ..., \frac{1}{m}, 0)$, there needs to be at least an extra m stake to make the PoS protocol secure. Suppose we add stakes such that new vector of stakes becomes $(a_1, ..., a_m, a_{m+1})$. From now on we distinguish three cases:

1. If there is a subset of validators, whose stakes sum up to a number between 1 and 2, then it must be that the total stake, $\sum_{i=1}^{m+1} a_i$, is at least $2m+1$. Otherwise, this subset would be able to attack the service and the attack would be profitable. This implies that the extra stake size is at least $2m = 2m + 1 - m\frac{1}{m}$.

2. If there is a subset of validators, whose stakes sum up to a number between 0.5 and 1, then it must be that the total stake, $\sum_{i=1}^{m+1} a_i$, is at least $m+1$. Otherwise, this subset would be able to attack the service and the attack would be profitable. This implies that the extra stake size is at least m.

3. If there is no such subset for either of the cases 1. and 2., then it must be that more than half of the numbers among $(a_1, ..., a_m, a_{m+1})$ are at least 2, which implies that at least $\frac{m+1}{2}(2 - \frac{1}{m}) \geq m$ extra stake was added to the initial stake distribution of $(\frac{1}{m}, ..., \frac{1}{m}, 0)$.

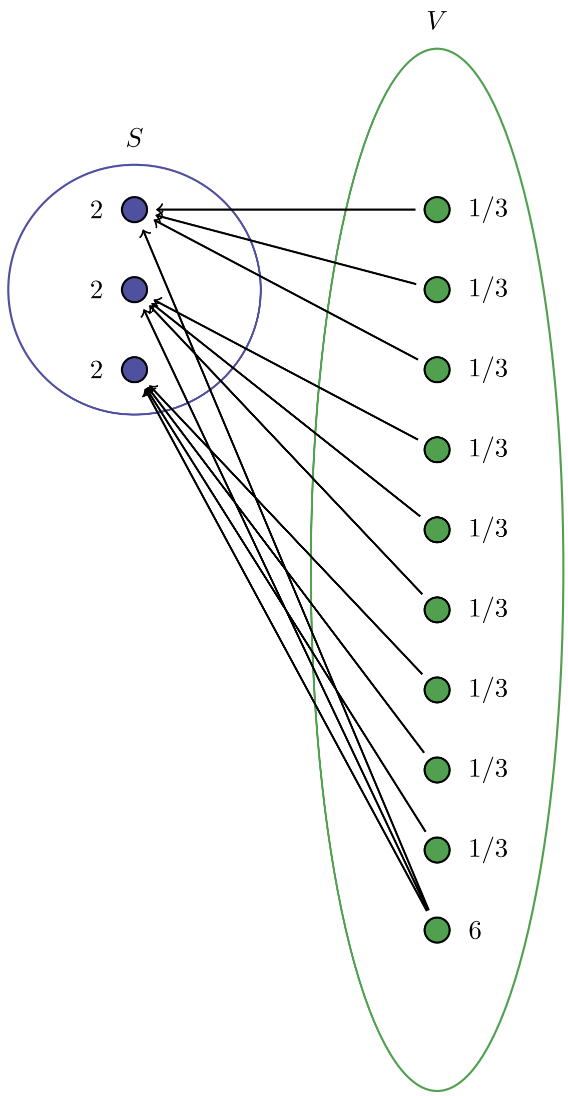

Figure 1: Example construction for $m = 3$.

Initially, $2m$ stake of validator indexed m^2+1 is available, therefore, we need to add at least extra $m \cdot m - 2m$ stake:

- there are m services to secure,
- and for each service we need to add extra stake at least m.

On the other hand, original total stake amount is $T(\sigma) = 3m$, therefore, restaking savings satisfy

$$RS(G) \geq \frac{m \cdot m - 2m}{3m} \in \Omega(m).$$

This shows a lower bound on the required stake.

It is easy to show that $O(m^2)$ total stake is enough to add to validators in V, so that the resulting stake vector $\sigma \in \mathcal{W}(G)$. It can, for example, be done by adding the stake $2m^2 - m$ to the validator $m^2 + 1$ so that it total stake become $(2m+1)m$. Then splitting this stake into m stakes of $2m+1$ each in each of m PoS protocols. Small stakers in these protocols can not attack any of the services even if they all coordinate, while the large staker does not find any attack profitable. This proves the claim of the theorem, $RS(G) \in \Theta(m)$.

□

The sets S_L and V_L from the proof of Theorem 1 end up being the sets S and V, respectively. The highest degree in V_L is equal to $m = \lfloor \sqrt{n} \rfloor$. This provides an asymptotically matching lower bound to the upper bound result derived in Theorem 1.

Corollary 1. *There are instances of secure restaking graphs G in which $RS(G) \in \Theta(\sqrt{n})$.*

Note that in the lower bound example construction from Theorem 2, inverse of the security parameter $\frac{1}{\alpha_s} = 2m + 1$ for any $s \in S$, hence, this construction gives an asymptotically matching lower bound to the $\frac{1}{\alpha_s} - 1$ upper bound as well. The upper bounds obtained in Proposition 2, $\max_{s \in S} d_G(s)$ and K, are also of the order $\Theta(m)$, hence asymptotically matching the lower bound from Theorem 2.

The total stake in the example of Theorem 2 is equal to $3m$, while the lower bound requirement on the total stake derived from condition (1) is equal to $\sum_{s \in S} \frac{\pi_s}{\alpha_s} = m(2m+1)$, which shows that in some cases the condition (1) requires a substantially higher total stake than would be enough to secure the restaking graph. Moreover, the condition for any validator v except the last validator $v = m^2 + 1$ translates into

$$\frac{1}{m} \geq \frac{1/m}{2m+1} \cdot \frac{2}{\frac{1}{2m+1}} = \frac{2}{m},$$

which is violated by a factor two. The condition for the validator $v = m^2 + 1$ is violated by a very large multiplicative factor, as it is equivalent to:

$$2m \geq \sum_{s=1}^{m} \frac{2m}{2m+1} \cdot \frac{2}{\frac{1}{2m+1}} = 4m^2.$$

Definition 4. *Let $RS := \sup_G RS(G)$ denote the restaking savings in the extremal case on all secure restaking graphs.*

Then, the direct corollary of Theorem 2 is the following:

Corollary 2. $RS = \infty$.

That is, restaking savings are not bounded by a constant, and they may grow unboundedly as the number of services and validators grow. Note that we need both values to grow unboundedly. Otherwise, upper bounds obtained in Proposition 2 ensure that $RS(G)$ is constant.

3.2 PoS Savings

Suppose a PoS protocol centered at any service $s \in S$, having security parameter α_s and allocated σ_v^s stakes for any $v \in N_G(s)$ is safe, by the same definition as above: no attacking set of validators find it profitable to attack service s. Let the partition of the stake vector σ be denoted by σ^S. Consider the total stake for any validator $v \in V$ defined as $\sigma_v = \sum_{s \in N_G(v)}$. That is, we aggregate all stakes that the validator has staked across all PoS protocols in which it participates. We ask a similar question to the previous section:

Question 2. *How much stake do we need to add to the aggregate validityator stakes, σ_v, so that the resulting restaking graph is secure?*

The security of the restaking graph is considered by the original Definition 2. More formally, for each restaking graph, we define:

Definition 5. *For a given aggregate restaking graph G, PoS savings, $PoSS(G)$ denotes a minimum additional total value to stakes in σ vector in relative terms to $T(\sigma)$ so that the resulting restaking graph G is secure:*

$$PoSS(G) = \min_{\sigma' \in D(\sigma)} \frac{T(\sigma') - T(\sigma)}{T(\sigma)}$$

First, we note the following inequality that relates the total value to capture from services and the total staked amount of validators. It will be useful to obtain an upper bound on the PoS savings.

Lemma 1. $T(\sigma) \geq T(\pi)$.

Proof. A PoS protocol centered at $s \in S$ is secure implies that

$$\sum_{v \in N_G(s)} \sigma_v^s \geq \pi_s, \tag{2}$$

as otherwise all validators in $N_G(s)$ would be able to profitably attack the service s. Summing up left hand side of (2) for all $s \in S$ and the definition of σ_v imply the claim of the lemma:

$$T(\sigma) = \sum_{v \in V} \sigma_v = \sum_{s \in S} \sum_{v \in N_G(s)} \sigma_v^s \geq \sum_{s \in S} \pi_s = T(\pi).$$

□

Similarly to restaking savings RS, we can define a measure $PoSS$, which is maximized on all instances of the underlying graph G and initial secure PoS protocols.

Definition 6. *Let* $PoSS := \sup_{G, \sigma_v^S} PoSS(G)$ *denote the extreme value of PoS savings, where the stake vector* $\{\sigma_v^s : v \in N_G(s)\}$ *constitutes the secure PoS protocol for any* $s \in S$.

We obtain the following lower bound on this value, by constructing a family of graphs G with an increasing number of services for which $PoSS(G)$ converges to 1.

Proposition 4. $PoSS \geq 1$.

Proof. Consider PoS protocols with m services, each service $s \in S$ having value $\pi_s = 1$, security parameter $\alpha_s = \frac{1}{m+1}$ and two validators, labeled s and $m+1$ with stakes equal to $\sigma_s^s = 1$ and $\sigma_{m+1}^s = \frac{1}{m}$. Note that all PoS protocols are secure:

- large staked validator s does not attack since it is burning stake equal to the value it obtains,
- small staked validator $m+1$ can not attack the service since its stake share is equal to $\frac{\frac{1}{m}}{1+\frac{1}{m}} = \frac{1}{m+1}$, that is the same as a security parameter of the service.

We claim that the minimum extra stake required is $m-1$. In the resulting restaking protocol, validator $m+1$ has stake equal to $m\frac{1}{m} = 1$. It can attack all services unless one of the other validators gets an extra $m-1$ stake, in which case the claim is trivially true. This amount is enough. Consider adding stake $m-1$ to the validator $m+1$ so that it now controls the stake amount m. This validator does not attack services since it is losing at least the amount she can gain by attacking. Other validators are not able to attack their corresponding services, and it is also not profitable to attack.

Hence, $PoSS \geq \frac{m-1}{m+1}$ for any m, since the original total stake amount among the validators was $m+1$. Taking $m \to \infty$ gives $PoSS \geq 1$. □

Next, we provide an upper bound on $PoSS(G)$, similar to the upper bound obtained for restaking savings in Proposition 2.

Proposition 5. $PoSS(G) \leq \max_{s \in S} d_G(s)$.

Proof. For each service $s \in S$, add to all its validators $v \in N_G(s)$ a stake equal to π_s. The resulting restaking graph is secure. For any subset of validators, $U \subseteq V$, they lose more stakes than the value they derive from the maximal set they attack, $M(U)$. The reason for this is that by our construction, at least one of the validators in U received an additional stake of π_s, for any $s \in M(U)$. We added an extra stake of

$$\sum_{s \in S} d_G(s)\pi_s \leq \max_{s \in S} d_G(s)T(\pi) \leq \max_{s \in S} d_G(s)T(\sigma),$$

where the last inequality is derived in Lemma 1. This completes the proof of the claim of the proposition. □

4 Conclusion

We started a framework to compare restaking and implied Proof-of-Stake protocols in terms of total stake requirements. We provide asymptotically matching lower and upper bounds on restaking savings, as well as a constant lower bound on the PoS savings and nonconstant upper bounds on it. Our results indicate that none of the solution concepts dominates another. Closing the gap for PoS savings remains an interesting question. Another open question is to come up with a (generic) sufficient condition for a restaking graph to be secure, that can be checked in a polynomial time, and the condition accepts some graphs with positive restaking savings.

References

[1] Tarun Chitra and Mallesh Pai. How much should you pay for restaking security? *CoRR*, abs/2408.00928, 2024.

[2] Soubhik Deb, Robert Raynor, and Sreeram Kannan. Stakesure: Proof of stake mechanisms with strong cryptoeconomic safety. *CoRR*, abs/2401.05797, 2024.

[3] Naveen Durvasula and Tim Roughgarden. Robust restaking networks. In Raghu Meka, editor, *16th Innovations in Theoretical Computer Science Conference, ITCS 2025, January 7-10, 2025, Columbia University, New York, NY, USA*, volume 325 of *LIPIcs*, pages 48:1–48:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

[4] Cyril Monnet. Rehypothecation. *Business Review, Federal Reserve Bank of Philadelphia*, pages 18–25, 2011.

[5] Abhimanyu Nag, Dhruv Bodani, and Abhishek Kumar. Economic security of multiple shared security protocols. *CoRR*, abs/2505.03843, 2025.

[6] Fahad Saleh. Blockchain without waste: Proof-of-stake. *The Review of Financial Studies*, 34:1156–1190, 2021.

[7] EigenLayer Team. Eigenlayer: The restaking collective, 2023.

[8] Roi Bar Zur and Ittay Eyal. Elastic restaking networks. *CoRR*, abs/2503.00170, 2025.