Committee Selection with Non-Proportional Weights

Yucheng Sun
Department of Computer Science
National University of Singapore
Republic of Singapore
sunyuch@comp.nus.edu.sg

Haifeng Yu
Department of Computer Science
National University of Singapore
Republic of Singapore
haifeng@comp.nus.edu.sg

Ruomu Hou
Department of Computer Science
National University of Singapore
Republic of Singapore
houruomu@gmail.com

Abstract

Committees are extensively used in the designs of various Proof-of-Stake (PoS) blockchains. A committee is simply a randomly selected subset of the parties/nodes in the system. Ideally, the committee should i) be as small as possible, and ii) properly represent the entire system, in terms of the corruption ratio. Existing committee selection schemes all follow the principle of *proportionality*, which says that a committee member should neither over-represent nor under-represent the stake it holds.

In this work, highly surprisingly, we discover that proportionality actually leads to sub-optimal designs. Namely, better security and smaller committee size can be achieved when parties over-represent/under-represent their stakes. We then explore such *non-proportional designs*, and show that they can help to reduce error by many orders of magnitude, under realistic settings and real-world stake distributions of 6 major cryptocurrencies.

CCS Concepts

• **Security and privacy → Security protocols**; **Distributed systems security**.

Keywords

Committee selection, blockchains, Proof-of-Stake, weight assignment

ACM Reference Format:
Yucheng Sun, Haifeng Yu, and Ruomu Hou. 2025. Committee Selection with Non-Proportional Weights. In . ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Committees are extensively used in various Proof-of-Stake (PoS) blockchains [15–17, 25, 26, 29, 30, 33, 42]. A committee is simply a randomly selected subset of the parties/nodes in the system. As some examples, Algorand [26] and GearBox [17] elect a committee to run their byzantine agreement (BA) protocol, while Ouroboros [33] implicitly elects a committee in each epoch to propose blocks, in their longest-chain design.

Background: Security and committee size. When selecting a committee, there are two common goals:

- **Security:** The committee should represent the entire system, in terms of the corruption ratio. Specifically, let $f_{\max}$ (e.g., $f_{\max} = \frac{1}{5}$ in [26]) be the maximum corruption fraction that the adversary can cause in the system. Then "security" of the committee means that the corruption fraction g in the committee should not be much larger than $f_{\max}$, even under the worst-case adversary, except for some negligible probability. This is because committees often run various security protocols, such as BA protocols. If g reaches, for example $\frac{1}{3}$ in [26], the system is no longer secure.
- **Committee size:** The size of the committee should be as small as possible, since the size directly relates to various costs/overheads in the system (see later).

There is a fundamental trade-off between security and committee size. For example, we would have the best security if the committee contains all nodes in the system. But the overheads would be prohibitive. On the other hand, if the committee size is too small, then g can be much larger than $f_{\max}$.

Even moderate reduction of committee size can:

- Significantly improve performance: For example, committees are often used to run BA or BFT (byzantine fault-tolerant) protocols. These protocols usually have quadratic (or even higher) message complexity.[1] Such a quadratic relation with committee size has also been confirmed in real experiments [1]. Given such, even 30% reduction in committee size could potentially lead to over 100% improvement in performance, while 50% reduction could potentially lead to 300% performance improvement.
- Significantly reduce other overheads: For example in Ethereum, transactions are validated by a committee of validators [20]. The gas fees [19] in Ethereum partly serve to compensate the validators for their work. A 30% reduction in committee size would imply a 30% reduction in total validation work, enabling lower gas fees. Such saving could be substantial, given that the magnitude of the gas fee reaches about one billion USD [13, 21, 22] per year.

Background: Committee selection scheme. A *committee selection scheme* conceptually has two related parts: *member selection* and *weight assignment*. *Member selection* determines which parties are selected into the committee, while *weight assignment* decides the weight of a party (once it is selected) in the committee.

[1] For example, PBFT [12] has $O(n^3)$ message complexity, while SBFT [27] and Sync Hotstuff [2] have about $O(n^2)$ message complexity. Some other works [5, 31, 36, 39] have complexity up to $O(n^4)$.

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

As an example, consider the basic committee selection scheme used in a number of PoS blockchains [17, 29, 30, 33, 42]. In a PoS blockchain, different nodes usually have different stakes. Assume that the stake is normalized, so that the total stake in the system is 1. Let s_p be the stake (i.e., amount of cryptocurrency) held by the party p. In this basic scheme [17, 29, 30, 33, 42], to elect one more committee member, the scheme simply chooses a party p with probability exactly s_p. For weight assignment, each chosen party has weight 1 in the committee. (If a party is selected total $x > 1$ times to join the committee, then it has weight x.)

There are also other committee selection schemes, such as those [15, 16, 26] using local randomness, as well as the recent *Fait Accompli* scheme [25]. See Section 3 for a review.

Background: Proportionality. In committee selection, weight assignment typically follows the *principle of proportionality* [25].[2] In fact, *all* schemes that we are aware of follow this principle.

Intuitively, proportionality means that a committee member should neither over-represent nor under-represent the stake it holds. For example, proportionality implies that:

- If $s_p = 2s_q$ and if party p and party q have the same probability of getting into the committee, then p should have twice the weight of q in the committee.
- If $s_p = 2s_q$ and if p already has twice the probability of getting into the committee as compared to q, then they should have the same weight in the committee.

Equation 3 later gives the full definition for *proportionality*, which says that the expectation of a party's weight should always equal its stake [25].

Background: Why proportionality. The adversary can strategically choose which nodes to corrupt. Without proportionality, the adversary can target those nodes who over-represent their stake, and gain undeserving advantage. To understand why, let $f_{\max}$ be the maximum fraction of corrupted stake (i.e., stake held by corrupted nodes) in the system, and imagine an adversary that indeed causes $f_{\max}$ fraction of the stake to be corrupted. Subject to this constraint, the adversary still has the freedom of choosing which subset of the nodes to corrupt. Let g be the fraction of corrupted weight in the committee, under such an adversary.

It is well-known [25] that if we stick to proportionality, then regardless of which subset of the nodes this adversary chooses to corrupt, we always have the expectation of g being equal to $f_{\max}$ (i.e., $\mathrm{E}[g] = f_{\max}$). Without proportionality, the adversary can always corrupt nodes such that $\mathrm{E}[g] > f_{\max}$. Putting it another way, $\mathrm{E}[g]$ is *minimized* iff proportionality is satisfied. Because of this, all existing schemes use proportional weights, and [25] calls such designs as *perfect* designs.

Our work. As the **first contribution** of our work, we discover for the very first time that proportionality leads to sub-optimal designs, which is a highly surprising finding. Namely, using *non-proportional weights*, where parties over-represent/under-represent their stakes, interestingly enables better security and smaller committee size. We also find this phenomenon to be almost universal: Non-proportional weights *always* help, except in some theoretical corner cases. We develop mathematical insights into why this phenomenon arises: Using non-proportional weights does increase $\mathrm{E}[g]$ and make $\mathrm{E}[g] > f_{\max}$, but it also decreases $\mathrm{Var}[g]$. This eventually leads to the improvement.

This unusual finding opens up a whole new direction in committee selection: It tells us that we should consider non-proportional weights, which has *never* been explored before.

As our **second contribution**, we explore the design space of non-proportional weights, to obtain deep theoretical understanding. Exploiting such understanding, we design a novel algorithm for efficiently finding the optimal non-proportional weights (for the worst-case adversary).

Finally, as our **third contribution**, we extensively quantify the practical benefits achieved by non-proportional weights, under the real-world stake distributions of 6 major cryptocurrencies.[3] Under practical settings, our results show that using non-proportional weights can reduce error by many orders of magnitude, given the same committee size. For achieving the same error, using non-proportional weights reduces committee size by up to over 50% (e.g., from about 800 to about 400 in Figure 7(a)). As explained earlier, even 30% reduction in committee size can significantly improve system performance and reduce overheads.

For all these improvements, the "amount" of non-proportionality that we use is actually limited — within a factor of 2 from proportional weights. Hence, even a small "amount" of non-proportionality goes a long way.

No caveats. Our results do not come with any caveat. We use a standard adversarial model, and consider the worst-case adversary. We allow the adversary to target parties who over-represent their stakes.

Fairness. Some blockchains give out rewards to committee members, as incentive [4]. Non-proportional weights are inherently unsuitable for such purpose. Fortunately, a committee can use different weights for different purposes: Non-proportional weights can be used when running for example, the BA protocol, to achieve better security. Simultaneously and independently, traditional proportional weights can be used when determining the reward, to ensure fairness.

Roadmap. Section 2 presents our model. Section 3 reviews existing schemes. Section 4 gives an example of how non-proportional weights can help. Section 5 explores the design space of non-proportional weights. Section 6 shows that non-proportional weights almost always help. Section 7 extends to schemes using local randomness. Section 8 experimentally quantifies the benefits of non-proportional weights. Finally, Section 9 discusses additional related works, and Section 10 draws conclusions.

2 System Model and Problem Formulation

Table 1 summarizes our key notations.

System model. We follow the standard system model for proof-of-stake as in [15, 16, 25, 26, 29, 30, 33, 42], as follows. We use $\mathbb{P}$ to denote the set of all *parties* in the system. We also call a *party* as a *node*, and use these two terms interchangeably. A node may be

[2] Gazi et al. [25] use the term "perfect" instead of "proportionality". Our "proportionality" is the same as their notion of "perfect".

[3] The source code for our implementation is available at [43].

Game: $\texttt{COMMITTEE-SELECT}^{\Pi}_{\mathcal{A}}(f_{\max}, t)$

1) $\mathbb{P}_{\text{corrupt}} \leftarrow \mathcal{A}(\Pi, f_{\max}, t)$
   // $\mathcal{A}$ can corrupt up to $f_{\max}$ stake
2) $\mathbb{C} \leftarrow \Pi(f_{\max}, t)$ // $\mathbb{C}$ is the committee
3) $\mathcal{A}$ wins iff $\frac{W_{\text{corrupt}}(\mathbb{C})}{W_{\text{total}}(\mathbb{C})} \geq t$

corrupted by an adversary – see later. A node $p \in \mathbb{P}$ holds s_p *stake* (e.g., cryptocurrency), where $s_p > 0$. Without loss of generality, we assume the s_p's are normalized so that $\sum_{p\in\mathbb{P}} s_p = 1$. We also refer to all the s_p values as the *stake distribution* $\mathbb{S}$.

This $\mathbb{S}$ is public knowledge, and is seen by all parties as well as the adversary. For example, a snapshot of $\mathbb{S}$ can be obtained from the current (confirmed) blockchain state. The corrupted parties may *redistribute* their stakes among themselves. If such redistribution is done before the snapshot is taken, then $\mathbb{S}$ already captures the outcomes of such redistribution. If it is done after the snapshot is taken, then it has no material effect for us.

Committee selection. A *committee selection scheme* Π selects a weighted set of nodes as the committee $\mathbb{C}$. Each node p in $\mathbb{C}$ is a *committee member*, and has a positive weight w_p. This w_p may be different from s_p. Occasionally, some applications (e.g., threshold signatures) may need w_p to be an integer. There is existing approach [44] for converting real-value weights to integer weights, while keeping the integers small. Hence we do not need to separately consider integer weights.

Define $W_{\text{total}}(\mathbb{C})$ (respectively, $W_{\text{corrupt}}(\mathbb{C})$) to be the total weight of all (respectively, corrupted) committee members in $\mathbb{C}$. If $\mathbb{C}$ is clear from the context, we simply write W_{total} and W_{corrupt}. We say that a committee selection scheme is *normalized* if it always ensures $W_{\text{total}} = 1$.

Adversary model. We consider the standard adversary model [15–17, 25, 26, 29, 30, 33, 42] for committee selection: There is some adversary $\mathcal{A}$ that plays the security game $\texttt{COMMITTEE-SELECT}$, as defined in the above. In the game, the adversary $\mathcal{A}$ first chooses and corrupts an *arbitrary* number of nodes, subject to the constraint that the total stake held by those nodes is *at most* $f_{\max}$ (e.g., $f_{\max} = 0.2$).

We also call $f_{\max}$ as the *budget* of the adversary. A given adversary may or may not use up its budget. Hence we separately use f_{actual} to denote the total stake held by the corrupted nodes under a *given* adversary, where $0 \leq f_{\text{actual}} \leq f_{\max}$. Corrupting more is always beneficial to the adversary — hence the worst-case adversary will have $f_{\text{actual}} = f_{\max}$. Same as prior works [15–17, 25, 26, 29, 30, 33, 42], we assume that $f_{\max}$ is known to the committee selection scheme Π, while f_{actual} is unknown.[4]

Let $\mathbb{P}_{\text{corrupt}}$ denote those corrupted nodes, under the given adversary $\mathcal{A}$. Next, the committee selection scheme Π selects a committee $\mathbb{C}$, without knowing $\mathbb{P}_{\text{corrupt}}$. The adversary $\mathcal{A}$ wins the game if $\frac{W_{\text{corrupt}}(\mathbb{C})}{W_{\text{total}}(\mathbb{C})} \geq t$. The value t here depends on what the committee is used for. For example, if it is used to run a byzantine agreement protocol, then t is typically either $\frac{1}{3}$ or $\frac{1}{2}$, since the system is no longer secure when $\frac{W_{\text{corrupt}}(\mathbb{C})}{W_{\text{total}}(\mathbb{C})}$ reaches $\frac{1}{3}$ or $\frac{1}{2}$. Same as in prior works [15–17, 25, 26, 29, 30, 33, 42], we assume $f_{\max} < t$ so that the system can provide meaningful guarantees.

p	a party/node
s_p	stake held by p
w_p	p's weight in the committee
$f_{\max}$	the *maximum* total malicious stake in the system (which is a constraint on the adversary), where $0 < f_{\max} < t < 1$
f_{actual}	the *actual* malicious stake in the system, under a given adversary, where $0 \leq f_{\text{actual}} \leq f_{\max}$
f_1	the *actual* stake held by corrupted large nodes, under a given adversary
f_2	the *actual* stake held by corrupted small nodes, under a given adversary
W_{corrupt}	weight of corrupted committee members
W_{total}	weight of all committee members
S_1	stake held by all large nodes
S_2	stake held by all small nodes
t	security threshold, where $0 < f_{\max} < t < 1$
$\mathcal{A}$	adversary attacking committee selection schemes
L, L_β	a TPL scheme
h, k	parameters in FA/bipartition/TPL/general scheme
m	committee size
$\mathbb{S}$	stake distribution

Table 1: Key notations.

Note that in this game, the adversary $\mathcal{A}$ decides which nodes to corrupt, *before* the committee selection scheme Π selects the committee. Such formulation is standard, and is the same as in prior works [15–17, 25, 26, 29, 30, 33, 42]. Fundamentally, if the adversary could adaptively corrupt the committee members after the committee is selected, then no committee would be secure. In practice, this is usually prevented for example, by using *verifiable random functions* (*VRFs*) [38] as the randomness in committee selection [16, 26], or by using each committee only once and quickly re-electing new committees [17, 29, 30, 33, 42].

Defining error. Given a stake distribution $\mathbb{S}$, the *error of* Π *under* $\mathcal{A}$ is:

$$error(\Pi, \mathcal{A}) \quad = \quad \Pr[\mathcal{A} \text{ wins the game}] = \Pr\left[\frac{W_{\text{corrupt}}}{W_{\text{total}}} \geq t\right] \qquad (1)$$

The *error* of the committee selection scheme Π is:

$$error(\Pi) \quad = \quad \max_{\mathcal{A}}(error(\Pi, \mathcal{A})) \qquad (2)$$

Namely, this is defined over the worst-case adversary. We usually want the error to be negligibly small.

3 Existing Designs for Committee Selection

This section reviews existing designs for committee selection, as well as the notion of proportionality.

[4] Some prior works *implicitly* assume the knowledge of $f_{\max}$, without explicitly stating that. For example, the committee size in [25] critically depends on their ϵ parameter, where smaller ϵ leads to larger committee size. Properly choosing ϵ requires knowledge of $f_{\max}$: For instance, to use their design in Algorand [26] whose tolerance is $\frac{1}{3}$, their work [25] explains that $f_{\text{actual}} + \epsilon < \frac{1}{3}$ needs to always hold. Since f_{actual} is unknown, one has to consider the worst-case of $f_{\text{actual}} = f_{\max}$. Hence an appropriate ϵ should be roughly $\frac{1}{3} - f_{\max}$, and $f_{\max}$ is needed for properly choosing ϵ.

A committee selection scheme may either use *public randomness* or *local randomness*. With *public randomness*, a party uses the randomness to determine all the members in the committee, based on the stake distribution. The randomness used by different parties is the same (hence *public*). For example, the schemes in [17, 29, 30, 33, 42] all use public randomness. This public randomness is typically provided by the blockchain itself, for example, via beacon generation [30, 33].

With *local randomness*, each party uses local randomness to determine whether itself is a committee member. In order for such local randomness to be verifiable by others, it is often generated via a *verifiable random function* (*VRF*) [38]. For example, the schemes in [16, 26] all use local randomness generated from VRF.

Our work applies to both kinds. To facilitate understanding, we first focus on committee selection using public randomness. Section 7 later will extend to local randomness.

3.1 Review of Existing Schemes (with Public Randomness)

Basic scheme [17, 29, 30, 33, 42]. In this *basic scheme*, to select a committee member, each node p is chosen with probability s_p. This process is then repeated m times, independently and with replacement, to get a committee of size m. Each time a node gets selected, it gets $\frac{1}{m}$ weight. (Here we use weight $\frac{1}{m}$, to make the scheme normalized.) If a node is selected x times in total, then its weight will be $\frac{x}{m}$.

Fait Accompli (FA) scheme [25]. Figure 1 illustrates the *Fait Accompli scheme* (or *FA scheme*) from [25]. This scheme first chooses some threshold h (e.g., $h = \Theta(\frac{1}{m})$ where m is the target committee size). A node p is called a *large-stake node* (or *large node* in short) if $s_p > h$, otherwise it is a *small-stake node* (or *small node*). A large node p is always in the committee, with a weight of $w_p = s_p$. The rough intuition [25] here is that a large node p would likely be in the committee anyway, even if we chose committee members randomly.

Besides the large nodes, the FA scheme [25] chooses additional $k \geq 1$ committee members from the small nodes: To select one more committee member, a small node p with s_p stake is chosen with probability $\frac{s_p}{S_2}$. Here S_2 (respectively, S_1) is the total stake of the small (respectively, large) nodes,[5] where $S_1 + S_2 = 1$. For example in Figure 1, the small node with 0.1 stake has $\frac{0.1}{0.35}$ probability of being chosen. Each time a small node p is chosen, its weight increases by $\frac{S_2}{k}$, or $\frac{0.35}{2}$ in the example in Figure 1. (If p is chosen x times, it has weight $w_p = \frac{xS_2}{k}$.) One can verify that the total weight of all committee members is always $S_1 + k \cdot \frac{S_2}{k} = 1$. Hence the FA scheme is a normalized scheme. The final committee size will be k plus the total number of large nodes.

It has been shown [25] that the FA scheme significantly outperforms the basic scheme, under real-world stake distributions.

Bipartition scheme. As a trivial generalization of all the above schemes, we define the *bipartition scheme*. Note that we still view the bipartition scheme as an *existing* scheme.

[5]The scheme requires S_2 to be positive, since otherwise all nodes are large nodes and the committee would contain all nodes.

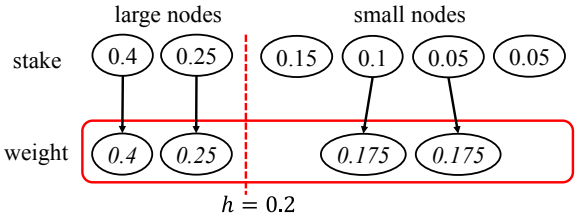


Figure 1: Illustrating the FA scheme [25], under $h = 0.2$ and $k = 2$. Here the two small nodes with stake 0.1 and 0.05 happen to be chosen to join the committee. Each of them has a weight of $\frac{S_2}{k} = \frac{0.35}{2} = 0.175$ in the committee. The other two small nodes, with stake 0.15 and 0.05 respectively, happen not to be elected into the committee.

A bipartition scheme works in the same way as the FA scheme, except that the threshold h is now adjustable and is viewed as a parameter. With $h = 1$, the bipartition scheme degrades to the basic scheme [17, 29, 30, 33, 42]. With h being chosen as in [25], it becomes the FA scheme. We further allow other h values: In fact, we will treat the bipartition scheme (with the best h) as the state-of-the-art existing scheme — doing so only makes the existing results better.

3.2 Review of the Principle of Proportionality

Proportionality. Recall that w_p is the weight of node p, if it is in the committee. For convenience, we define $w_p = 0$ if p is not in the committee. Since w_p might depend on how many times (or whether) p is selected to join the committee, it is a random variable. Without loss of generality, assume that the committee selection scheme is normalized. (If not, one could always normalize it first.)

Formally, *proportionality* [25] requires that for all node p:

$$\mathrm{E}[w_p] \quad = \quad s_p \tag{3}$$

If a committee selection scheme (after normalization) guarantees Equation 3, we say that it is *proportional* and uses *proportional weights*. Otherwise it is *non-proportional* and uses *non-proportional weights*.

3.3 All Existing Schemes are Proportional

One can verify that all existing committee selection schemes use proportional weights. In the basic scheme [17, 29, 30, 33, 42], by linearity of expectation, we have $\mathrm{E}[w_p] = \frac{s_p}{m} \times m = s_p$. In the FA scheme [25], for a large node p, we directly have $\mathrm{E}[w_p] = s_p$. For a small node p, by linearity of expectation, we have $\mathrm{E}[w_p] = (\frac{s_p}{S_2} \cdot \frac{S_2}{k}) \times k = s_p$. Similarly, the bipartition scheme also uses proportional weights.

3.4 Focus of Our Work

Our work will focus on weight assignment. We use the same member selection process as the bipartition scheme. But our weights will be different, and will be non-proportional.

4 Proportionality Leads to Sub-optimal Designs

This section shows, via a concrete example, that proportionality can lead to sub-optimal designs, and that using non-proportional

weights enables better security and smaller committee size. *Note that discovering this surprising phenomenon, by itself, is already a key contribution of our work, since the phenomenon is rather surprising and counter-intuitive.*

We will later generalize from this particular example, showing that it is not a corner case. In fact, this surprising phenomenon is almost *universal*.

4.1 A Concrete Example

State-of-the-art existing scheme. Recall from Section 3.1 the bipartition scheme, which is the state-of-the-art committee selection scheme and uses **proportional** weights. Figure 2 shows an example bipartition scheme Prop.

Example scheme using non-proportional weights. We now obtain an example **non-proportional** scheme NProp, by adjusting the weights in Prop. Specifically, let w_p be the weight of a committee member p, in the scheme Prop. Recall from Section 3.1 that in this bipartition scheme Prop, each p is either a small node or a large node. Then in NProp, the weight of p is:

- slightly decreased to be $w'_p = 0.9w_p$, if p is a small node, and
- slightly increased to be $w'_p = 1.093w_p$, if p is a large node.

Here "0.9" is simply an example value that deviates from 1, while "1.093" is calculated such that scheme NProp remains normalized under the Bitcoin stake distribution, given the first constant "0.9".

The weights in NProp are non-proportional, and do not satisfy Equation 3. Intuitively, here a large node slightly over-represents its stake, while a small-node slightly under-represents its stake. This also means that the adversary might choose to target the large nodes, in order to corrupt more weights in the committee.

Different adversaries. We next aim to compare the errors of Prop and NProp. The error of a scheme differs under different adversaries, and we need to consider the error under the worst-case adversary.

To capture the worst-case adversary, let f_1 (respectively, f_2) be the total stake of large (respectively, small) nodes that are corrupted by a given adversary. Obviously $f_1 + f_2 \le f_{\max}$. Since it is always better for the adversary to corrupt more nodes, for the worst-case adversary, we only need to consider $f_2 = f_{\max} - f_1$. One can see that for both Prop and NProp, their error is uniquely determined by the value f_1:

- Given f_1, it does not matter which subset of the large nodes get corrupted, since the total weight of the corrupted large-node committee members will always be f_1.
- Given $f_2 = f_{\max} - f_1$, it does not matter which subset of the small nodes get corrupted. This is because the total weight of the corrupted small-node committee members always follows the distribution $\frac{S_2}{k} \times \mathbf{Binom}(k, \frac{f_2}{S_2})$, where $\mathbf{Binom}()$ is the binomial distribution. Namely, we select small-node committee members total k times. For each selection, there is $\frac{f_2}{S_2}$ probability of selecting a corrupted small node. The weight of each selection is $\frac{S_2}{k}$.

Hence to capture the error under the worst-case adversary, we just need to consider the errors under all f_1 values, and then take the peak error.

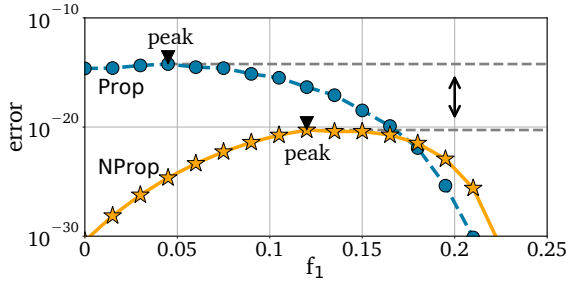


Figure 2: Scheme Prop **is an example bipartition scheme (which uses proportional weights) under** $f_{\max} = \frac{1}{4}$, $t = \frac{1}{3}$, $f_2 = f_{\max} - f_1$, $h = 0.00025$ **and** $k = 500$**, and the Bitcoin stake distribution [8]. Scheme** NProp **is an example non-proportional scheme. The figure shows that** NProp **outperforms** Prop**. Here the example values of** $f_{\max} = \frac{1}{4}$ **and** $t = \frac{1}{3}$ **are from [45] and [17, 26, 42], respectively. Note that the value of** t **comes from the fundamental requirement of the application that uses the committee, while the value of** $f_{\max}$ **is what the application chooses to tolerate. Hence they are not controlled by the adversary.**

Comparing the errors of NProp **and** Prop**.** Figure 2 now compares the errors of the two schemes, under various f_1 values. It shows that Prop gets the peak error of about 10^{-14}, when $f_1 \approx 0.045$. The error of Prop in non-peak regions is generally much smaller. But since we have to consider the worst-case adversary, it is the peak error that counts.

In contrast, NProp has a (peak) error of only about 10^{-20}, which is much smaller. Note that peak errors of Prop and NProp occur at different f_1 values. This is expected, since the worst-case adversaries against different schemes are likely different.

Figure 2 is *not* a cherry-picked corner case. One may wonder whether the results in Figure 2 are just some corner case phenomenon that arises only under the specific settings there (e.g., the specific values of $f_{\max}$, t, and so on). The answer is a categorical "no":

- Section 8 will present extensive experimental results under a wide variety of other settings (e.g., different $f_{\max}$ and t values) and the real-world stake distributions of 6 major cryptocurrencies. We observe the same phenomenon there: Using non-proportional weights *consistently* helps, across different settings and stake distributions.
- Section 5 will formally *prove* that this phenomenon broadly exists and is almost *universal*: Namely, non-proportional weights will outperform proportional weights, in *all* cases except for some theoretical corner cases.

4.2 Deeper Understanding of the Example

We next aim to get some deeper understanding, on why NProp can beat Prop in Figure 2. In particular, Figure 2 shows that NProp has smaller error than Prop when $f_1 < 0.18$, and larger error than Prop when $f_1 > 0.18$. We want to intuitively understand why this happens.

Both NProp and Prop are normalized. Hence the error is simply $\Pr[W_{\text{corrupt}} \geq t]$ under the worst-case f_1. Recall that NProp is obtained from Prop, by doing some *weight adjustments*, which slightly decrease (increase) the weights of the small (large) nodes. The effect of such weight adjustments on error (i.e., $\Pr[W_{\text{corrupt}} \geq t]$), roughly speaking, comes from their effects on $\mathrm{E}[W_{\text{corrupt}}]$ and $\mathrm{Var}[W_{\text{corrupt}}]$. Intuitively, smaller $\mathrm{E}[W_{\text{corrupt}}]$ and smaller $\mathrm{Var}[W_{\text{corrupt}}]$ will lead to small error $\Pr[W_{\text{corrupt}} \geq t]$.

Appendix A will show that NProp's weight adjustments always decrease $\mathrm{Var}[W_{\text{corrupt}}]$, regardless of f_1. Appendix A will also show that those adjustments will i) decrease $\mathrm{E}[W_{\text{corrupt}}]$ when f_1 is small, and ii) increase $\mathrm{E}[W_{\text{corrupt}}]$ when f_1 is large. Furthermore, the larger f_1 is, the larger the amount of increase will be caused by those weight adjustments. In Figure 2, NProp and Prop intersect when $f_1 \approx 0.18$. This is where the effect of the increase in $\mathrm{E}[W_{\text{corrupt}}]$ and the effect of the decrease in $\mathrm{Var}[W_{\text{corrupt}}]$ cancel out. Hence scheme NProp gives smaller error than Prop when $f_1 < 0.18$, and larger error than Prop when $f_1 > 0.18$. But since only the peak error matters, the error of NProp under the worst-case f_1 is still better than Prop under the worst-case f_1.

5 Exploring Non-Proportional Schemes

Section 4 showed the promise of non-proportional weights. This opens up a whole new direction in committee selection: It tells us that we should consider non-proportional weights, which has *never* been explored before. Given such, as the second key contribution of our work, this section explores the design space, to find the best non-proportional weights.

Recall from Section 2 that f_{max} is known to the committee selection scheme, while f_{actual} is not. For finding the best design, we will consider $f_{\text{actual}} = f_{\text{max}}$. This is needed since we have to guard against the worst-case adversary with $f_{\text{actual}} = f_{\text{max}}$. Hence the various optimality claims in this section will be for this worst-case. If f_{actual} turns out to be smaller than f_{max}, for example under some non-worst-case adversary, then the committee we find may be larger than necessary (but will still be secure). Our approach here is similar to prior works [15–17, 25, 26, 29, 30, 33, 42] on committee selection. In particular, note that without knowing f_{actual}, fundamentally it is not possible to select a committee that is optimal for all f_{actual} values. In fact when $f_{\text{actual}} = 0$, the optimal committee should have a size of 1. But such a committee obviously is insecure when f_{actual} is larger (but still below f_{max}).

5.1 General Form of Non-proportional Weights

This section defines the general form of committee selection schemes that use non-proportional weights. Recall from Section 3.1 that the bipartition scheme captures *all* existing committee selection schemes. In the bipartition scheme, for every node p:

- If node p is a large node, it gets a weight of s_p in the committee.
- If node p is a small node, it gets a weight of $\frac{S_2}{k}$ for each time it is selected into the committee.

Now to explore non-proportional weights, we generalize how weight is assigned in the bipartition scheme. (The member selection part in the bipartition scheme is not affected.) To this end, we define the *general scheme* Π_{σ_1,σ_2} to be the same as the bipartition scheme except that:

- If node p is a large node, it gets a weight of $\sigma_1(s_p)$ in the committee.
- If node p is a small node, it gets a weight of $\sigma_2(s_p)$ for each time it is selected into the committee. (Hence if selected x times, p will get total weight $x \cdot \sigma_2(s_p)$.)

Here $\sigma_1()$ and $\sigma_2()$ are arbitrary functions of s_p, allowing us to freely deviate from proportionality.

Examples of $\sigma_1()$ and $\sigma_2()$. The bipartition scheme is simply Π_{σ_1,σ_2} with $\sigma_1(s_p) = s_p$ and $\sigma_2(s_p) = \frac{S_2}{k}$. The non-proportional scheme NProp in Section 4.1 is a general scheme Π_{σ_1,σ_2} with $\sigma_1(s_p) = 1.093 s_p$ and $\sigma_2(s_p) = 0.9\frac{S_2}{k}$. Another example non-proportional scheme would be Π_{σ_1,σ_2} with $\sigma_1(s_p) = \sqrt{s_p}$ and $\sigma_2(s_p) = (s_p \cdot \frac{S_2}{k})^2$.

5.2 Searching for Best Non-proportional Weights: Overview and Intuitions

Our goal is to find the best general scheme, or equivalently, the best $\sigma_1()$ and $\sigma_2()$. This is challenging since $\sigma_1()$ and $\sigma_2()$ can be arbitrary functions. At a high level, we first determine the best "form" of $\sigma_1()$ and $\sigma_2()$, and then find the best "parameters" in that form.

Best "form" under given f_1 and f_2. To help understanding, let us start with a simplified case with *fixed* f_1 and f_2 values, where $f_1 + f_2 = f_{\text{max}}$. Imagine an adversary such that the total stake of corrupted large (respectively, small) nodes is f_1 (respectively, f_2). Except for the constraints of f_1 and f_2, the adversary has not yet decided exactly which large nodes (or small) nodes to corrupt.

Under such given f_1 and f_2, we can prove (in Lemma 6) that the best $\sigma_1()$ and $\sigma_2()$ must be of *two-piece-linear-form*. Here *two-piece-linear-form* means i) $\frac{\sigma_1(s_p)}{s_p}$ is a constant independent of s_p, and ii) $\sigma_2(s_p)$ is a constant indepedent of s_p. Intuitively, the two-piece-linear-form ensures that the adversary will always get the same "reward-cost-ratio", regardless of which large (small) nodes it chooses to corrupt, under the given f_1 (f_2) value:

- Imagine that the adversary corrupts a large node with s_p stake, by spending (as its "cost") s_p out of the budget f_1. As its "reward", the adversary will get $\sigma_1(s_p)$ corrupted weight in the committee. If $\frac{\sigma_1(s_p)}{s_p}$ is a constant, then the "reward-cost-ratio" will be the same for different large nodes with different s_p values. Intuitively, this prevents the adversary from gaining extra advantage by targeting nodes with higher "reward-cost-ratios".
- Imagine that the adversary spends s_p out of the budget f_2 to corrupt a small node with s_p stake. Then each time we choose a committee member from the small nodes, as the adversary's "reward", with probability $\frac{s_p}{S_2}$ the adversary gets $\sigma_2(s_p)$ corrupted weight in the committee.
  Here note that the probability $\frac{s_p}{S_2}$ *already* grows linearly with the "cost" s_p. Hence to make the "reward-cost-ratio" of corrupting different small nodes to be the same, intuitively, we need $\sigma_2(s_p)$ to be a constant independent of s_p.

general scheme Π_{σ_1,σ_2}	this work	non-proportional weight	—
TPL scheme L, L_β	this work	non-proportional weight	special case of general scheme
bipartition scheme	prior work	proportional weight	special case of TPL scheme
Fait Accompli (FA) scheme [25]	prior work	proportional weight	special case of bipartition scheme
basic scheme [17, 29, 30, 33, 42]	prior work	proportional weight	special case of bipartition scheme

Table 2: Summary of committee selection schemes.

Formalizing the above intuitions, however, will be subtle, and our actual proofs will need a careful coupling of the randomness used in committee selection.

Best "form" under worst-case adversary. The above discussion is for the simplified case with *given* f_1 and f_2 values. For our actual result, we need to consider the worst-case adversary with the worst-case f_1 value. (For the worst-case adversary, f_2 will be $f_2 = \min(f_{\max} - f_1, S_2)$.[6] Hence we do not need to additionally mention the value of f_2.)

The tricky issue here is that the worst-case f_1 depends on $\sigma_1()$ and $\sigma_2()$. Hence when we change $\sigma_1()$ or $\sigma_2()$, the worst-case f_1 value also changes. Putting it another way, the peak error of different committee selection schemes occurs at different f_1 values. Nevertheless, our Theorem 5 later will show that the best $\sigma_1()$ and $\sigma_2()$ must still be of two-piece-linear-form, when we consider the worst-case adversary with the worst-case f_1 value.[7] The proof of Theorem 5 is rather involved, but at a high level, Theorem 5 proves this by showing:

- The optimal general scheme (with the optimal $\sigma_1()$ and $\sigma_2()$) must have an error of at least err^*, under the worst-case adversary. Here err^* is as defined in Equation 4 later.
- There exist some two-piece-linear-form $\sigma_1()$ and $\sigma_2()$ such that the resulting scheme has error of err^*, under the worst-case adversary.

Finding optimal "parameters". After knowing that the best $\sigma_1()$ and $\sigma_2()$ must be of two-piece-linear-form, we now only need to find the optimal "parameters" (i.e., the constants) in the two-piece-linear-form. Doing this is non-trivial, since the parameters are real-values, making exhaustive enumeration impossible. We will develop a novel algorithm for searching for such optimal parameters.

Roadmap. Our overview in Section 5.2 has been **top-down**. For rigorous reasoning, however, Section 5.3 through 5.5 need to discuss the details in a **bottom-up** fashion:

- Section 5.3 defines *Two-Piece-Linear* (*TPL*) schemes. A TPL scheme is essentially Π_{σ_1,σ_2} where $\sigma_1()$ and $\sigma_2()$ are of two-piece-linear-form. We then prove several key properties of TPL schemes.
- Section 5.4 designs a novel algorithm for finding the optimal "parameters" in TPL scheme.
- Section 5.5 proves that the best $\sigma_1()$ and $\sigma_2()$ must be of two-piece-linear-form. This means that the best general scheme must be a TPL scheme.

[6] By definition, f_2 cannot exceed the total stake held by the small nodes, which is S_2.
[7] If there are multiple optimal general schemes (with different $\sigma_1()$'s and $\sigma_2()$'s) that have the same error, then we will prove that at least one of them must have two-piece-linear-form $\sigma_1()$ and $\sigma_2()$.

5.3 TPL Schemes and Their Properties

For clarity, Table 2 summarizes the various committee selection schemes.

TPL schemes. Intuitively, a TPL scheme is a general scheme Π_{σ_1,σ_2} where $\sigma_1()$ and $\sigma_2()$ are of two-piece-linear-form. Formally, a *Two-Piece-Linear scheme* (or *TPL scheme*) has a parameter $\beta \geq 0$, and is denoted as L_β. The scheme L_β is defined to be the general scheme Π_{σ_1,σ_2} with:

- $\sigma_1(s_p) = \alpha \cdot s_p$, where $\alpha = \frac{1-\beta \cdot S_2}{S_1}$
- $\sigma_2(s_p) = \beta \cdot \frac{S_2}{k}$

Why having the $\frac{S_2}{k}$ term. One may wonder why we require $\sigma_2(s_p) = \beta \cdot \frac{S_2}{k}$, instead of $\sigma_2(s_p) = \beta$. Actually both versions would work. But our version facilitates discussion, as follows.

With our definition, the bipartition scheme is simply the TPL scheme L_1, namely, the TPL scheme with $\alpha = \beta = 1$. Compared to the bipartition scheme, the TPL scheme L_β:

- multiplies weight of large-node committee member by α, and
- multiplies weight of small-node committee member by β.

With $\beta \neq 1$, a TPL scheme uses non-proportional weights. The scheme NProp in Figure 2 is essentially the TPL scheme $\mathsf{L}_{0.9}$.

Why α is not a parameter. One may wonder why a TPL scheme does not have both α and β as parameters. This is simply because we want a TPL scheme to always be normalized. With such normalization, α and β cannot both be "free". Our definition here chooses to treat β as "free", and α as a function of β.[8]

Properties of TPL schemes. TPL schemes have strong mathematical properties. We will need such properties in later proofs. Specifically, we will prove for TPL scheme L_β:

- Decreasing β will decrease the error under $f_1 < tS_1$.
- Decreasing β will not affect the error under $f_1 = tS_1$.
- Decreasing β will increase the error under $f_1 > tS_1$.

Figure 3 demonstrates these properties. In particular, Figure 3 illustrates the **pivot point**, through which every TPL scheme must pass. This pivot point exists because, as we prove later, changing β has no impact on the error of a TPL scheme when $f_1 = tS_1$. We will further prove (see Theorem 2) that this pivot point is at (tS_1, err^*), with:

$$err^* = \Pr[\mathbf{Binom}(k, \frac{f_{\max} - tS_1}{S_2}) \geq kt] \quad (4)$$

[8] As a technicality, when $S_1 = 0$, the value of α is not well-defined. But when $S_1 = 0$, there are no large nodes, and hence the value of α is irrelevant. Also, when $S_1 = 0$, we require $\beta = 1$, since this is the only β value that can make the scheme normalized.

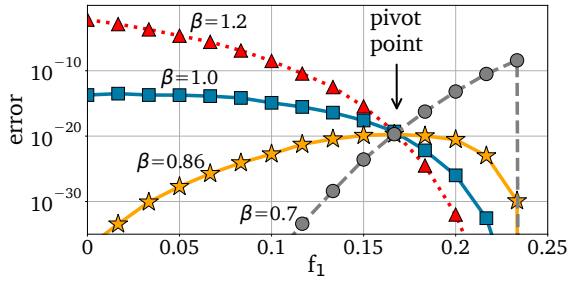


Figure 3: How β affects the error of a TPL scheme, under different f_1 values. Here tS_1 is roughly 0.17. These results are under $f_{\max} = \frac{1}{4}$, $t = \frac{1}{3}$, $h = 0.0025$, $k = 500$, and the Ethereum stake distribution [23].

Formal theorems. We next prove these properties. Define:

$$\mathbb{A}_{f_1,f_2} = \{\mathcal{A} \mid \text{under adversary } \mathcal{A}, \text{ the stake of corrupted large (small) nodes is } f_1 \ (f_2)\} \quad (5)$$

$$error(\Pi, \mathbb{A}_{f_1,f_2}) = \max_{\mathcal{A}\in\mathbb{A}_{f_1,f_2}} (error(\Pi, \mathcal{A})) \quad (6)$$

Obviously, we have $error(\Pi) = \max_{f_1+f_2\le f_{\max}} (error(\Pi, \mathbb{A}_{f_1,f_2}))$.

Theorem 1. *Consider any $0 \le f_1 < tS_1$ and $0 \le f_2 \le S_2$. Then for all $0 \le \beta_1 \le \beta_2$, we have $error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) \le error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$.*

Proof. Since $0 \le f_1 < tS_1$, we must have $S_1 > 0$. Now consider any TPL scheme L_β where $\beta > 0$. For all adversaries in $\mathbb{A}_{f_1,f_2}$, the total stake of corrupted large nodes is f_1. Hence the total weight of corrupted large-node committee members is $f_1 \cdot \alpha$. Next, for all adversaries in $\mathbb{A}_{f_1,f_2}$, the total stake of corrupted small nodes is f_2. This means that each time the scheme chooses a small-node committee member, there is $\frac{f_2}{S_2}$ probability of that member being corrupted. Recall that we choose small-node committee members total k times. Hence the total weight of corrupted small-node committee members is $\mathbf{Binom}(k, \frac{f_2}{S_2}) \cdot \frac{\beta S_2}{k}$, where $\mathbf{Binom}()$ is the binomial distribution. Putting all these together gives $W_{\text{corrupt}} = f_1 \cdot \alpha + \mathbf{Binom}(k, \frac{f_2}{S_2}) \cdot \frac{\beta S_2}{k}$.

We then have $error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2}) = \Pr[W_{\text{corrupt}}/W_{\text{total}} \ge t] = \Pr[f_1 \cdot \alpha + \mathbf{Binom}(k, \frac{f_2}{S_2}) \cdot \frac{\beta S_2}{k} \ge t] = \Pr[\mathbf{Binom}(k, \frac{f_2}{S_2}) \ge \frac{t - f_1 \cdot \alpha}{\beta S_2} \cdot k]$, which in turn gives:

$$error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2}) = \Pr[\mathbf{Binom}(k, \tfrac{f_2}{S_2}) \ge k(\tfrac{tS_1 - f_1}{\beta S_2 S_1} + \tfrac{f_1}{S_1})] \quad (7)$$

Since $f_1 < tS_1$, when β decreases, the term $\frac{tS_1 - f_1}{\beta S_2 S_1}$ increases and thus $error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2})$ never increases. This implies $error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) \le error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$, if $0 < \beta_1 \le \beta_2$. Finally, if $\beta_1 = 0$, then we trivially have $error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) = \Pr[f_1 \cdot \alpha \ge t] = \Pr[f_1 \cdot \frac{1}{S_1} \ge t] = 0 \le error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$. □

We defer to Appendix B and C the proofs of the following two theorems, which are similar to the proof of Theorem 1.

Theorem 2. *Consider $f_1 = tS_1$ and any $0 \le f_2 \le S_2$. Then for all $\beta_1 > 0$ and $\beta_2 > 0$, we have $error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) = error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2}) = \Pr[\mathbf{Binom}(k, \frac{f_2}{S_2}) \ge kt]$.*

Theorem 3. *Consider any $tS_1 \le f_1 \le S_1$ and $0 \le f_2 \le S_2$. Then for all $0 \le \beta_1 \le \beta_2$, we have $error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) \ge error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$.*

5.4 Finding Optimal Parameters for TPL Scheme

This section develops a novel algorithm (Algorithm 1) for finding the optimal parameters for TPL scheme. A TPL scheme has an explicit parameter β. Furthermore, a TPL scheme implicitly inherits the parameters h and k from the bipartition scheme. The values of m, $f_{\max}$, t, and $\mathbb{S}$ are inputs. Hence the goal of Algorithm 1 is, for a given $(m, f_{\max}, t, \mathbb{S})$ tuple, to find the optimal (h, k, β) tuple that minimizes the error of the resulting TPL scheme.

In Algorithm 1 at Line 1, we exhaustively test all possible h values so that we can eventually pick the best one. Such exhaustive test is possible because $\mathbb{S}$ has total $|\mathbb{P}|$ nodes, and hence we only need to enumerate $|\mathbb{P}| + 1$ different values for h. Next, the value k is uniquely determined by h (see Line 2), since k must equal $m - l$, where l is the number of large nodes based on h and $\mathbb{S}$.

Central step: Search for best β. Under fixed h and k values, Algorithm 1 invokes Algorithm 2 to find the best β value. This is the central step in our algorithm. This step is challenging, because β is a real value and cannot be exhaustively enumerated. A binary search is not possible since the relation between β and $error(\mathsf{L}_\beta)$ might not be monotonic. One could do a brute-force search, but it is unclear what would be suitable granularity: Too coarse granularity will result in poor results, while too fine granularity makes it computationally expensive.

Fortunately, by leveraging Theorem 1 and 3, we do not have to resort to brute-force: While the relation between β and $error(\mathsf{L}_\beta)$ is not monotonic overall, Theorem 1 shows that the relation is monotonic under $f_1 < tS_1$. Similarly, Theorem 3 shows a monotonic relation under $f_1 \ge tS_1$. Hence we can do binary search for the two regions separately. (Note that for given $f_{\max}$ and f_1, we need to consider the worst-case f_2, which is $f_2 = \min(f_{\max} - f_1, S_2)$. Hence f_1 is the only free variable.)

Based on this idea, Algorithm 2 finds the optimal β, via 3 binary searches. It first does a binary search over $[0, 1]$, with the goal of finding the minimum error achievable. For each value x tested in this binary search, Algorithm 2 (Line 8 through Line 13) needs to determine whether there exists any L_β such that $error(\mathsf{L}_\beta) \le x$.

Directly making this determination can be difficult. Instead, Algorithm 2 first finds β_1, where β_1 is the maximum β value such that L_β has error at most x for all $f_1 < tS_1$ (see Figure 4). The monotonicity proved by Theorem 1 will enable us to simply use a binary search to find β_1. Specifically, our goal here is find the maximum β value such that the *peak* error of L_β is at most x for $f_1 \in [0, tS_1)$. Our binary search relies on the monotonicity of the peak error with respect to β: Consider β and β' where $\beta < \beta'$. Imagine that the peak error of L_β (when $f_1 \in [0, tS_1)$) is achieved when $f_1 = z$. Similarly, imagine that the peak error of $\mathsf{L}_{\beta'}$ is achieved when $f_1 = z'$. Then the peak error L_β (achieved at $f_1 = z$) must be no larger than the peak error of $\mathsf{L}_{\beta'}$ (achieved at $f_1 = z'$), because:

- By Theorem 1, when $f_1 = z$, the error of L_β is no larger than the error of $\mathsf{L}_{\beta'}$.
- By definition of peak error, the error of $\mathsf{L}_{\beta'}$ at $f_1 = z$ is no larger than the error of $\mathsf{L}_{\beta'}$ at $f_1 = z'$.

We have explained how to find β_1. Next, similarly, define β_2 to be the the minimum β value, such that L_β has error at most x for

Algorithm 1: SearchTPL($m, f_{\max}, t, \mathbb{S}$)

1) enumerate all possible h values based on the stake distribution $\mathbb{S}$, where for each h value:
2) $\quad k \leftarrow m - l$, where l is the number of large nodes based on h and $\mathbb{S}$;
3) $\quad \langle$the minimum error achievable, the corresponding $\beta\rangle \leftarrow$ SearchTPLBeta($h, k, m, f_{\max}, t, \mathbb{S}$);
4) **return** the minimum error achievable across all h values, as well as the corresponding (h, k, β) tuple;

Algorithm 2: SeachTPLBeta($h, k, m, f_{\max}, t, \mathbb{S}$)

5) calculate S_1 and S_2 based on h and $\mathbb{S}$; **if** ($S_1 = 0$) **return** $error(\mathsf{L}_1)$ and $\beta = 1$;
6) do **binary search** over $[0, 1]$ (the goal is to find the minimum $x \in [0, 1]$ where there exists some TPL scheme with error $\le x$):
7) $\quad$ **foreach** x value to be checked in this binary search:
8) $\quad\quad$ define β_1 to be the maximum β such that L_β has error at most x for all $f_1 < tS_1$;
9) $\quad\quad$ find β_1 via a **binary search** over the range $[0, \frac{1}{S_2}]$; // β cannot exceed $\frac{1}{S_2}$ (otherwise α would be negative)
10) $\quad\quad$ define β_2 to be the minimum β such that L_β has error at most x for all $f_1 \ge tS_1$;
11) $\quad\quad$ **if** $f_{\max} \ge tS_1$ **then** find β_2 via a **binary search** over the range $[0, \frac{1}{S_2}]$; **else** $\beta_2 \leftarrow 0$;
12) $\quad\quad$ **if** $\beta_2 \le \beta_1$ **then** all L_β with $\beta \in [\beta_2, \beta_1]$ can achieve error at most x;
13) $\quad\quad$ **if** $\beta_2 > \beta_1$ **then** no TPL scheme can achieve error at most x;
14) $\quad$ **endforeach**
15) **endbinarysearch**
16) **return** $\langle$the minimum error x found by the binary search, some arbitrary $\beta \in [\beta_2, \beta_1]$ corresponding to $x\rangle$;

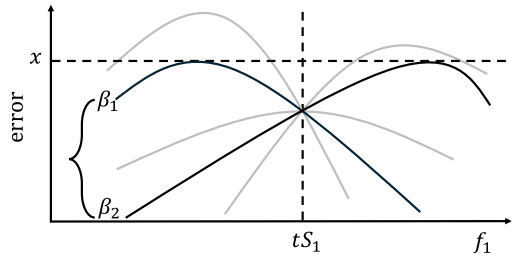


Figure 4: This conceptual figure illustrates β_1 and β_2.

all $f_1 \ge tS_1$ (see Figure 4). By exploiting Theorem 3, Line 11 again uses a binary search[9] to find β_2.

After finding β_1 and β_2, if $\beta_2 \le \beta_1$, then every L_β with $\beta \in [\beta_2, \beta_1]$ will have error at most x for all f_1 values. Otherwise no such TPL scheme exists. Eventually, Algorithm 2 returns some $\beta \in [\beta_2, \beta_1]$, as well as the error achieved by L_β.

Correctness and complexity. We next prove the correctness of our approach for searching for the best β, and then comment on the computational/space complexity.

Theorem 4. **[Correctness of Algorithm 2]** *Algorithm 2 must return the optimal β value that minimizes the error of the resulting TPL scheme, under the given h, k, m, $f_{\max}$, t, and $\mathbb{S}$.*

Proof. For the corner case of $S_1 = 0$, Line 5 in the algorithm directly returns $\beta = 1$. This is optimal since when $S_1 = 0$, the only possible TPL scheme is L_1. We move on to prove the theorem for the more interesting case of $S_1 > 0$. First, if Algorithm 2 finds a TPL scheme L_β for some give error target x, then the error of L_β must be at most x, when $f_1 < tS_1$ (given how β_1 is found), and also when $f_1 \ge tS_1$ (given how β_2 is found). This in turns means that $error(\mathsf{L}_\beta) \le x$.

Next, for any given error target x, if there exists some TPL scheme L_β with $error(\mathsf{L}_\beta) \le x$, then we claim that Algorithm 2 must be able to find β' such that $error(\mathsf{L}_{\beta'}) \le x$. (Here β' may or may not equal β.) To prove, note that by Theorem 1, we must have $\beta_1 \ge \beta$. By Theorem 3, we must have $\beta_2 \le \beta$. This means that Algorithm 2 must return some $\beta' \in [\beta_2, \beta_1]$, where $error(\mathsf{L}_{\beta'}) \le x$. □

The above theorem has shown the correctness of our algorithm. We next comment on the computational complexity and space complexity. In actual implementation, there are some trivial performance optimizations that can be done:

- At Line 1, Algorithm 1 exhaustively tests all possible h values. Since $\mathbb{S}$ has total $|\mathbb{P}|$ nodes, there are $|\mathbb{P}|+1$ possible different values for h. But note that at Line 2, the value of $m-l$ cannot be negative. Hence at Line 1, we only need to try m (instead of $|\mathbb{P}|+1$) possible values for h.
- Algorithm 2 hence will be invoked only $O(m)$ times. The calculation of S_1 and S_2 at Line 5 can be done incrementally, from invocation to invocation. Namely, S_1 and S_2 in the current invocation can be computed with $O(1)$ computational complexity, based on the S_1 and S_2 in the previous invocation.
- Computing the error in various places of Algorithm 2 involves determining the tail probability of binomial distribution. As a trivial optimization, one can pre-compute and store the various terms of the binomial distribution, by incurring $O(m)$ computational complexity, together with $O(m)$ space complexity.

[9] If $f_{\max} < tS_1$, then it is impossible for $f_1 \ge tS_1$ to happen. In such a case, we directly set $\beta_2 = 0$ at Line 11.

With these trivial optimizations, the final computational complexity of Algorithm 1 (including its invocations of Algorithm 2) is $O(\text{poly}(m + \log \frac{1}{\eta}))$, where η is the precision for the real values in the binary search in Algorithm 2. More specifically, Algorithm 2 has a complexity of $O(m + \log^2(\frac{1}{\eta})) = O(\text{poly}(m + \log \frac{1}{\eta}))$, while Algorithm 1 simply invokes Algorithm 2 for $O(m)$ times. The space complexity of Algorithm 1 and Algorithm 2 are both just $O(m)$.

5.5 Best General Scheme Must Be TPL scheme

This section proves that the best general scheme must be a TPL scheme, or equivalently, that the best $\sigma_1()$ and $\sigma_2()$ must be of two-piece-linear-form.

Theorem 5. *Let Π be the best general scheme, namely the scheme with the smallest error under the worst-case adversary, among all possible general schemes. Then there must exist a TPL scheme L, such that $error(\mathsf{L}) = error(\Pi)$.*

The proof for this theorem, deferred to Appendix D, is rather involved. Here we provide some high-level intuitions. Roughly, we will first prove that for **any** given (f_1, f_2), there must exist TPL scheme L such that $error(\mathsf{L}, \mathbb{A}_{f_1,f_2}) \le error(\Pi, \mathbb{A}_{f_1,f_2})$. Here the construction of L is specific to the given f_1 and f_2.

Next, we invoke the above result with $f_1 = tS_1$ and $f_2 = f_{\max} - tS_1$. These f_1 and f_2 values correspond to the *pivot point* in Figure 3. Based on Theorem 2, under these f_1 and f_2 values, regardless of what L is, we must have $error(\mathsf{L}, \mathbb{A}_{f_1,f_2}) = err^*$ where err^* is defined in Equation 4. In turn, this means $err^* = error(\mathsf{L}, \mathbb{A}_{f_1,f_2}) \le error(\Pi, \mathbb{A}_{f_1,f_2}) \le error(\Pi)$. Essentially, this tells us that err^* is a lower bound on the error of Π.

In the final step, we show there exists some TPL scheme L whose maximum/peak error is achieved exactly at the pivot point. This then means that the error of L under the worst-case adversary is err^*. Then we have $error(\mathsf{L}) = err^* \le error(\Pi)$. By the definition of Π, we also trivially have $error(\mathsf{L}) \ge error(\Pi)$. This then leads to $error(\mathsf{L}) = error(\Pi)$.

In the proofs for the results in this section and the next section, sometimes to avoid a technicality caused by division-by-zero, our proofs will consider $f_{\max} \ne tS_1$. This treatment has no impact in practice: If $f_{\max}$ exactly equals tS_1, one can simply add a tiny amount (e.g., 10^{-10}) to $f_{\max}$, which gives the adversary some negligible extra power and has no practical impact. Also, for mathematical convenience, we sometimes use normal distribution to approximate the binomial distribution in the proofs.

6 Non-proportional Weight Almost Always Help

Section 4 showed that non-proportional schemes can outperform proportional schemes. But is the example in Section 4 just an exception, or does this phenomenon broadly exist?

Building upon the formal results in Section 5, we can now prove that this phenomenon broadly exists: Non-proportional weights will outperform proportional weights, *in all cases except for some theoretical corner cases*. This theoretical result is also consistent with our experiments later in Section 8.

Intuition. The proportional scheme (i.e., bipartition scheme) is simply the TPL scheme L_β with $\beta = 1$. For non-proportional weights to be beneficial, we just need to show that the optimal β (that minimizes $error(\mathsf{L}_\beta)$) is not 1.

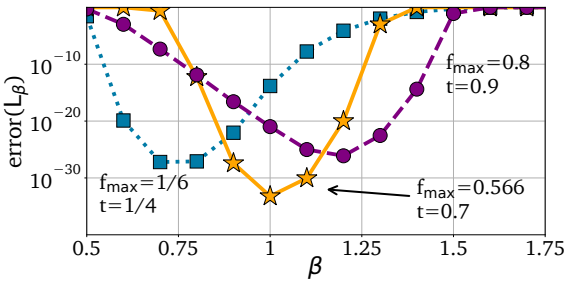


Figure 5: How the error of L_β under the worst-case adversary changes with β, for different $f_{\max}$ and t values. To provide a complete picture, these $f_{\max}$ and t values might not be practical values. These results are under $h = 0.0025$, $k = 500$, and Ethereum stake distribution [23].

Figure 5 illustrates how $error(\mathsf{L}_\beta)$ changes as a function of β. For example, for $f_{\max} = \frac{1}{6}$ and $t = \frac{1}{4}$, the optimal β is about 0.7. Depending on t and $f_{\max}$, the optimal β may either be below 1 or above 1. But in order for the optimal β to be exactly 1, we have to use a rather specific setting (i.e., $f_{\max} = 0.566$ and $t = 0.7$) in the figure. Hence it seems that the optimal β is 1, only in some corner cases.

Formal proofs. Appendix E formalizes the above intuition, by i) proving a necessary condition for the optimal β to be 1, and ii) showing that this necessary condition is rather hard to satisfy. We defer the details to Appendix E.

7 Extending to Local Randomness

Section 3 explained that committee selection may either use public or local randomness. So far this paper has been focusing on public randomness. We now naturally extend to local randomness.

Note on verifiability. For committee selection, it is crucial that the local randomness used by a node can be verified by others. Hence the local randomness is typically generated via a *verifiable random function* (*VRF*) [38], as in existing blockchains [15, 16, 26]. The specific design of VRF is entirely orthogonal to our work, which we do not separately discuss.

7.1 Existing Schemes with Local Randomness

We first describe the local randomness versions of the existing committee selection schemes.

Existing schemes. Under public randomness, Section 3 reviewed all the existing committee selection schemes: i) *basic scheme*, ii) *Fait Accompli (FA) scheme*, and iii) *bipartition scheme*.

Now with local randomness, in the *basic scheme* [15, 16, 26], each node p simulates $\frac{s_p}{u}$ *virtual nodes*. Here u is a sufficiently small constant, so that $\frac{s_p}{u}$ is an integer for all p. Conceptually, using the local randomness on p, each virtual node gets into the committee independently with the same probability. Node p is a committee member iff at least one of its virtual nodes gets in. The weight of p will be $\frac{y}{z}$, where y is the number of node p's virtual

nodes in the committee, and z is the total number of virtual nodes in the committee.

With local randomness, the *FA scheme* [25] chooses committee members from the *large* nodes in the same way as its public randomness version. But for choosing additional committee members from the *small* nodes, the FA scheme now uses the basic scheme (described above) with local randomness, except that the weight of small-node p is set to be $\frac{y}{z}S_2$ (instead of $\frac{y}{z}$). Recall that under public randomness, the FA scheme has a parameter $k \geq 1$, which is the number of committee members chosen from the small nodes. Now under local randomness, k is defined [25] to be the expectation of z. The probability x of each virtual node being elected into the committee can then be expressed as a function of k.[10]

Finally, the *bipartition scheme* is a trivial generalization of the FA scheme. Hence if local randomness is used in the FA scheme, then the corresponding bipartition scheme will also use local randomness.

Proportionality. One can verify that the weights used in all these schemes are always proportional, regardless of whether public or local randomness is used.

7.2 Our Schemes with Local Randomness

To explore non-proportional schemes, under public randomness, Section 5 defined Π_{σ_1,σ_2} and L_β. We now extend these concepts to the local randomness setting.

With local randomness, we define the *general scheme* Π_{σ_1,σ_2} to be the same as the bipartition scheme (with local randomness) except that:

- If node p is a large node, it has weight $\sigma_1(s_p)$.
- If node p is a small node, it has weight $\sigma_2(s_p) \cdot \frac{y}{z}$, where y and z are defined in the same way as in the FA scheme in the above.

Here $\sigma_1()$ and $\sigma_2()$ are arbitrary functions.

Next, with local randomness, the *TPL scheme* L_β is defined to be the general scheme Π_{σ_1,σ_2} with:

- $\sigma_1(s_p) = \alpha \cdot s_p$, where $\alpha = \frac{1-\beta \cdot S_2}{S_1}$
- $\sigma_2(s_p) = \beta \cdot S_2$

Again, we require a TPL scheme to always be normalized — hence α has to be a function of β.[11]

7.3 Extending Our Results to Local Randomness

Section 8 will show that, with local randomness, non-proportional weights continue to offer better security and smaller committee size than proportional weights. The amount of improvement is also quite similar to the public randomness case.

Under local randomness, our Algorithm 1 and 2 continue to work. The reason is that Theorem 1 through Theorem 4 continue to hold, under local randomness. (The proof of Theorem 4 continues to hold, without any change, under local randomness. The proofs of the other theorems only need to be slightly adapted – see Appendix F for the adapted version). Our final results in Section 8 under local randomness hence are still based on Algorithm 1 and 2.

8 Experimental Results

We have fully implemented our *non-proportional* committee selection scheme in Java.[12] This section uses our implementation to quantify the benefits of our non-proportional scheme.

8.1 Experimental Methodology

Using best parameter for existing schemes. Section 3 explained that *all* existing committee selection schemes [17, 25, 29, 30, 33, 42] can be viewed as special cases of the bipartition scheme with different h parameter values. In our experiments, we will exhaustively find the optimal h, and then use the bipartition scheme with that optimal h, as the state-of-the-art scheme. Doing so only makes the state-of-the-art results better. We will call this bipartition scheme, with the optimal h, as the *SOTA scheme*.

Our scheme. For our non-proportional committee selection scheme, we choose to use the TPL scheme found by Algorithm 1. In order to show that even limited "amount" of non-proportionality can help substantially, we constrain the β parameter in our scheme to be between 0.5 and 2. Such additional constraint (which is only applied to our scheme) can only make our results worse, and our conclusion stronger.

Lower/upper bound. To determine the error for each scheme, we need to use the worst-case adversary against that scheme. Recall that for both the bipartition scheme and our scheme, the worst-case adversary is captured by the worst-case f_1 and f_2 values. A minor difficulty here is that these values are continuous, which prevents exhaustive enumeration. We deal with this as follows.

For the bipartition scheme, we test $f_1 = 0, \delta, 2\delta, \ldots$ and $f_2 = f_{\max} - f_1$. Here δ is a constant, and our experiments use $\delta = 0.001$. We then find the worst-case f_1 and f_2 values among these. We do not test other f_1 and f_2 values. Doing so gives us a *lower bound* on the error of the existing schemes. By using such lower bound as their error, we can only make their results better.

For our scheme, we test $f_1 = 0, \delta, 2\delta, \ldots$ and $f_2 = f_{\max} - f_1 + \delta$. Note that we intentionally use $f_1 + f_2 = f_{\max} + \delta > f_{\max}$. We then view the worst-case error, under these f_1 and f_2 values, as the error of our scheme. One can trivially verify that if the error of our scheme is x under $f_1 = c\delta$ and $f_2 = f_{\max} - f_1 + \delta = f_{\max} - (c-1)\delta$, then its error must be[13] at most x under all $f_1 \in [(c-1)\delta, c\delta]$ and $f_2 = f_{\max} - f_1 \in [f_{\max} - c\delta, f_{\max} - (c-1)\delta]$. Putting it another way, our approach here gives an *upper bound* on the error. By using this upper bound as our error, we can only make our results worse.

Values of t and $f_{\max}$ in our experiments. Recall that the value of t comes from the requirement of the application (i.e., security protocol) that uses the committee, and is usually an inherent property of the application. The value of $f_{\max}$ is the maximum corruption level that the application aims/chooses to tolerate. These values are not controlled by the adversary. For a committee selection scheme, these two values can be viewed as part of the "environment". Our

[10] Specifically, since $k = \mathrm{E}[z] = \frac{S_2}{u}x$, we have $x = \frac{ku}{S_2}$.

[11] Same as before, as a technicality, when $S_1 = 0$, the value of α is not relevant. Also, when $S_1 = 0$, we require $\beta = 1$, since this is the only β value that can make the scheme normalized.

[12] The source code for our implementation is available at [43].

[13] This is simply because for all f_1 and f_2 values in those two ranges, we always have $f_1 \leq c\delta$ and $f_2 \leq f_{\max} - (c-1)\delta$.

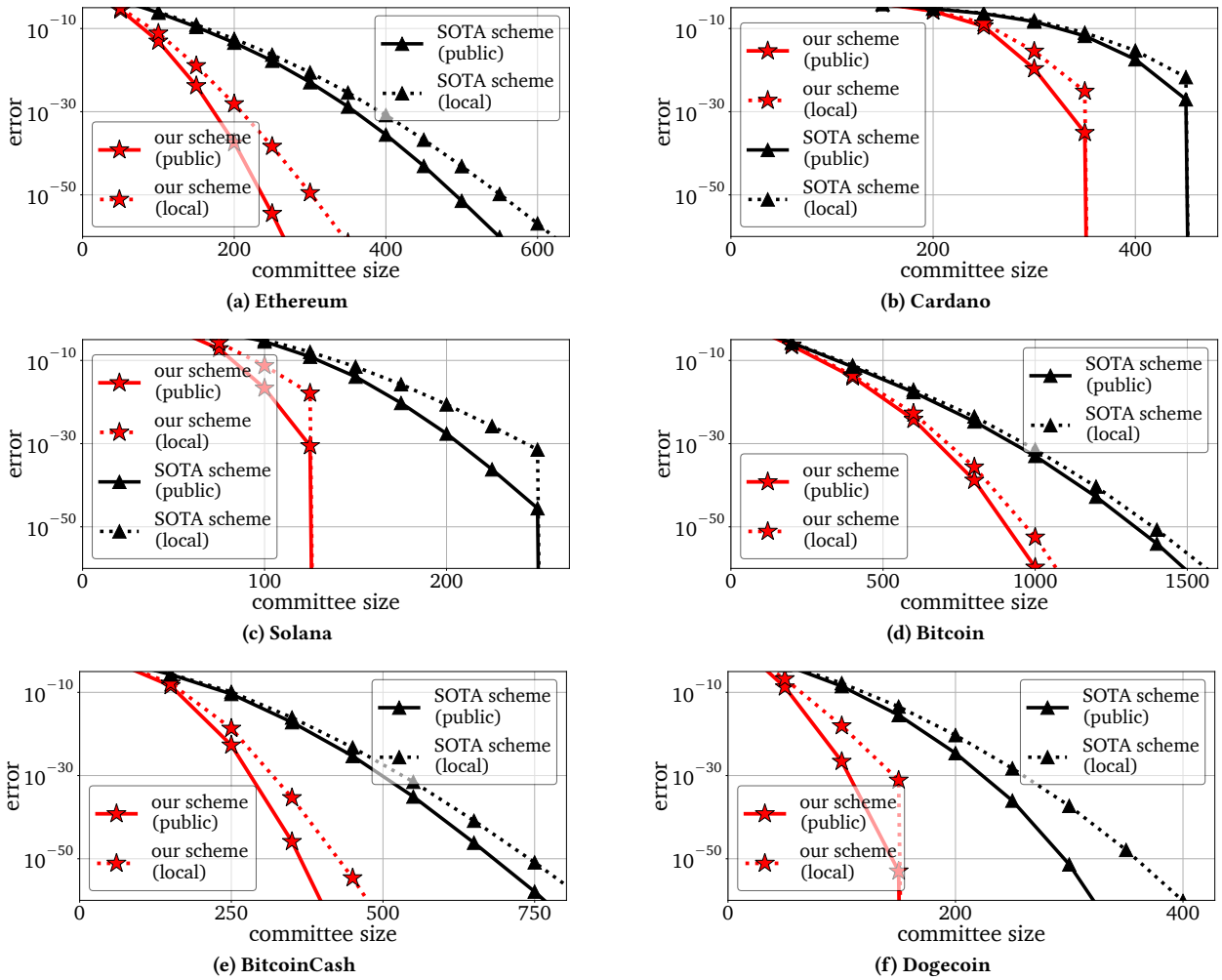


Figure 6: Benefits of non-proportional weights (i.e., our scheme) under various stake distributions (with $f_{\max} = \frac{1}{5}$ and $t = \frac{1}{3}$).

experiments will extensively consider various practical t and $f_{\max}$ values appeared in prior works [3, 15–17, 26, 28, 32, 33, 42].

8.2 Results under Various Stake Distributions

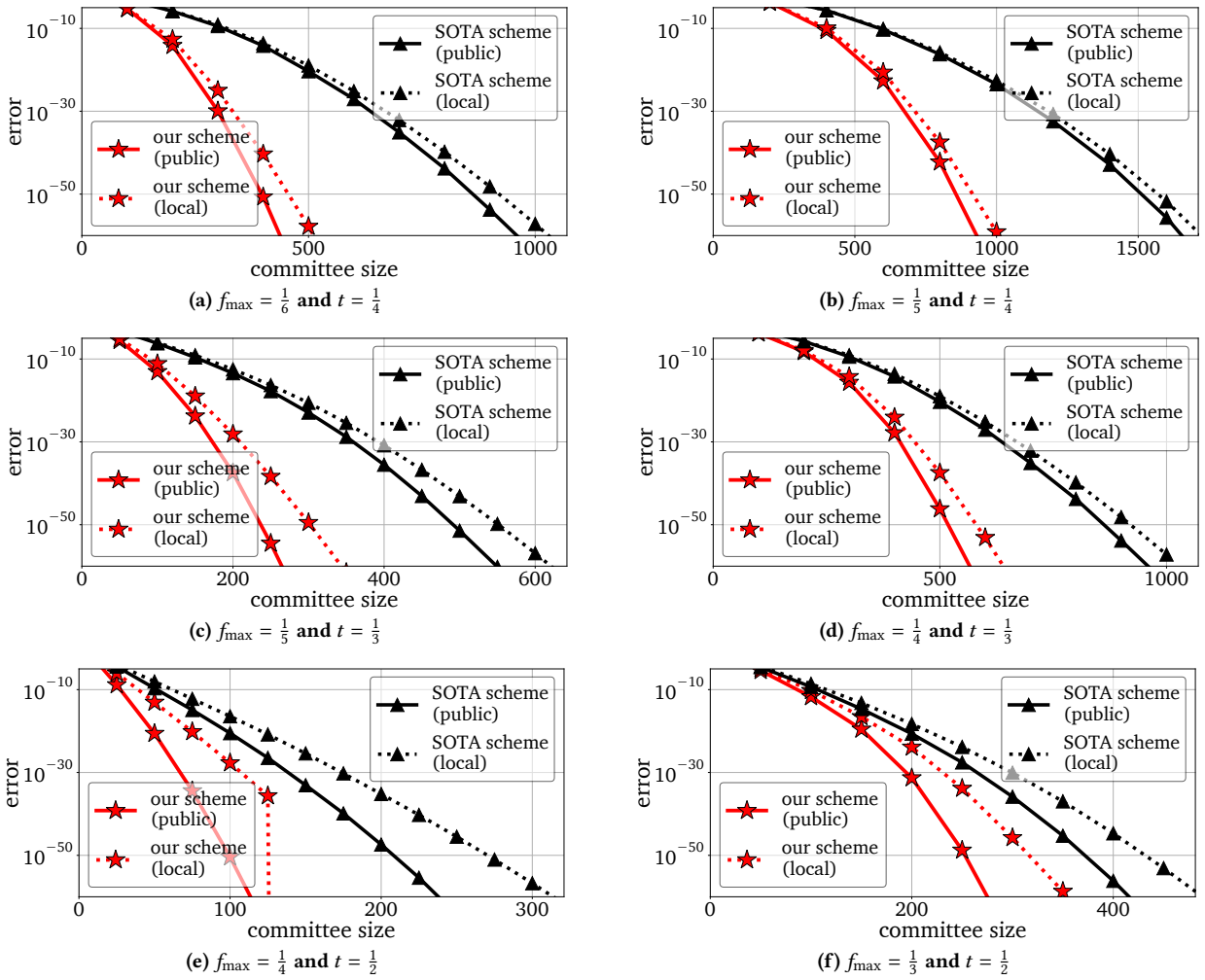
To use real-world stake distributions, we consider the top 15 cryptocurrencies in terms of market value [14]. The stake distributions for 6 of them are publicly available, at [23] for Ethereum, [6] for Solana, [24] for Cardano, [8] for Bitcoin, [9] for BitcoinCash, and [10] for Dogecoin. These will be the distributions that we use, where each distribution usually contains thousands of nodes.[14] We use $f_{\max} = \frac{1}{5}$ and $t = \frac{1}{3}$, as a practical setting, for results in this section. These $f_{\max}$ and t values are used in Algorand [26] and GearBox [17] for their committees. The next section explores other $f_{\max}$ and t values.

Using the Ethereum distribution, Figure 6(a) compares the error achieved by the SOTA scheme and by our scheme. Recall that the SOTA scheme and our scheme select committee members in exactly the same way. The only difference is that our scheme uses non-proportional weights, for the chosen committee members. For each scheme, the figure presents results for both the public and the local randomness version. Under public randomness, the committee size is simply the total number of large nodes (who are always in the committee) plus the k additional committee members selected from the small nodes. Committee size under local randomness is the same, except that k is the expected number of committee members selected from the small nodes [25].

Smaller error. Figure 6(a) shows that with the same committee size, our scheme reduces error by many orders of magnitude, compared to the SOTA scheme. For example under public randomness and with a committee size of 200, our scheme has about 10^{-38} error, while the SOTA scheme has about 10^{-14} error. Under local randomness and with a committee size of about 250, our scheme achieves a similar reduction of error from 10^{-16} to 10^{-39}.

Smaller committee size. Figure 6(a) also shows that to achieve the same target error, our scheme reduces the committee size by about 50%, as compared to the SOTA scheme. (Section 1 explained that even a 30% reduction of committee size is significant.)

[14] Some of these (e.g., Bitcoin) are not based on Proof-of-Stake, but the notion of stake distribution (i.e., how much cryptocurrency each party holds) is still well-defined.



(a) $f_{\max} = \frac{1}{6}$ and $t = \frac{1}{4}$ (b) $f_{\max} = \frac{1}{5}$ and $t = \frac{1}{4}$

(c) $f_{\max} = \frac{1}{5}$ and $t = \frac{1}{3}$ (d) $f_{\max} = \frac{1}{4}$ and $t = \frac{1}{3}$

(e) $f_{\max} = \frac{1}{4}$ and $t = \frac{1}{2}$ (f) $f_{\max} = \frac{1}{3}$ and $t = \frac{1}{2}$

Figure 7: Benefits of non-proportional weights (i.e., our scheme) under various $(f_{\max}, t)$ values and the Ethereum distribution.

Specifically, let us consider 10^{-40} as an example target error, which is the error of the committee selection scheme in Algorand's deployment [11]. As shown in [11], because committee selection is done over and over again in a blockchain, its error needs to be rather small. If committee selection were viewed as a crypto primitive, then 10^{-40} would roughly correspond to 128-bit security. Now to achieve 10^{-40} error, Figure 6(a) shows that our scheme only needs a committee size of about 210 (under public randomness) or 260 (under local randomness). In comparison, the SOTA scheme needs about 420 (under public randomness) or 470 (under local randomness).

Improvements under other stake distributions. Figure 6(b)-(f) presents further results under the stake distributions of the 5 other cryptocurrencies. Consistent with Figure 6(a), these results also show that non-proportional weights (i.e., our scheme) help to significantly decrease error. For achieving the same target error, for 4 out of the 6 distributions, non-proportional weights help to reduce committee size by about 50%. The reduction for the remaining 2 distributions is smaller. But even for the worst one (in Figure 6(b)), the reduction is still non-trivial, around 20% to 25%.

8.3 Results under Various $f_{\max}$ and t Values

Our results so far are for $f_{\max} = \frac{1}{5}$ and $t = \frac{1}{3}$. We now experiment with other $f_{\max}$ and t values, using the Ethereum stake distribution. For the value of t, we consider:

- $t = \frac{1}{2}$ as in [15, 16, 28, 33]
- $t = \frac{1}{3}$ as in [17, 26, 42]
- $t = \frac{1}{4}$ as in [3, 32]

For each t value, we experiment with two different $f_{\max}$ values. Figure 7 presents all these results. Figure 7 shows that under all these $f_{\max}$ and t values, non-proportional weights (i.e., our scheme) consistently help to decrease error and reduce committee size. While the exact amount of improvement varies, the improvement is significant in most, if not all, cases. These results are in line with our theoretical results in Section 6, which show that non-proportional weights can always help, except in some corner cases.

Besides the Ethereum distribution, using the stake distributions of the 5 other cryptocurrencies, we have also similarly experimented with various $f_{\max}$ and t values. The findings there are similar, and due to space constraints, we do not separately plot those.

9 Additional Related Works

Section 3 and 7.1 already reviewed existing committee selection schemes [15–17, 25, 26, 29, 30, 33, 42] in PoS blockchains. As explained there, we explore *non-proportional* weights, while they all use *proportional* weights. This section discusses additional related works from a broader context.

Committee selection in PoW blockchains. Proof-of-Work (PoW) blockchains sometimes also use committees. For example, in [34, 35, 37, 41, 46], a node gets a weight of w in the committee, if it finds w solutions for the computational puzzle. Fundamentally, this is equivalent to the basic committee selection scheme (using local randomness) in the PoS setting. Namely, we can view s_p as the total computational power of a party p, and u as the amount of computation needed for trying a *single* attempted solution to the computational puzzle. Then party p conceptually simulates $\frac{s_p}{u}$ *virtual nodes*, where each virtual node has the same probability x of getting into the committee. Here x simply corresponds to the probability that the attempted solution happens to be a correct solution. We explained in Section 7.1 that such a scheme always uses proportional weights. In contrast, our work focuses on non-proportional weights.

Selfish behavior of users. Benhaim et al. [7] study equilibrium behavior of voting strategies of selfish committee members in blockchains, in the context of blockchain governance. In the context of Algorand, Dimitri [18] studies the trade-off between a user participating in committee selection (which would cause the corresponding stake to be "locked" for some time) and not participating, from the perspective of that user's utility. The focuses of these works are orthogonal to ours.

10 Conclusions

This work investigates committee selection schemes in PoS blockchains. Surprisingly, we discover that the principle of *proportionality*, which all existing committee selection schemes stick to, leads to sub-optimal designs. We then explore the design space of schemes that deviate from proportionality, to find the best design. Our final committee selection scheme, which is non-proportional, significantly outperforms existing schemes under realistic settings and real-world stake distributions.

References

[1] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. 2017. Solida: A Blockchain Protocol Based on Reconfigurable Byzantine Consensus. In *International Conference on Principles of Distributed Systems*.

[2] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Maofan Yin. 2020. Sync HotStuff: Simple and Practical Synchronous State Machine Replication. In *IEEE S&P*.

[3] Ittai Abraham, Ling Ren, and Zhuolun Xiang. 2022. Good-Case and Bad-Case Latency of Unauthenticated Byzantine Broadcast: A Complete Categorization. In *OPODIS*.

[4] Yackolley Amoussou-Guenou, Antonella Del Pozzo, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. 2020. On Fairness in Committee-Based Blockchains. In *2th International Conference on Blockchain Economics, Security and Protocols, Tokenomics*.

[5] Pierre-Louis Aublin, Sonia Ben Mokhtar, and Vivien Quéma. 2013. RBFT: Redundant Byzantine Fault Tolerance. In *ICDCS*.

[6] Solana Beach. 2024. *Solana stake distribution.* https://solanabeach.io/validators.

[7] Alon Benhaim, Brett Hemenway Falk, and Gerry Tsoukalas. 2023. Scaling Blockchains: Can Committee-Based Consensus Help? *Management Science* (2023).

[8] BitInfoCharts. 2024. *Bitcoin stake distributions.* Data are spread across multiple webpages: https://bitinfocharts.com/top-100-richest-bitcoin-addresses-1.html, https://bitinfocharts.com/top-100-richest-bitcoin-addresses-2.html, . . ., https://bitinfocharts.com/top-100-richest-bitcoin-addresses-100.html.

[9] BitInfoCharts. 2024. *BitcoinCash stake distributions.* Data are spread across multiple webpages: https://bitinfocharts.com/top-100-richest-bitcoin%20cash-addresses-1.html, https://bitinfocharts.com/top-100-richest-bitcoin%20cash-addresses-2.html, . . ., https://bitinfocharts.com/top-100-richest-bitcoin%20cash-addresses-100.html.

[10] BitInfoCharts. 2024. *Dogecoin stake distributions.* Data are spread across multiple webpages: https://bitinfocharts.com/top-100-richest-dogecoin-addresses-1.html, https://bitinfocharts.com/top-100-richest-dogecoin-addresses-2.html, . . ., https://bitinfocharts.com/top-100-richest-dogecoin-addresses-100.html.

[11] Erica Blum, Derek Leung, Julian Loss, Jonathan Katz, and Tal Rabin. 2023. Analyzing the Real-World Security of the Algorand Blockchain. In *CCS*.

[12] Miguel Castro and Barbara Liskov. 1999. Practical Byzantine Fault Tolerance. In *Symposium on Operating Systems Design and Implementation*.

[13] Coinbase. 2024. *Ethereum USD conversion.* https://www.coinbase.com/en-sg/converter/eth/usd.

[14] CoinMarketCap. 2024. *Today's Cryptocurrency Prices by Market Cap.* https://coinmarketcap.com.

[15] Phil Daian, Rafael Pass, and Elaine Shi. 2019. Snow white: Robustly reconfigurable consensus and applications to provably secure proof of stake. In *FC*.

[16] Bernardo David, Peter Gaži, Aggelos Kiayias, and Alexander Russell. 2018. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In *EUROCRYPT*.

[17] Bernardo David, Bernardo Magri, Christian Matt, Jesper Buus Nielsen, and Daniel Tschudi. 2022. GearBox: Optimal-size Shard Committees by Leveraging the Safety-Liveness Dichotomy. In *CCS*.

[18] Nicola Dimitri. 2022. The Economics of Consensus in Algorand. *FinTech* 1, 2 (2022), 164–179.

[19] Ethereum. 2024. *Ethereum Gas and Fees.* https://ethereum.org/en/developers/docs/gas.

[20] Ethereum. 2024. Ethereum Proof-of-Stake (PoS). https://ethereum.org/en/developers/docs/consensus-mechanisms/pos.

[21] Etherscan. 2024. *Ethereum Average Gas Price Chart.* https://etherscan.io/chart/gasprice.

[22] Etherscan. 2024. *Ethereum Daily Gas Used Chart.* https://etherscan.io/chart/gasused.

[23] Etherscan. 2024. *Ethereum stake distribution.* https://etherscan.io/accounts.

[24] Cardano Explorer. 2024. *Cardano stake distribution.* https://cexplorer.io.

[25] Peter Gazi, Aggelos Kiayias, and Alexander Russell. 2023. Fait Accompli Committee Selection: Improving the Size-Security Tradeoff of Stake-Based Committees. In *CCS*.

[26] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling byzantine agreements for cryptocurrencies. In *SOSP*.

[27] Guy Golan Gueta, Ittai Abraham, Shelly Grossman, Dahlia Malkhi, Benny Pinkas, Michael Reiter, Dragos-Adrian Seredinschi, Orr Tamir, and Alin Tomescu. 2019. SBFT: A Scalable and Decentralized Trust Infrastructure. In *DSN*.

[28] Vipul Goyal, Hanjun Li, and Justin Raizes. 2021. Instant Block Confirmation in the Sleepy Model. In *FC*.

[29] Ruomu Hou and Haifeng Yu. 2023. Optimistic Fast Confirmation While Tolerating Malicious Majority in Blockchains. In *IEEE S&P*.

[30] Ruomu Hou, Haifeng Yu, and Prateek Saxena. 2022. Using throughput-centric byzantine broadcast to tolerate malicious majority in blockchains. In *IEEE S&P*.

[31] Idit Keidar, Eleftherios Kokoris-Kogias, Oded Naor, and Alexander Spiegelman. 2021. All You Need is DAG. In *PODC*.

[32] Mahimna Kelkar, Soubhik Deb, Sishan Long, Ari Juels, and Sreeram Kannan. 2023. Themis: Fast, Strong Order-Fairness in Byzantine Consensus. In *CCS*.

[33] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *CRYPTO*.

[34] Eleftherios Kokoris Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. 2016. Enhancing bitcoin security and performance with strong consistency via collective signing. In *USENIX Security*.

[35] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ewa Syta, and Bryan Ford. 2018. Omniledger: A secure, scale-out, decentralized ledger via sharding. In *IEEE S&P*.

[36] Ramakrishna Kotla, Lorenzo Alvisi, Mike Dahlin, Allen Clement, and Edmund Wong. 2010. Zyzzyva: Speculative Byzantine fault tolerance. *ACM Transactions on Computer Systems (TOCS)* 27, 4 (2010), 1–39.

[37] Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. 2016. A secure sharding protocol for open blockchains. In *CCS*.

[38] Silvio Micali, Salil Vadhan, and Michael Rabin. 1999. Verifiable Random Functions. In *FOCS*.

[39] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. 2016. The Honey Badger of BFT Protocols. In *CCS*.

[40] Michael Mitzenmacher and Eli Upfal. 2005. *Probability and computing: Randomized algorithms and probabilistic analysis.* Cambridge university press.

[41] Rafael Pass and Elaine Shi. 2017. Hybrid consensus: Efficient consensus in the permissionless model. In *DISC*.

[42] Daniel Reijsbergen, Pawel Szalachowski, Junming Ke, Zengpeng Li, and Jianying Zhou. 2021. LaKSA: A Probabilistic Proof-of-Stake Protocol. In *NDSS*.
[43] Yucheng Sun. 2025. Source code for experiments in this paper. https://www.comp.nus.edu.sg/~sunyuch/projects/ccs25/ccs25.html.
[44] Andrei Tonkikh and Luciano Freitas. 2024. Swiper: A new paradigm for efficient weighted distributed protocols. In *PODC*.
[45] Yibin Xu, Jingyi Zheng, Boris Düdder, Tijs Slaats, and Yongluan Zhou. 2024. A Two-Layer Blockchain Sharding Protocol Leveraging Safety and Liveness for Enhanced Performance. In *NDSS*.
[46] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. 2018. Rapidchain: Scaling blockchain via full sharding. In *CCS*.

A Developing the Intuitions in Section 4.2

Recall the two committee selection schemes NProp and Prop in Section 4.2. There NProp is obtained from Prop, by doing some *weight adjustments*, which slightly decrease (increase) the weights of the small (large) nodes. This appendix aims to derive intuitions on how such weight adjustments affect $\mathrm{E}[W_{\mathsf{corrupt}}]$, $\mathrm{Var}[W_{\mathsf{corrupt}}]$, and $\Pr[W_{\mathsf{corrupt}} \geq t]$. Such intuitions are summarized in Table 3.

Let $\mathbf{Binom}()$ denote the binomial distribution. It is easy to verify that for the bipartition scheme Prop, under given f_1 and f_2, we have:

$$W_{\mathsf{corrupt}} \quad \sim \quad f_1 + \frac{S_2}{k}\mathbf{Binom}(k, \frac{f_2}{S_2}) \tag{8}$$

Namely, we select small-node committee members total k times. For each selection, there is $\frac{f_2}{S_2}$ probability of selecting a corrupted node. The weight of each selection is $\frac{S_2}{k}$. This gives us the term $\frac{S_2}{k}\mathbf{Binom}(k, \frac{f_2}{S_2})$. Finally, all the large-nodes are always included in the committee, and the total stake of corrupted large-node committee members is always f_1. This gives us the term f_1 in the above equation.

Effect of NProp's weight adjustments on $\mathrm{Var}[W_{\mathsf{corrupt}}]$. The weight adjustments done by scheme NProp reduces the weight of the $\frac{S_2}{k}\mathbf{Binom}(k, \frac{f_2}{S_2})$ term in Equation 8. Since only this term contributes to $\mathrm{Var}[W_{\mathsf{corrupt}}]$, these adjustments always reduce $\mathrm{Var}[W_{\mathsf{corrupt}}]$.

Effect of NProp's weight adjustments on $\mathrm{E}[W_{\mathsf{corrupt}}]$. Next, how these weight adjustments affect $\mathrm{E}[W_{\mathsf{corrupt}}]$ depends on the value of f_1:

- When $f_1 < f_{\max}S_1$, these adjustments always decrease $\mathrm{E}[W_{\mathsf{corrupt}}]$. This is because $\frac{f_1}{S_1} < f_{\max} < \frac{f_2}{S_2}$ when $f_1 < f_{\max}S_1$. Hence the large nodes are "less corrupted". Reducing the weight given to the small nodes, who are "more corrupted", hence always decreases $\mathrm{E}[W_{\mathsf{corrupt}}]$.
- When $f_1 = f_{\max}S_1$, these adjustments do not affect $\mathrm{E}[W_{\mathsf{corrupt}}]$, since the large node and the small nodes have the same corruption ratio. Namely, $f_{\max}$ fraction (in terms of stake) of the small nodes are corrupted, and the same applies to the large nodes.
- When $f_1 > f_{\max}S_1$, these adjustments always increase $\mathrm{E}[W_{\mathsf{corrupt}}]$, by a similar argument as earlier.

Effect of NProp's weight adjustments on $\Pr[W_{\mathsf{corrupt}} \geq t]$. Usually smaller $\mathrm{E}[W_{\mathsf{corrupt}}]$ and smaller $\mathrm{Var}[W_{\mathsf{corrupt}}]$ will lead to smaller error (i.e., $\Pr[W_{\mathsf{corrupt}} \geq t]$). The final effects on error, intuitively, can be explained by the changes in $\mathrm{Var}[W_{\mathsf{corrupt}}]$ and $\mathrm{E}[W_{\mathsf{corrupt}}]$:

- When $f_1 < f_{\max}S_1$, the adjustments done by NProp will likely reduce error, since they will decrease both $\mathrm{Var}[W_{\mathsf{corrupt}}]$ and $\mathrm{E}[W_{\mathsf{corrupt}}]$.
- When $f_1 = f_{\max}S_1$, these adjustments will likely reduce error, since they will decrease $\mathrm{Var}[W_{\mathsf{corrupt}}]$ and will not affect $\mathrm{E}[W_{\mathsf{corrupt}}]$.
- When $f_1 > f_{\max}S_1$, these adjustments will decrease $\mathrm{Var}[W_{\mathsf{corrupt}}]$ but increase $\mathrm{E}[W_{\mathsf{corrupt}}]$:
  (1) If f_1 is slightly larger than $f_{\max}S_1$, then they will likely reduce error. This is because the increase of $\mathrm{E}[W_{\mathsf{corrupt}}]$ will be insignificant, and the reduction of $\mathrm{Var}[W_{\mathsf{corrupt}}]$ will be the dominating factor.
  (2) Intuitively, there will be some f_1 value where the two effects cancel out. At that equilibrium point, these adjustments have no impact on error.
  (3) If f_1 is much larger than $f_{\max}S_1$, then these adjustments will likely increase error. This is because the increase of $\mathrm{E}[W_{\mathsf{corrupt}}]$ will be substantial and be the dominating factor.

B Proof for Theorem 2

Proof. If $S_1 > 0$, let us consider any TPL scheme L_β where $\beta > 0$. By Equation 7:

$$\begin{aligned} & \mathit{error}(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2}) \\ = \quad & \Pr[\mathbf{Binom}(k, \frac{f_2}{S_2}) \geq k(\frac{tS_1 - f_1}{\beta S_2 S_1} + \frac{f_1}{S_1})] \\ = \quad & \Pr[\mathbf{Binom}(k, \frac{f_2}{S_2}) \geq kt] \qquad (\text{when } f_1 = tS_1) \end{aligned}$$

Note that the last term does not depend on β. Hence:

$$\mathit{error}(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) = \mathit{error}(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2}) = \Pr[\mathbf{Binom}(k, \frac{f_2}{S_2}) \geq kt] \tag{9}$$

Finally, if $S_1 = 0$, then we must have $\beta_1 = \beta_2 = 1$, and $\mathit{error}(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) = \mathit{error}(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$ trivially holds. □

C Proof for Theorem 3

Proof. If $S_1 > 0$, let us consider any TPL scheme L_β where $\beta > 0$. By Equation 7, we have $\mathit{error}(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2}) = \Pr[\mathbf{Binom}(k, \frac{f_2}{S_2}) \geq k(\frac{tS_1 - f_1}{\beta S_2 S_1} + \frac{f_1}{S_1})]$. Since $f_1 \geq tS_1$, when β decreases, the term $\frac{tS_1 - f_1}{\beta S_2 S_1}$ decreases and thus $\mathit{error}(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2})$ never decreases. This implies $\mathit{error}(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) \geq \mathit{error}(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$, if $0 < \beta_1 \leq \beta_2$.

Finally, if $\beta_1 = 0$ and $S_1 > 0$, then we have $\mathit{error}(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) = \Pr[f_1 \cdot \alpha \geq t] = \Pr[f_1 \cdot \frac{1}{S_1} \geq t] = 1$, which is $\geq \mathit{error}(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$. If $S_1 = 0$, then we must have $\beta_1 = \beta_2 = 1$, and $\mathit{error}(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) \geq \mathit{error}(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$ trivially holds. □

D Proof for Theorem 5

Proof. By definition of Π, we must have $\mathit{error}(\mathsf{L}) \geq \mathit{error}(\Pi)$. Hence it suffices to prove $\mathit{error}(\mathsf{L}) \leq \mathit{error}(\Pi)$.

Consider the threshold h in the general scheme Π. We first prove the theorem, when h is such that $S_1 = 0$. In such a case, by Lemma 6 later, there must exist a TPL scheme L, such that $\mathit{error}(\mathsf{L}, \mathbb{A}_{0,f_{\max}}) \leq \mathit{error}(\Pi, \mathbb{A}_{0,f_{\max}})$. Since $S_1 = 0$, the only f_1 and f_2 values are $f_1 =$

value of f_1	$\text{Var}[W_{\text{corrupt}}]$	$\text{E}[W_{\text{corrupt}}]$	error (i.e., $\Pr[W_{\text{corrupt}} \geq t]$)
if $f_1 < f_{\max} \cdot S_1$	↘	↘	↘
if $f_1 = f_{\max} \cdot S_1$	↘	no change	↘
if f_1 is slightly larger than $f_{\max} \cdot S_1$	↘	↗ (insignificant)	↘
if $f_1 > f_{\max} \cdot S_1$ and is at some equilibrium value (e.g., when $f_1 \approx 0.18$ in Figure 2)	↘	↗	**no change**
if f_1 is much larger than $f_{\max} \cdot S_1$	↘	↗ (significant)	↗

Table 3: Intuitions on the effects of the weight adjustments done by scheme NProp.

0 and $f_2 = f_{\max}$. Hence we have $error(\mathsf{L}) = error(\mathsf{L}, \mathbb{A}_{0,f_{\max}}) \leq error(\Pi, \mathbb{A}_{0,f_{\max}}) \leq error(\Pi)$.

We next prove the theorem, when h is such that $S_1 > 0$. Our proof depends on whether $f_{\max} > tS_1$. We first prove the harder case of $f_{\max} > tS_1$.

Let $f_1 = tS_1$ and $f_2 = f_{\max} - tS_1$. By Lemma 6 later, there must exist a TPL scheme B, such that $error(\Pi, \mathbb{A}_{f_1,f_2}) \geq error(\mathsf{B}, \mathbb{A}_{f_1,f_2})$. If the β parameter in B is 0, then we must have $error(\mathsf{B}, \mathbb{A}_{f_1,f_2}) = 1$. Hence $error(\Pi) \geq error(\Pi, \mathbb{A}_{f_1,f_2}) \geq error(\mathsf{B}, \mathbb{A}_{f_1,f_2}) = 1 = error(\mathsf{B})$. Then we are done since we can directly use B as L, so that $error(\mathsf{L}) \leq error(\Pi)$.

Now if the β parameter in B is not 0, then by Theorem 2, we have $error(\mathsf{B}, \mathbb{A}_{f_1,f_2}) = err^*$, where err^* is defined in Equation 4. Hence $error(\Pi) \geq error(\Pi, \mathbb{A}_{f_1,f_2}) \geq error(\mathsf{B}, \mathbb{A}_{f_1,f_2}) = err^*$. To prove the theorem, we just need to find a TPL scheme L such that $error(\mathsf{L}) \leq err^*$.

By Equation 7, for all $\beta > 0$, we have $error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_{\max}-f_1}) = \Pr[\textbf{Binom}(k, \frac{f_{\max}-f_1}{S_2}) \geq k(\frac{tS_1-f_1}{\beta S_2 S_1} + \frac{f_1}{S_1})]$. Using a normal approximation of the binomial distribution, the above probability becomes $\Pr[X \geq k(\frac{tS_1-f_1}{\beta S_2 S_1} + \frac{f_1}{S_1})]$, where $X \sim \mathbf{N}(\frac{k\cdot(f_{\max}-f_1)}{S_2}, \frac{k(f_{\max}-f_1)}{S_2}(1 - \frac{f_{\max}-f_1}{S_2}))$. Here $\mathbf{N}()$ denotes the normal distribution. Define Y to be the standard normal random variable, namely, $Y \sim \mathbf{N}(0,1)$. Then for all $f_1 \notin \{f_{\max} - S_2, f_{\max}\}$, we have:

$$\begin{aligned} & error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_{\max}-f_1}) \\ = \;& \Pr[X \geq k(\frac{tS_1 - f_1}{\beta S_2 S_1} + \frac{f_1}{S_1})] \\ = \;& \Pr\left[\frac{X - \frac{k\cdot(f_{\max}-f_1)}{S_2}}{\sqrt{\frac{k\cdot(f_{\max}-f_1)}{S_2} \cdot (1 - \frac{f_{\max}-f_1}{S_2})}} \geq \lambda(f_1)\right] \\ = \;& \Pr[Y \geq \lambda(f_1)], \end{aligned} \tag{10}$$

where:

$$\lambda(f_1) \;=\; \frac{k(\frac{tS_1-f_1}{\beta S_2 S_1} + \frac{f_1}{S_1}) - \frac{k\cdot(f_{\max}-f_1)}{S_2}}{\sqrt{\frac{k\cdot(f_{\max}-f_1)}{S_2} \cdot (1 - \frac{f_{\max}-f_1}{S_2})}} \tag{11}$$

Obviously, $\Pr[Y \geq \lambda(f_1)]$ increases as $\lambda(f_1)$ decreases. Since $error(\mathsf{L}_\beta)$ is defined over the worst-case f_1, we need to reason about the minimum value of $\lambda(f_1)$, across all f_1 values. We take the derivative:

$$\frac{\partial\lambda(f_1)}{\partial f_1} = \frac{a \cdot f_1 + b}{2\beta S_1 \sqrt{\frac{S_2}{k}} \left((f_{\max} - f_1)(1 - \frac{f_{\max}-f_1}{S_2})\right)^{3/2}} \tag{12}$$

where:

$$a = \beta(2f_{\max} - 1) + 1 + \frac{2tS_1}{S_2} - \frac{2f_{\max}}{S_2} \tag{13}$$

$$b = \beta(f_{\max} - 2f_{\max}^2 + f_{\max}S_2) - 2f_{\max} - \frac{2tf_{\max}S_1}{S_2} + \frac{2f_{\max}^2}{S_2} + tS_1 \tag{14}$$

Note that on the right-hand side of Equation 12, the denominator is always positive, while the numerator $a \cdot f_1 + b$ is linear with respect to f_1.

Define β^* to be the β value such that $a \cdot f_1 + b = 0$ when $f_1 = tS_1$. Lemma 7 later will prove that such β^* must exist, and that:

$$0 < \frac{f_{\max} - tS_1}{f_{\max}S_2} < \beta^* < \frac{tS_1 - f_{\max} + S_2}{(1 - f_{\max})S_2} < \frac{1}{S_2} \tag{15}$$

We conjecture that the TPL scheme L_{β^*} offers the desirable property of $error(\mathsf{L}_{\beta^*}) \leq err^*$. To prove this key conjecture, we need to show that the error of L_{β^*} is at most err^*, under all possible f_1 values. The value of f_1 must satisfy:

$$\max(0, f_{\max} - S_2) \leq f_1 \leq \min(f_{\max}, S_1) \tag{16}$$

We next prove this key conjecture, by considering all possible f_1 values in Equation 16:

- $error(\mathsf{L}_{\beta^*}, \mathbb{A}_{f_1,f_{\max}-f_1}) = 0$ if $f_{\max} - S_2 \geq 0$ and $f_1 = f_{\max} - S_2$.
- $error(\mathsf{L}_{\beta^*}, \mathbb{A}_{f_1,f_{\max}-f_1}) = 0$ if $f_{\max} \leq S_1$ and $f_1 = f_{\max}$.
- $error(\mathsf{L}_{\beta^*}, \mathbb{A}_{f_1,f_{\max}-f_1}) \leq err^*$ when $f_1 \notin \{f_{\max} - S_2, f_{\max}\}$.

Here the two cases of $f_1 = f_{\max} - S_2$ and $f_1 = f_{\max}$ are the boundary values in Equation 16 for f_1, and need to be separately treated, because Equation 10 does not hold under those two cases.

If $f_{\max} - S_2 \geq 0$ and $f_1 = f_{\max} - S_2$, by Equation 7 and Equation 15, we have $error(\mathsf{L}_{\beta^*}, \mathbb{A}_{f_1,f_{\max}-f_1}) = \Pr[\textbf{Binom}(k, \frac{f_{\max}-f_1}{S_2}) \geq k(\frac{tS_1-f_1}{\beta^* S_1 S_2} + \frac{f_1}{S_1})] = \Pr[\textbf{Binom}(k,1) \geq k(\frac{tS_1-f_{\max}+S_2}{\beta^* S_1 S_2} + \frac{f_{\max}-S_2}{S_1})] = \Pr[k \geq k(\frac{tS_1-f_{\max}+S_2}{\beta^* S_1 S_2} + \frac{f_{\max}-S_2}{S_1})] = \Pr[\beta^* \geq \frac{tS_1-f_{\max}+S_2}{(1-f_{\max})S_2}] = 0$.

If $f_{\max} \leq S_1$ and $f_1 = f_{\max}$, by Equation 7 and Equation 15, we have $error(\mathsf{L}_{\beta^*}, \mathbb{A}_{f_1,f_{\max}-f_1}) = \Pr[\textbf{Binom}(k, \frac{f_{\max}-f_1}{S_2}) \geq k(\frac{tS_1-f_1}{\beta^* S_1 S_2} + \frac{f_1}{S_1})] = \Pr[\textbf{Binom}(k,0) \geq k(\frac{tS_1-f_{\max}}{\beta^* S_1 S_2} + \frac{f_{\max}}{S_1})] = \Pr[0 \geq k(\frac{tS_1-f_{\max}}{\beta^* S_1 S_2} + \frac{f_{\max}}{S_1})] = \Pr[\beta^* \leq \frac{f_{\max}-tS_1}{f_{\max}S_2}] = 0$.

Finally for the case of $f_1 \notin \{f_{\max} - S_2, f_{\max}\}$, first note that $f_1 = tS_1$ is a valid f_1 value in the sense that it satisfies Equation 16, since $f_{\max} - S_2 < tS_1 < f_{\max}$. Next by Theorem 2, we must have $error(\mathsf{L}_{\beta^*}, \mathbb{A}_{f_1,f_{\max}-f_1}) = err^*$, when $f_1 = tS_1$. To prove $error(\mathsf{L}_{\beta^*}) \leq err^*$, it suffices to show that for all $f_1 \notin \{f_{\max} - S_2, f_{\max}\}$, the error of L_{β^*} is maximized exactly when $f_1 = tS_1$. By Equation 10, for

all $f_1 \notin \{f_{\max} - S_2, f_{\max}\}$, we have $error(\mathsf{L}_{\beta^*}, \mathbb{A}_{f_1, f_{\max}-f_1}) = \Pr[Y \geq \lambda(f_1)]$. To maximize the error, the term $\lambda(f_1)$ should be minimized.

To determine when $\lambda(f_1)$ is minimized, we examine the derivative of $\lambda(f_1)$ in Equation 12, and the numerator $af_1 + b$ there. Lemma 8 later will prove $a \geq 0$, when $\beta = \beta^*$. Directly from the definition of β^*, we further have $af_1 + b = 0$ when $\beta = \beta^*$ and $f_1 = tS_1$. Now since $af_1 + b$ is linear with f_1, we must have $af_1 + b \leq 0$ when $f_1 < tS_1$, $af_1 + b = 0$ when $f_1 = tS_1$, and $af_1 + b \geq 0$ when $f_1 > tS_1$. In turn, this means that $\frac{\partial \lambda}{\partial f_1} \leq 0$ when $f_1 < tS_1$, $\frac{\partial \lambda}{\partial f_1} = 0$ when $f_1 = tS_1$, and $\frac{\partial \lambda}{\partial f_1} \geq 0$ when $f_1 > tS_1$. All this means that $\lambda(f_1)$ is minimized exactly when $f_1 = tS_1$.

We have now completed proving the theorem for $f_{\max} > tS_1$. We still need to prove the theorem for $f_{\max} < tS_1$. We claim that when $f_{\max} < tS_1$, we must have $error(\mathsf{L}_0) = 0 \leq error(\Pi)$. This is simply because $f_1 \leq f_{\max} < tS_1$ and $error(\mathsf{L}_0) = \Pr[f_1 \cdot \alpha \geq t] = \Pr[f_1 \cdot \frac{1}{S_1} \geq t] = 0$. □

We next prove the various technical lemmas invoked by the above proof for Theorem 5.

Lemma 6. *Consider any given $0 \leq f_1 \leq S_1$ and $0 \leq f_2 \leq S_2$. For any given general scheme Π_{σ_1,σ_2}, there must exist a TPL scheme L_β, such that $error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2}) \leq error(\Pi_{\sigma_1,\sigma_2}, \mathbb{A}_{f_1,f_2})$.*

Proof. We first construct an adversary $\mathcal{A} \in \mathbb{A}_{f_1,f_2}$, as follows. The adversary $\mathcal{A}$ orders all the large nodes, in decreasing order of $\frac{\sigma_1(s_p)}{s_p}$ ratios, where s_p is the node's stake. Next $\mathcal{A}$ corrupts nodes one by one in this sequence, until the total stake held by the corrupted large nodes reaches f_1. Let α' be the $\frac{\sigma_1(s_p)}{s_p}$ ratio of the last large node corrupted by $\mathcal{A}$ in this sequence.

Next, $\mathcal{A}$ orders all the small nodes, in decreasing order of their $\sigma_2(s_p)$ values (instead of $\frac{\sigma_2(s_p)}{s_p}$). Then $\mathcal{A}$ again corrupts nodes one by one, until the total stake held by the corrupted small nodes reaches f_2. Let β' be the $\sigma_2(s_p)$ value of the last small node corrupted.

We define the general scheme Γ' to be the general scheme $\Pi_{\sigma_1',\sigma_2'}$ with:

- $\sigma_1'(s_p) = \alpha' \cdot s_p$
- $\sigma_2'(s_p) = \beta'$

Note that Γ' is not a TPL scheme, since it is not normalized. As a theoretical corner case, if $\alpha' = \beta' = 0$, then $error(\Pi_{\sigma_1,\sigma_2}, \mathbb{A}_{f_1,f_2}) = 1$. In such a case, we trivially have $error(\mathsf{L}_1, \mathbb{A}_{f_1,f_2}) \leq error(\Pi_{\sigma_1,\sigma_2}, \mathbb{A}_{f_1,f_2})$ and we are done. So we only need to consider the case where at least one of α' and β' is positive, in the remainder of the proof.

Let Γ be the general scheme Π_{σ_1,σ_2}. We next prove $error(\Gamma', \mathcal{A}) \leq error(\Gamma, \mathcal{A})$. We first consider the small nodes. Under any given randomness for choosing committee members from the small nodes, let us compare Γ and Γ' under the adversary $\mathcal{A}$. For scheme Γ, let x_1^Γ and (respectively, x_2^Γ) be the weight of corrupted (respectively, non-corrupted) committee members that are small nodes. Similarly define $x_1^{\Gamma'}$ and $x_2^{\Gamma'}$. It is clear that $x_1^{\Gamma'} \leq x_1^\Gamma$ and $x_2^{\Gamma'} \geq x_2^\Gamma$. We move on to the large nodes. For scheme Γ, let y_1^Γ and (respectively, y_2^Γ) be the weight of corrupted (respectively, non-corrupted) committee members that are large nodes. Similarly define $y_1^{\Gamma'}$ and $y_2^{\Gamma'}$. By a similar argument, we have $y_1^{\Gamma'} \leq y_1^\Gamma$ and $y_2^{\Gamma'} \geq y_2^\Gamma$. This implies that under the given randomness, we must have $\frac{x_1^{\Gamma'} + y_1^{\Gamma'}}{x_1^{\Gamma'} + x_2^{\Gamma'} + y_1^{\Gamma'} + y_2^{\Gamma'}} \leq \frac{x_1^\Gamma + y_1^\Gamma}{x_1^\Gamma + x_2^\Gamma + y_1^\Gamma + y_2^\Gamma}$. Hence $error(\Gamma', \mathcal{A}) \leq error(\Gamma, \mathcal{A})$.

We next claim that $error(\Gamma', \mathbb{A}_{f_1,f_2}) = error(\Gamma', \mathcal{A})$. To see why, note that in Γ', every small node (if selected) will have the same weight in the committee. Hence it does not matter which small nodes are corrupted, as long as the total corrupted stake is f_2. Similarly, it does not matter which large nodes are corrupted, as long as the total corrupted stake is f_1. Hence we have $error(\Gamma', \mathbb{A}_{f_1,f_2}) = error(\Gamma', \mathcal{A})$. Putting everything together, we now have $error(\Gamma', \mathbb{A}_{f_1,f_2}) = error(\Gamma', \mathcal{A}) \leq error(\Gamma, \mathcal{A}) \leq error(\Gamma, \mathbb{A}_{f_1,f_2})$, or $error(\Gamma', \mathbb{A}_{f_1,f_2}) \leq error(\Gamma, \mathbb{A}_{f_1,f_2})$.

The total weight of all the committee members in scheme Γ' is $z = \alpha' S_1 + \beta' k$. Let $\beta = \frac{\beta'}{z}$. Essentially, L_β is obtained by normalizing Γ'. Now we have $error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2}) = error(\Gamma', \mathbb{A}_{f_1,f_2}) \leq error(\Gamma, \mathbb{A}_{f_1,f_2}) = error(\Pi_{\sigma_1,\sigma_2}, \mathbb{A}_{f_1,f_2})$. □

Lemma 7. *In the proof of theorem 5, the value β^* must exist, and we must have:*

$$0 < \frac{f_{\max} - tS_1}{f_{\max} S_2} < \beta^* < \frac{tS_1 - f_{\max} + S_2}{(1 - f_{\max})S_2} < \frac{1}{S_2}$$

Proof. First, in the proof of Theorem 5, the value β^* is used for the case when $S_1 > 0$ and $f_{\max} > tS_1$. Hence we inherit those two conditions. One can easily verify that $0 < \frac{f_{\max} - tS_1}{f_{\max} S_2} < \frac{tS_1 - f_{\max} + S_2}{(1-f_{\max})S_2} < \frac{1}{S_2}$. We only need to prove $\frac{f_{\max} - tS_1}{f_{\max} S_2} < \beta^* < \frac{tS_1 - f_{\max} + S_2}{(1-f_{\max})S_2}$.

Define the function $r(\beta) = a \cdot tS_1 + b$. By definition, β^* is the solution for the equation:

$$r(\beta) = a \cdot tS_1 + b \quad = \quad 0 \tag{17}$$

To prove the lemma, since $r(\beta)$ is a linear function of β, it suffices to show that $r(\frac{f_{\max} - tS_1}{f_{\max} S_2}) < 0$ and $r(\frac{tS_1 - f_{\max} + S_2}{(1-f_{\max})S_2}) > 0$. We have:

$$r\left(\frac{f_{\max} - tS_1}{f_{\max} S_2}\right) \quad = \quad \frac{S_1(t - f_{\max})(tS_1 - f_{\max})}{f_{\max} S_2} < 0$$

$$r\left(\frac{tS_1 - f_{\max} + S_2}{(1 - f_{\max})S_2}\right) \quad = \quad \frac{S_1(t - f_{\max})(tS_1 - f_{\max} + S_2)}{(1 - f_{\max})S_2} > 0$$

In the last step, we have used the fact that $tS_1 - f_{\max} + S_2 > tS_1 - f_{\max} + tS_2 = t - f_{\max} > 0$. □

Lemma 8. *In the proof of theorem 5, consider the variable a defined in Equation 13. Then we must have $a \geq 0$ when $\beta = \beta^*$.*

Proof. To prove this, we first solve for β^* based on Equation 17, and then plug in $\beta = \beta^*$ into the Equation 13. Doing so will eventually give:

$$a = \frac{S_1(t - f_{\max})}{-2f_{\max}^2 + (2S_1 t + S_2 + 1)f_{\max} - tS_1} \tag{18}$$

In Equation 18, the numerator of $S_1(t - f_{\max})$ is always non-negative. Define $r(f_{\max}) = -2f_{\max}^2 + (2S_1 t + S_2 + 1)f_{\max} - tS_1$, which is the denominator. The function $r(f_{\max})$ is quadratic with respect to $f_{\max}$, with the quadratic term being "$-2f_{\max}^2$". We know that $tS_1 < f_{\max} < t$. One can easily verily that $r(tS_1) = S_1 S_2 t \geq 0$, and that $r(t) = 2S_2 t(1 - t) > 0$. This then implies that $r(f_{\max}) > 0$ for all $f_{\max} \in (tS_1, t)$, which completes our proof for $a \geq 0$. □

E Non-proportional Weights Almost Always Help — Formal Proofs

Recall from Section 6 that non-proportional weights will help, as long as the optimal β for the TPL scheme is not 1. In order to formally show that non-proportional weights almost always help, here we first prove a necessary condition for the optimal β to be 1. Next, we show that this necessary condition is rather hard to satisfy.

A necessary condition.

Theorem 9. *Let ϵ be the minimum stake held by a node in the system. A necessary condition for the optimal β to be 1 is that at least one of the following two equations holds:*

$$f_{\max} - \frac{S_2}{2} = tS_1 \tag{19}$$

$$S_2 + f_{\max} - \epsilon < t \tag{20}$$

Proof. To prove, we need to show that if neither of the two equations holds, then the optimal β cannot be 1. Our proof depends on whether $f_{\max} \geq tS_1$. We first prove for the hard case of $f_{\max} \geq tS_1$. In such a case, the optimal β is the β^* defined via Equation 17. Based on that equation and with some tedious derivation, one can eventually show that in order for $\beta^* = 1$, we need to have either $tS_1 = f_{\max} - f_{\max}S_2$ or $tS_1 = f_{\max} - \frac{S_2}{2}$. However, since $f_{\max} - f_{\max}S_2 = f_{\max}S_1 < tS_1$, it is impossible for $tS_1 = f_{\max} - f_{\max}S_2$ to hold. Also, since Equation 19 does not hold, it is impossible for $tS_1 = f_{\max} - \frac{S_2}{2}$ to hold either. Hence β^* will not be 1.

Next if $f_{\max} < tS_1$, then as shown in the proof of Theorem 5, we must have $error(\mathsf{L}_0) = 0$. We next show that $error(\mathsf{L}_1) > 0$, which in turn implies that the optimal β cannot be 1. We let the adversary spend ϵ of its budget for corrupting the node with the smallest stake in the system, while the remaining $f_{\max} - \epsilon$ budget is used to corrupt large nodes. With some positive probability, we will have:

$$W_{\text{corrupt}} = (f_{\max} - \epsilon) + k \cdot \frac{S_2}{k} = S_2 + f_{\max} - \epsilon$$

Since Equation 20 does not hold, we must have $S_2 + f_{\max} - \epsilon \geq t$. This means $\Pr[W_{\text{corrupt}} \geq t] > 0$ and $error(\mathsf{L}_1) > 0$. □

Condition is hard to satisfy. A careful examination of Equation 19 and 20 reveals that this necessary condition almost never gets satisfied in practice. Specifically, Equation 19 requires a certain equality about t and $f_{\max}$ to *exactly* hold, which is unlikely in practice. For Equation 20, ϵ is usually close to zero, while S_2 tends to be around or above 0.4 in our experiments with real-world stake distributions. For Equation 20 to hold with such S_2, the gap between $f_{\max}$ and t needs to reach around 0.4. Since t is usually only $\frac{1}{2}$ or $\frac{1}{3}$, such a gap is unrealistic in practice.

F Adapted Proofs under Local Randomness

Recall the various definitions from Section 5.3 and Section 7.

Proof for Theorem 1 with local randomness.

Proof. Let random variable M (respectively, H) denote the total number of virtual nodes that belong to corrupted (respectively, non-corrupted) small-nodes and that get into the committee. By discussions in Section 7 and Footnote 10, M follows the binomial distribution of $\mathbf{Binom}(\frac{f_2}{u}, \frac{ku}{S_2})$. By [25, 40], as $u \to 0$, this binomial distribution becomes the Poisson distribution $\mathbf{Pois}(\frac{f_2}{S_2}k)$. Hence we have: $M \sim \mathbf{Pois}(\frac{f_2}{S_2}k)$. By a similar reasoning, we have $H \sim \mathbf{Pois}(\frac{S_2 - f_2}{S_2}k)$.

Next, since $0 \leq f_1 < tS_1$, we must have $S_1 > 0$. For all TPL scheme L_β, if $\beta > 0$, then we have:

$$\begin{aligned} error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2}) &= \Pr[f_1 \cdot \alpha + \beta S_2 \cdot \frac{M}{M+H} \geq t] \\ &= \Pr[\frac{M}{M+H} \geq \frac{t - f_1\alpha}{\beta S_2}] \\ &= \Pr[\frac{M}{M+H} \geq \frac{tS_1 - f_1}{\beta S_1 S_2} + \frac{f_1}{S_1}] \end{aligned} \tag{21}$$

Since $f_1 < tS_1$, when β decreases, the term $\frac{tS_1 - f_1}{\beta S_2 S_1}$ increases and thus $error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2})$ never increases. This implies $error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) \leq error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$, if $0 < \beta_1 \leq \beta_2$. Finally, if $\beta_1 = 0$, then we trivially have $error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) = \Pr[f_1 \cdot \alpha \geq t] = \Pr[f_1 \cdot \frac{1}{S_1} \geq t] = 0 \leq error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$. □

Proof for Theorem 2 with local randomness.

Proof. Define random variables M and H in the same way as earlier. If $S_1 > 0$, let us consider any TPL scheme L_β where $\beta > 0$. By Equation 21:

$$\begin{aligned} error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2}) &= \Pr[\frac{M}{M+H} \geq \frac{tS_1 - f_1}{\beta S_1 S_2} + \frac{f_1}{S_1}] \\ &= \Pr[\frac{M}{M+H} \geq t] \qquad (\text{when } f_1 = tS_1) \end{aligned}$$

Note that the last term does not depend on β. Hence:

$$error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) = error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2}) = \Pr[\frac{M}{M+H} \geq t] \tag{22}$$

Finally, if $S_1 = 0$, then we must have $\beta_1 = \beta_2 = 1$, and $error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) = error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$ trivially holds. □

Proof for Theorem 3 with local randomness.

Proof. Define random variables M and H in the same way as earlier. If $S_1 > 0$, let us consider any TPL scheme L_β where $\beta > 0$. By Equation 21, we have $error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2}) = \Pr[\frac{M}{M+H} \geq \frac{tS_1 - f_1}{\beta S_1 S_2} + \frac{f_1}{S_1}]$. Since $f_1 \geq tS_1$, when β decreases, the term $\frac{tS_1 - f_1}{\beta S_2 S_1}$ never increases and thus $error(\mathsf{L}_\beta, \mathbb{A}_{f_1,f_2})$ never decreases. This implies $error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) \geq error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$, if $0 < \beta_1 \leq \beta_2$.

Finally, if $\beta_1 = 0$ and $S_1 > 0$, then we trivially have $error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) = \Pr[f_1 \cdot \alpha \geq t] = \Pr[f_1 \cdot \frac{1}{S_1} \geq t] = 1 \geq error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$. If $S_1 = 0$, then we must have $\beta_1 = \beta_2 = 1$, and $error(\mathsf{L}_{\beta_1}, \mathbb{A}_{f_1,f_2}) \geq error(\mathsf{L}_{\beta_2}, \mathbb{A}_{f_1,f_2})$ trivially holds. □