THE NATIONAL UNIVERSITY
of SINGAPORE

School of Computing
Computing 1, 13 Computing Drive, Singapore 117417

TRA9/24

Robust and Low-degree Overlay for Secure Flooding Against Resource-bounded Adversaries

Yucheng Sun, Ruomu Hou, and Haifeng Yu

September 2024

Technical Report

Foreword

This technical report contains a research paper, development or tutorial article, which has been submitted for publication in a journal or for consideration by the commissioning organization. The report represents the ideas of its author, and should not be taken as the official views of the School or the University. Any discussion of the content of the report should be sent to the author, at the address shown on the cover.

TAN Kian Lee
Dean of School

Robust and Low-degree Overlay for Secure Flooding Against Resource-bounded Adversaries

(Technical Report)

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

Ruomu Hou
Department of Computer Science
National University of Singapore
Republic of Singapore
houruomu@comp.nus.edu.sg

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

***Abstract*—The security of large-scale blockchains relies on successful message flooding among the honest parties, on an overlay topology. The crux of doing such flooding successfully is to have a *robust* and *low-degree* overlay topology: *Robust* means that even if the malicious parties all refuse to relay messages, the remaining honest parties should still constitute a connected component in the overlay network. *Low-degree* means that the nodes in the overlay network should have relatively small node degrees. The central challenge of designing such robust and low-degree topology is that in permissionless blockchain context, the adversary is often bounded by *resource* (such as computation power or stake), rather than by the total number of malicious parties. We show that existing works of designing robust overlay against such *resource-bounded adversaries* all require excessively large node degrees (e.g., 25000 or more under real-world settings). As our main contribution, we propose a novel LOR overlay topology that is robust against such resource-bounded adversaries. Our design is *the very first* such design with practically-feasible node degrees (e.g., 200 to 400 under real-world settings).**

***Keywords*-Permissionless blockchains, Secure flooding, Overlay network, Proof-of-Stake.**

I. Introduction

A. Background and Motivation

Blockchains and consensus. Over the past two decades, blockchains have enjoyed unprecedented growth. The central functionality provided by a blockchain is a shared distributed ledger that all honest parties agree on, despite malicious attacks. To achieve this, almost every blockchain today, at its core, relies on a byzantine consensus mechanism/protocol [1]–[13].

Secure flooding. All these consensus mechanisms, implicitly, rely on successful message propagation among the honest parties. For example, Bitcoin [1] relies on each new block being propagated to at least a majority of all the honest parties, so that the parties can continue mining by extending that block. Algorand [4] relies on the voting messages in their BA$*$ protocol being seen by most honest parties.

In large-scale blockchains, it is infeasible for a node to directly send a message to every other node in the system. Hence such message propagation is typically done on an *overlay network*. In an overlay network, each node has a number of neighbors, and is responsible for pushing the message to its neighbors [4], [5], [7], [9], [14], [15]. Following [16], [17], we call such message propagation as *flooding*. The flooding mechanism is *secure*, if the message propagation can reach most honest parties, despite the malicious parties in the overlay network.

A blockchain is secure, only if *both* the consensus mechanism *and* the flooding mechanism are secure. Unfortunately, there has been a paucity of work, as well as a lack of deep understanding, on secure flooding. In fact, provably secure flooding has not attracted much attention from blockchain researchers, until very recently [16]–[18].

Overlay topologies. To achieve secure flooding, all these recent efforts [16]–[18] observe that the crux is to design a *robust* and *low-degree* overlay topology:

- *Robust*: Being *robust* means that even if the malicious parties all refuse to relay messages, most honest parties still constitute a connected component to enable proper message propagation. In other words, most of the honest parties should still be connected with each other, even if all the malicious parties are deleted from the topology.
- *Low-degree*: This means that the nodes in the overlay network should have relatively small node degrees. If the degree reaches, for example, as large as several thousands, then during flooding a node will not have sufficient resources (such as bandwidth) to push the messages to its outgoing neighbors (if they do not yet have the message). This in turn invalidates the robustness guarantees. In fact, if we were not concerned with degree, then using a complete graph as the topology would already be robust. A further corollary is that it is the *maximum* node degree

This research is partly supported by the Ministry of Education, Singapore, under its MOE Academic Research Fund Tier 2 (research grant number: MOE-T2EP20221-0002). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

in the topology, rather than the *average* node degree, that matters. This is simply because the problem of not having sufficient resource to propagate messages will arise on high-degree nodes first, who are the bottleneck.

Resource-bounded adversary. Existing works [17], [18] have further observed that designing robust and low-degree topologies is challenging, for permissionless blockchains. This is because in permissionless blockchains, the adversary can freely introduce new (malicious) parties/nodes into the system. In turn, we cannot assume an upper bound on the *number (or fraction) of malicious parties/nodes* in the topology.

The only assumption we can make on the adversary is its limited *resource*. For example, in PoW/PoS blockchains, the adversary is bounded by its computational power or stake (i.e., amount of cryptocurrency). Namely, the aggregate computational power or stake held by all the malicious parties will be bounded. For convenience, we call such adversary as *resource-bounded* adversary.

An example. To highlight the above challenge, consider a resource-bounded adversary that controls 30% of the total stake in the system. The adversary can introduce a large number of malicious parties, where each malicious party holds a relatively small amount of stake. By doing so, the fraction of malicious parties/nodes in the overlay network can easily exceed 30%, and may even approach 100%.[1] This example also implies that a simple random topology, where each party connects to some uniformly randomly chosen other parties, is *not* robust.

B. Existing State-of-the-art Approaches

To our knowledge, currently there are only two overlay designs [17], [18] that are robust against resource-bounded adversaries. Both are quite recent, and both remain robust in our earlier example, where the malicious parties comprise close to 100% of the total number of parties, but control only 30% of the stake.

To achieve such a strong guarantee, these designs [17], [18] construct topologies among all the parties, based on how much stake each party holds. Specifically, each party in the overlay topology will simulate a number of *virtual nodes*. The more stake a party holds, the more virtual nodes it simulates. Section III-A reviews these designs in more detail.

It is worth noting that these designs [17], [18] are all *theory-oriented*. There have never been any concrete numerical results for their node degrees under real-world stake distributions.

C. Our Results

Quantifying node degrees in [17], [18]. As our first step, we experimentally quantify the node degrees needed by the designs in [17], [18], under the real-world stake distributions of 6 major cryptocurrencies and practical settings. Perhaps somewhat surprisingly, our results reveal that their maximum node degrees easily reach **25000 or more**. This makes these designs unusable in practice, despite that they do hope to achieve low node degrees. Part of the reason behind such large degrees is that in their designs, sometimes a party needs to simulate *thousands* of virtual nodes.

Novel LOR overlay design. As our primary contribution, we propose a novel and elegant design for robust overlay called LOR (i.e., low-degree and robust), against resource-bounded adversaries. We formally prove that our LOR topology is robust, against all possible strategies of the adversary. At the same time, our topology has practically-feasible node degrees: Under the same settings as earlier, our design only needs maximum node degree of around **200 to 400**. Our degrees are quite feasible in practice: Many nodes in the Ethereum overlay already have degrees of a few hundreds [19].

Asymptotically, the maximum node degree in our LOR topology is $O(\log n)$, with n being the total number parties. In comparison, the maximum degrees in [17], [18] are at least $O(s_{\max} \cdot n)$ and $O(s_{\max} \cdot n \log n)$, respectively.[2] Here $s_{\max}$ is the maximum stake held by a node. Note that as a sharp contrast, our maximum node degree only contains a "$\log n$" term instead of an "n" term. Finally, same as [17], [18], our topology has small $O(\log n)$ diameter asymptotically.

Our techniques. The designs in [17], [18] resort to using virtual nodes (which leads to large degrees), in order to prevent the adversary from exploiting the *stake disparity* among the parties (i.e., some parties are much richer/poorer than others). In comparison, our design completely avoids using virtual nodes. We deal with stake disparity by first grouping the parties into groups, where each group internally has only *bounded-disparity*. Now consider any subset of nodes in the group, where the subset corresponds to x (y) fraction of the total nodes (stake) in the group. We then repeatedly exploit a simple yet powerful property: In a *bounded-disparity group*, the value y must be both upper and lower bounded by certain functions of x. Doing so eventually enables us to design robust and low-degree topology to offer both *intra-group* and *inter-group* connectivity.

Summary of contributions. To summarize, our key contribution is a novel robust overlay design against resource-bounded adversaries:

- Our design is *the very first* such design that comes with practically-feasible node degrees: Under real-world settings, the maximum node degree in our design is around **200 to 400**, as compared to **25000 or more** in prior works [17], [18].
- Asymptotically, the maximum node degree in our topology is only $O(\log n)$, as compared to $O(s_{\max} \cdot n)$ and $O(s_{\max} \cdot n \log n)$ in [17], [18].
- We formally prove that our topology is robust, against *all possible adversarial strategies*.

[1] Note that one cannot simply view all nodes holding small amount of stake as malicious nodes: First, some honest nodes may also hold only small amount of stake. Second, the adversary may use a different attack strategy.

[2] The maximum degree in [18] is $O(\frac{s_{\max}}{s_{\min}})$, where $s_{\min}$ is the minimum stake held by a node. Since $s_{\min}$ is at most on the order of $\frac{1}{n}$, the term $O(\frac{s_{\max}}{s_{\min}})$ is at least $O(s_{\max} \cdot n)$.

Roadmap. Section II presents our system/adversary model. Section III quantifies node degrees in [17], [18]. Section IV through VI elaborates our design, its security proof, and its experimental results.[3] Section VII discusses further issues. Section VIII elaborates on related works.

II. System Model and Adversary Model

We will mainly use PoS blockchains as an example context. We will later easily generalize to PoW blockchains.

Resource-bounded adversary. Without loss of generality, we assume that the total stake (i.e., cryptocurrency) in the PoS blockchain is 1.0. Namely, we normalize the amount of stake. Among this, the (resource-bounded) adversary controls up to f stake. This means that the total stake held by all the malicious parties is at most f. We also call f as the *budget* of the adversary. These malicious parties may either be honest parties who have been corrupted by the adversary, or new malicious parties created by the adversary (e.g., via sybil attack [21]). The adversary may freely *redistribute* the stake among the malicious parties. We use n to denote the total number of honest parties and malicious parties. Among these n parties, there is no constraint on the fraction of malicious parties, as long as they collectively hold no more than f stake. A malicious party is fully byzantine, and all malicious parties collude.

Each party corresponds to a public/private key pair. The parties aim to form an overlay network among themselves. Hence each party also corresponds to a *node* in the overlay. We will use "party" and "node" interchangeably. We always view the overlay topology as a directed graph.

Robustness: (ϵ, δ)-guarantee. Following [18], we say that the overlay topology is *robust* if it provides the (ϵ, δ)-*guarantee*: With at least $1-\delta$ probability, all honest nodes form a strongly connected component in the overlay topology, except a small fraction of honest nodes whose total stake does not exceed ϵ. We also call such honest nodes as being *eclipsed*.

Coretti et al. [18] have shown that in blockchains, it usually suffices for message propagation to reach the vast majority of the honest nodes, allowing ϵ eclipsed honest stake. This is because the consensus mechanism can often afford to have a small fraction of honest nodes who do not get all messages. Fundamentally, in practice, blockchains already need to accommodate a small fraction of honest nodes with temporary network connection problems. Those eclipsed honest nodes themselves can use existing mechanisms [22]–[24] to detect that they are eclipsed, and can then preserve safety by temporarily giving up liveness.

Constructing topology based on stake. Same as [17], [18], we construct the topology based on the *stake distribution* of all the parties. This distribution describes how much stake each party holds. The stake distribution is part of the blockchain state — by checking the blockchain state, one can easily determine how much stake each party holds. Hence it is *public information*, available to all parties including the adversary. The stake distribution can change over time.

To construct an overlay at time t_1, we use the stake distribution $\mathbb{D}$ at time t_2, where t_2 is slightly before t_1, so that all nodes agree on what $\mathbb{D}$ was at t_2. Based on $\mathbb{D}$, our design will generate a random topology. When doing so, we (our design) obviously do not know which parties are honest/malicious. It is well-known that both PoS and PoW blockchains can provide a beacon generation functionality [2]–[4], which periodically generates a random public beacon seen by all parties. Such beacons are often used, for example, for committee selection in those blockchains. We will use this beacon as the randomness when constructing our topology.

With the beacon and $\mathbb{D}$, each party can locally determine which parties should be its outgoing neighbors in the overlay. During flooding, the party can then push the message to these outgoing neighbors. When doing so, a party will need to know the IP addresses of its neighbors. Section VII will explain how to do so, without compromising node *anonymity*.

Adaptivity of the adversary. Same as [17], we expect the overlay topology to be short-lived and periodically re-formed, for example, upon the release of every new beacon. The overhead of re-forming the topology is minimum: When a node needs to forward a message and if a new beacon has been released, then the node just needs to determine its new neighbors in the new topology, and forward the message to those new neighbors instead of its old neighbors. A node does not need to always keep TCP connections with its neighbors.

Following [17], we assume that the adversary is *mildly-adaptive*: While the adversary can corrupt nodes on the fly, it takes some time to do so. In particular, with the topology being short-lived, we assume that the adversary cannot cherry-pick nodes to corrupt, based on the randomness (i.e., the beacon) used to construct the random topology.

Incoming vs. outgoing degrees. As explain in Section I, the maximum node degree in the overlay topology should be small. A node has an incoming degree and an outgoing degree. We will provide results on both incoming degrees and outgoing degrees. But to facilitate discussion, we will use *maximum outgoing degree* as the main metric — unless otherwise mentioned, whenever we mention degree, it refers to *outgoing* degree.

III. Quantifying Node Degrees in [17], [18] under Real-world Stake Distributions

The designs in [17], [18] are all theory-oriented. This section aim to quantify, for the first time, the maximum node degrees in these designs under real-world stake distributions.

A. Brief Review of Designs in [17], [18]

We first quickly review these two designs.

Coretti et al.'s design [18]. In this design, the parties are categorized into *small-stake parties* and *non-small-stake parties*, depending on whether a party holds less than a certain threshold t amount of stake. Their design aims to provide good

[3]The source code for the experiments in this paper is available at [20].

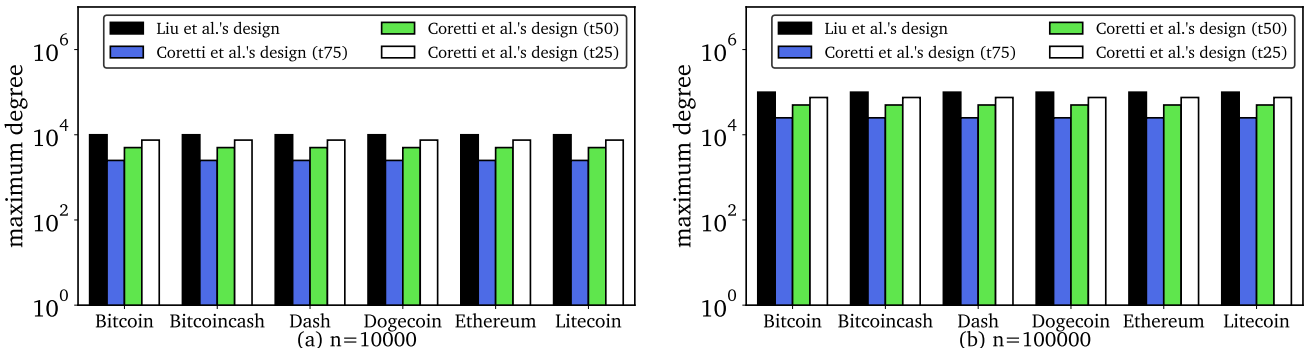

Fig. 1. Quantifying maximum node degrees in Liu et al.'s design [17] and Coretti et al.'s design [18], under real-world stake distributions of 6 major cryptocurrencies, for realistic settings of $f = 0.2$, $\epsilon = 0.1$, and $\delta = 0.01$.

connectivity among the non-small-stake parties *only*. (This is unique to their design [18] — our design does not make such compromise.)

Conceptually, a non-small-stake party with s stake will simulate $\lceil \frac{s}{t} \rceil$ *virtual nodes*, where each virtual node has λ random outgoing edges. Here $\lambda \geq 1$ is a parameter in their design. Specifically, a non-small-stake party A with s stake creates $\lambda \cdot \lceil \frac{s}{t} \rceil$ outgoing directed edges. For each edge, a non-small-stake party with stake x will be chosen as the destination with $\frac{x}{y}$ probability, where y is the total stake of all non-small-stake parties. It is possible for A to choose itself, and for the same party to be chosen as the destination multiple times. Such self-loops and duplicate edges do *not* count toward node degree. The key intuition of their design is that if the adversary corrupts f fraction of stake, then it must corrupt only about f fraction of all the virtual nodes. This is despite that the faction of corrupted parties can well exceed f.

Small-stake nodes are not included in their design of the topology. The reason is that otherwise t may be excessively small, inflating the number of virtual nodes simulated by other nodes. For all analysis/experiments of their design, we will not count the small-stake nodes toward the ϵ term — doing so only makes their results better.

Liu et al.'s design [17]. Liu et al.'s also use the idea of virtual nodes, and observe that a node with very small stake can cause other nodes to simulate a very large number of virtual nodes. To overcome this, conceptually they let a node with stake s simulate $\lceil s \cdot n \rceil$ virtual nodes. Let s_1 through s_n be the stake of the n nodes, respectively. Recall that $\sum_i s_i = 1$. We then have $\sum_i \lceil s_i \cdot n \rceil \leq \sum_i (s_i \cdot n + 1) = n + \sum_i (s_i \cdot n) = 2n$. This means that the total number of virtual nodes in the system will be at most $2n$, regardless of how small stake a node may have. Their design also has a parameter $\lambda \geq 1$, which corresponds to the outgoing degree of a virtual node.

Specifically in their design, a node A with stake s will create $\lambda \cdot \lceil s \cdot n \rceil$ outgoing edges. The destinations of these edges are chosen without replacement: When A creates an edge, let the set E be all the nodes who are not A and who have not been chosen so far. Then a node in E with stake s' will be chosen as the destination of that edge with $\frac{\lceil s' \cdot n \rceil}{\sum_{y \in E} \lceil (y\text{'s stake}) \cdot n \rceil}$ probability.

Liu et al.'s design [17] focuses on the case of $\epsilon = 0$. In comparison, both Coretti et al.'s design [18] and our design are for positive ϵ. Section II has explained why having small positive ϵ usually suffices. Nevertheless, to gain deeper insights into the subject, we still obtain results for Liu et al.'s design in our experiments.

B. Node Degrees in [17], [18]

Having reviewed the designs in [17], [18], we now experimentally quantify the maximum node degrees in these designs.

Our experiments with $n = 10000$ nodes use the real-world stake distributions of the top 10000 nodes in 6 popular cryptocurrencies — Bitcoin, Bitcoincash, Dash, Dogecoin, Ethereum, and Litecoin. These blockchains are not necessarily all PoS blockchains, but the notion of stake distribution is nevertheless still well-defined in these blockchains. These stake distributions are publicly available, from [25] for Ethereum and from [26] for the remaining 5 cryptocurrencies. These data unfortunately are only for 10000 nodes. For larger system size[4] of $n = 100000$, we use the stake histograms from a measurement study [28] of these 6 cryptocurrencies. We draw 100000 random nodes from those stake histograms as the stake distribution.

Section III-C will explain what λ value we use in the experiments. Coretti et al.'s design [18] excludes small-stake nodes (whose stake is less than t) from the overlay topology, without providing specific ways of choosing t. But regardless, one would expect that the overlay should not exclude too many nodes. Hence we consider t values such that x fraction of the nodes are excluded from the overlay, for $x = 25\%$, 50%, and 75%. We will use "$t25$", "$t50$", and "$t75$" to label these three versions.

Figure 1 presents our experimental results on the maximum node degrees. Under all the 12 cryptocurrency stake distributions, the maximum node degree in Liu et al.'s design [17] always exceeds $0.99n$. Similarly, the maximum node degree in Coretti et al.'s design [18] always exceeds $0.99 \times (1-x)n$, in all three versions ("$t25$", "$t50$", "$t75$"). Note that there are total only $(1-x)n$ nodes in their overlay topology. This implies that the node with the maximum degree *has an edge to almost every node in the overlay network*. In particular, for

[4]As a reference point for system size, measurement study [27] by Donet et al. discovers 111475 Bitcoin full nodes.

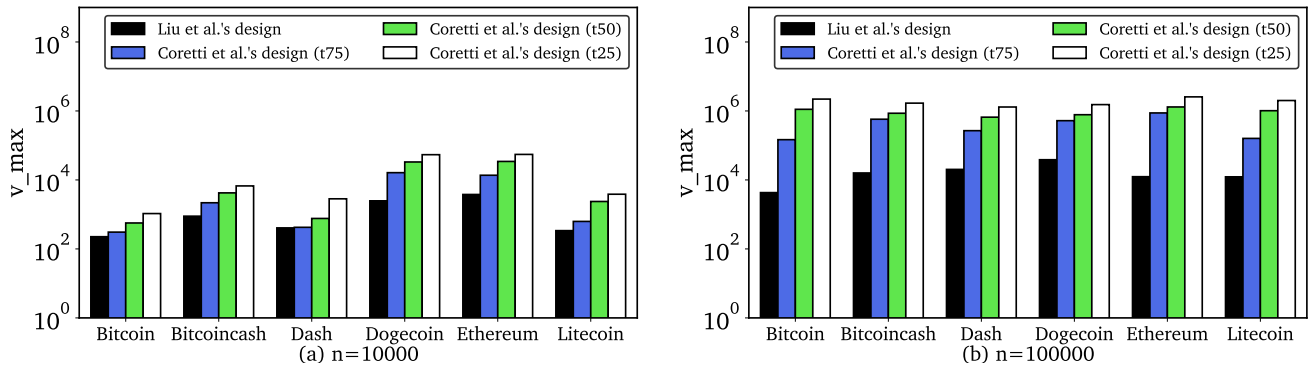

Fig. 2. Quantifying $v_{\max}$ in Liu et al.'s design [17] and Coretti et al.'s design [18], under real-world stake distributions of 6 major cryptocurrencies.

$n = 100000$, the maximum degrees in both designs [17], [18] reach roughly 25000.

C. Deeper Insights and Understanding

To gain further insights into these results, observe that in both designs [17], [18], the number of edges created by a (real) node equals λ times the number virtual node simulated. We now quantify these two terms separately.

Number of virtual nodes simulated. Let $v_{\max}$ be the maximum number of virtual nodes that a (real) node needs to simulate. Figure 2 plots the value of $v_{\max}$ in Liu et al.'s design [17] and in Coretti et al.'s design [18]. The figure shows that $v_{\max}$ easily reaches 10^3 or 10^4.

Value of the parameter λ. For Coretti et al.'s design [18], the proof of Lemma A.7 in their full paper [29] requires $\lambda > \frac{3(a+2\ln 2)}{\kappa^2 b^2}$, where $a \geq -\ln 2$, $\kappa = \frac{1}{2}$, and $b = \frac{\epsilon}{16}$. Under the experimental settings, the formula gives $\lambda > 2.1 \times 10^5$ (for all n). Our experiments simply use $\lambda = 2.1 \times 10^5$, which can only make their results better. For Liu et al.'s design,[5] Corollary 3 of [17] proves that the error probability is bounded by $ne^{-\frac{(n-1)\lambda}{n\cdot 24}} + e^{-(1-f)\cdot n\cdot(\frac{\lambda\cdot(1-f)}{54}-2)} + (2\cdot n\cdot(e^{-\frac{\lambda\cdot(1-f)}{108}} + (6\cdot\log(\frac{n\cdot 6}{\lambda\cdot(1-f)})+1)\cdot e^{-\frac{7\cdot\lambda\cdot(1-f)}{648}}))$. Under the experimental settings, solving this formula gives $\lambda = 1816$ (for $n = 10000$) or $\lambda = 2409$ (for $n = 100000$). These are also the λ values we use in our experiments. These λ values show that, even if a (real) node simulates only one virtual node, it already needs to have 1816 outgoing edges or more.

Better parameter values will not help. These large λ values could be artifacts of their pessimistic analysis. However, fundamentally by their designs [17], [18], λ is at least 1. Now *even with $\lambda = 1$, which is potentially insecure*, a node in their topologies may still need to create over 10^4 outgoing edges, simply because $v_{\max}$ reaches 10^4. This implies that searching for tighter parameter values, as we later do in our design, will not help in their designs.

[5]The analysis in [17] is for $\epsilon = 0$ only — they do not provide analysis for $\epsilon > 0$. It is unclear how to determine λ for $\epsilon > 0$. Hence we can only use this analysis. Under $\epsilon > 0$, λ will likely decrease. But we will quickly see that even with $\lambda = 1$ in their design, the node degree remains excessive.

IV. Our Novel LoR Topology Design

A. Our Overall Idea

The crux of designing robust overlay, against a resource-bounded adversary, is to deal with *stake disparity* among the nodes. Due to such disparity, the fraction of malicious nodes in the overlay can approach 100%, despite that f is only for example, 0.3. To deal with stake disparity, prior works [17], [18] resort to using virtual nodes, which however leads to large node degrees.

To deal with stake disparity without resorting to virtual nodes, our design first groups the nodes into *groups*, so that the nodes within each group have *bounded* stake disparity. (We actually group nodes by "weight" and not "stake". But to help understanding, for now assume that we group by stake.) The total stake in a group can be smaller than f. Because of this, unfortunately, the fraction of malicious (honest) nodes in a group can still approach 100% (0%). To deal with this, we consider all the groups:

- Let H be the set of honest nodes in a given group. If H constitutes only a rather small fraction of the *total number of nodes* in the group, then given the *bounded disparity* within the group, the stake held by H must also be only a small fraction of the *total stake* in that group. In turn, we can afford to simply throw those honest nodes in H into the small ϵ term.
- Otherwise the group still has a reasonable fraction of honest nodes. Our design will then ensure robust *intra-group* and *inter-group* connectivity for all such groups, while using low node degrees. (Our design does not need to know which groups fall into this category.)

B. Details of Our LoR Topology

We now present the details of our design. Table I summarizes our key notations.

Step 1: Grouping based on weights. For each node with s stake, our design assigns it a *weight* of $s + \frac{f}{10n}$. (We later explain why $\frac{f}{10n}$.) Since s must be in $(0, 1.0]$, the weight of a node must be in $(\frac{f}{10n}, 1+\frac{f}{10n}]$. We partition all the nodes into z *groups*, where $z = \lfloor \log_g((1+\frac{f}{10n})/(\frac{f}{10n}))\rfloor + 1 = O(\log n)$. For $1 \leq j \leq z$, the j-th group contains nodes with weights in

symbol	meaning
n	number of parties/nodes in the system
f	total stake held by malicious parties, as a fraction of all stake in the system
ϵ, δ	from the definition of (ϵ, δ)-guarantee
g	protocol parameter: Nodes in a group differ by at most g factor in weight.
k	protocol parameter: A node creates k outgoing edges to form the intra-group topology.
l	protocol parameter: Each group has l leaders.
z	total number of groups in our design, and $z = \lfloor \log_g((1 + \frac{f}{10n})/(\frac{f}{10n})) \rfloor + 1$

TABLE I
OUR KEY NOTATIONS.

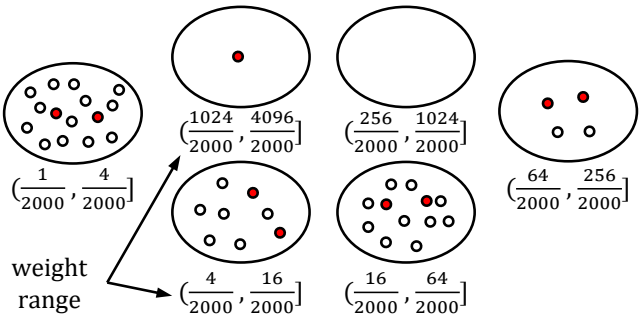

Fig. 3. Example grouping under $g = 4$, $l = 2$, and $\frac{f}{10n} = \frac{1}{2000}$.

the range of $(\frac{f}{10n} \cdot g^{j-1}, \frac{f}{10n} \cdot g^j]$. Here $g = O(1)$ is a protocol parameter (e.g., $g = 4$). Figure 3 illustrates such grouping.

Intuitions of Step 1. Our grouping ensures that the weights of the nodes in the same group differ by at most a factor of g. If g is not too large, then the nodes in the same group will not have much disparity in terms of weight.

We group based on weights and not stake, because a node's stake can be excessively small. In fact, the adversary may intentionally create nodes with very small stake. In contrast, a node's weight is at least $\frac{f}{10n}$, leading to at most $O(\log n)$ groups.

Recall that the adversary corrupts up to f stake, and aims to eclipse more than ϵ honest stake. We can easily translate these to weight: First, to eclipse more than ϵ stake, the adversary obviously needs to eclipse more than ϵ weight. Second, with its budget of f on stake, we claim that the adversary can at most corrupt $1.1f$ weight. This is because each node only gets $\frac{f}{10n}$ weight "for free", and the total amount of "free" weight that we give out is $\frac{f}{10n} \cdot n = 0.1f$. Hence the total weight that the adversary can afford to corrupt is at most $f + 0.1f = 1.1f$. Making the $\frac{f}{10n}$ term smaller decreases the "free" budget given to the adversary, at the cost of increasing the number of groups. We hence take $\frac{f}{10}$ as a simple trade-off.

Step 2: Intra-group topology. Within each group, each node creates $k = O(\log n)$ outgoing (directed) edges, where k is a protocol parameter. The k destinations are chosen uniformly randomly, without replacement, from all the remaining nodes in the group.

Intuitions of Step 2. As long as g is not too large, nodes in the same group have similar weights. To have a robust topology among these largely homogeneous nodes, it suffices to form a simple random graph. Having $k = O(\log n)$ enables the intra-group topology to have good connectivity, as long as there is still some constant fraction of honest parties left in that group.

Since the total weight of a group can be small, the adversary may potentially corrupt most/all parties in a given group. In such case, the intra-group topology will no longer be connected. But on the other hand, in such a case, the remaining honest parties only constitute a small fraction of the group. Then we can easily count those honest parties toward ϵ.

Step 3: Choosing leaders. From each group, we select up to $l = O(1)$ (e.g., $l = 30$) leaders without replacement (see Figure 3, where the shaded/red nodes are leaders). Roughly speaking, the parameter l is chosen such that, as long as there is still some constant fraction of honest parties left in that group, the group is likely to have some honest leader. For a given group, to choose one more leader, a non-leader party with x weight is chosen with $\frac{x}{y}$ probability, where y is the total weight of all non-leader parties currently in that group. If a group has no more than l parties, then all of them will be leaders. Finally, all the leaders from all the groups form a complete graph. We view each undirected edge in this complete graph as a pair of directed edges.

What if adversary targets the leaders. The leaders provide inter-group connectivity. An immediate question is what if the adversary targets the leaders. This turns out not to be a problem: Recall from Section II that following prior works on secure flooding [16], [17], we consider mildly-adaptive adversaries, and the topology is short-lived and periodically re-formed. This implies that the mildly-adaptive adversary will not have sufficient time to corrupt the leaders, after seeing the randomness used to select the leaders. Putting it another way, by the time the adversary corrupts the leaders, the topology has already expired and new topology (with new leaders) has already been formed.

Furthermore, our use of leaders is similar to the use of *committee* in the consensus mechanisms of various PoS blockchain designs [2], [3], [14], [15], [30]. For these committee-based designs to be secure, one already needs to assume the adversary to be mildly-adaptive. Otherwise the adversary can similarly target the committee members.

C. Intuitions on Why Our LoR Topology is Robust

Having explained our design, we now quickly get some intuitions on why it is robust. Formal proofs will be given in Section V.

We call a group as *surviving*, if the fraction of honest weight in that group (out of the total weight in that group) is at least some value c (e.g., $c = 0.05$) to be determined later.

- **Non-surviving groups.** Since a non-surviving group only has less that c fraction of honest weight, the total honest weight in all non-surviving groups, even added together, will be small. We can then throw this part to the ϵ term.
- **Surviving groups with no surviving leader.** With appropriate parameter $l = O(1)$, a surviving group usually contains an honest leader. But with some small probability, it might not. We can later show that the total honest weight in these leaderless surviving groups is small, and can be thrown into the ϵ term.
- **Surviving groups with surviving leader.** With appropriate parameter $k = O(\log n)$, for each surviving group containing some honest leader, we can show that most of the honest nodes (weights) in that group are reachable to/from some honest leader in that group. Recall that all the leaders from all groups form a complete graph. Hence most of the honest nodes (weights) in all surviving groups with surviving leaders will belong to the same strongly-connected component, and are not eclipsed.

D. Parameter in Our LoR Topology

Our LoR topology has three parameters (g, k, l). Larger k and l make the topology more robust, at the cost of larger node degree. Smaller g makes each group more homogeneous, and hence makes the topology more robust. But smaller g also increases z (i.e., total number of groups), and hence node degree.

The maximum node degree in our LoR topology is essentially determined by the parameters (g, k, l). Specifically, let l_j be the number of leaders in the j-th group. The j-th group has at most $(1 + \frac{f}{10})/(\frac{f}{10n} \cdot g^{j-1})$ nodes, since the total weight in the systems is $1 + \frac{f}{10}$. We hence must have $l_j \leq \min(l, (1 + \frac{f}{10})/(\frac{f}{10n} \cdot g^{j-1}))$. The maximum outgoing degree of a node in our LoR topology is then at most $k + \sum_{j=1}^{z} l_j$. This is because a node has k edges pointing to other nodes in its group. If the node is further a leader, then it will have at most $\sum_{j=1}^{z} l_j$ additional edges pointing to other leaders.

Finally, to construct the LoR topology, one needs concrete values for (g, k, l). Because nodes with different stakes play similar roles in our topology, we are able to make (g, k, l) *independent* of the stake distribution among the nodes. Our (g, k, l) values will hence only depend on n, f, ϵ, and δ, and not on the stake distribution.

E. Finding Suitable Parameters

A typical approach to obtain appropriate parameter values (g, k, l) would be to derive formulae for computing (g, k, l), based on (n, f, ϵ, δ), via an analysis showing that the resulting topology will achieve the desired guarantees. But to get tighter parameter values, we instead use an explicit algorithm to search for parameters, in a way similar to constraint optimization. For space constraints, we defer the details to Appendix B.

Finally, we emphasize that for existing designs [17], [18], using tighter parameter values will not help, as shown in Section III-C. This means that our improvement is not (only) due to tighter parameters.

V. Security Analysis of Our Design

Theorem 1 below proves the robustness of our LoR topology, against all possible adversarial strategies. We will prove that the LoR topology achieves the (ϵ, δ)-guarantee, and also prove the resulting node degree and diameter in our topology.

The proof of Theorem 1 invokes Lemma 7. For space constraints, we defer Lemma 7 to Appendix A. The proof of Lemma 7 is non-trivial, and needs to build upon several existing results about random graphs in [17], [31].

Theorem 1. *Consider any given $f \in [0, 1)$, $\epsilon \in (0, 1]$, $\delta \in (0, 1]$. There exists $g = O(1)$, $k = O(\log n)$, and $l = O(1)$, such that with such parameter values, our* LoR *topology achieves the (ϵ, δ)-guarantee, under all possible adversarial strategies and all possible stake distributions (including adversarially generated ones). Furthermore, our* LoR *topology has $O(\log n)$ maximum node degree and except for δ probability, $O(\log n)$ diameter.*

Proof: Let $g = 2$. For any given group G, let $W(G)$ be the total weight of nodes in G, and $W_h(G)$ be the total weight of the honest nodes in G. We say that G is a *surviving group* if $\frac{W_h(G)}{W(G)} \geq w_0$, where $w_0 = \frac{\epsilon}{2.2f+\epsilon}$. Otherwise G is a *non-surviving* group. If a surviving group contains some honest leader, we call it a *surviving group with surviving leader* (or *SGL*). Otherwise it is a *surviving group with no surviving leader* (or *SGN*). We reason about these 3 kinds of groups separately:

- **Non-surviving groups:** Lemma 2 later will prove that the total weight of all the honest nodes in all the non-surviving groups is at most $\epsilon/2$.
- **SGN:** Lemma 3 later will prove that except for $\delta/2$ probability, the total weight of all the honest nodes in all the SGNs is at most $\epsilon/2$.
- **SGL:** Lemma 7 later will prove that except for $\delta/2$ probability, for every SGL, all honest nodes in that SGL form a strongly-connected component, with a diameter of $O(\log n)$. An SGL contains at least one honest leader. Recall that all leaders from all groups form a complete graph. Hence except for $\delta/2$ probability, all honest nodes in all SGLs form a strongly-connected component, with a diameter of $O(\log n)$.

Combining the above 3 cases then immediately leads to the (ϵ, δ)-guarantee. Furthermore, the maximum node degree is at most $k + zl = O(\log n)$. Finally, it directly follows from Lemma 7 that except for δ probability, the topology has a diameter of $O(\log n)$. ■

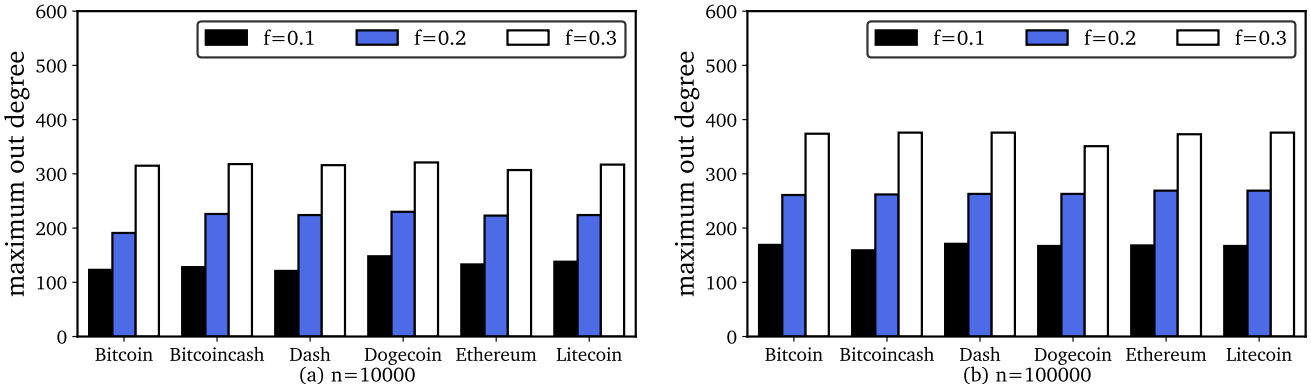

Fig. 4. Maximum node degree in our LOR topology, to achieve $\epsilon = 0.1$ and $\delta = 0.01$. These are under the same real-world stake distributions of 6 major cryptocurrencies as in Figure 1.

Lemma 2. *The total weight of all the honest nodes in all the non-surviving groups is at most $\epsilon/2$, namely, $\sum_{\text{non-surviving } G} W_h(G) \leq \frac{\epsilon}{2}$.*

Proof: If all groups are surviving, the lemma holds trivially. Otherwise consider any non-surviving group G. By definition, we must have $W_h(G) < w_0 W(G)$. Let $W_m(G)$ denote the total weight of malicious nodes in G. Then $W_m(G) = W(G) - W_h(G) > (1 - w_0)W(G_i)$. Define $x = \sum_{\text{non-surviving } G} W_m(G)$, then the previous inequality implies that $x > (1 - w_0) \sum_{\text{non-surviving } G} W(G_i)$, which then gives that $\sum_{\text{non-surviving } G} W(G) < \frac{x}{1-w_0}$.

On the other hand, our construction ensures that (see Section IV-B) $x \leq \sum_G W_m(G) \leq 1.1f$. Thus, we must have:

$$\begin{aligned}\sum_{\text{non-surviving } G} W_h(G) &= \sum_{\text{non-surviving } G} (W(G) - W_m(G)) \\ &= (\sum_{\text{non-surviving } G} W(G)) - x \\ &< (\frac{1}{1-w_0} - 1) \cdot x = \frac{x \cdot \epsilon}{2.2f} \leq \frac{\epsilon}{2}\end{aligned}$$

■

Lemma 3. *Let $l = \lceil \frac{\log(((\delta/2)\cdot(\epsilon/2))/(1+f/10))}{\log(1-w_0)} \rceil = O(1)$. Then except for $\frac{\delta}{2}$ probability, we have $\sum_{G \text{ is SGN}} W_h(G) \leq \frac{\epsilon}{2}$.*

Proof: For every surviving group G, define $X_G \in \{0, 1\}$ as an indicator random variable where $X_G = 1$ iff there is no honest leader in G. By the definition of surviving group, and by how we choose leaders in a group, we have $\Pr[X_G = 1] \leq (1-w_0)^l$. Let Y denote the total weight of all the honest nodes in SGNs. Specifically, $Y = \sum_{\text{surviving } G}(W_h(G) \cdot X_G)$. Hence, $\mathrm{E}[Y] \leq (1-w_0)^l \cdot \sum_{\text{surviving } G} W_h(G) \leq (1-w_0)^l \cdot (1+\frac{f}{10})$. By Markov's inequality, we have $\Pr[Y \geq \frac{\epsilon}{2}] \leq \frac{(1-w_0)^l \cdot (1+\frac{f}{10})}{\epsilon/2} \leq \frac{\delta}{2}$, which proves the lemma. ■

VI. Experimental Results of Our Design

We have fully implemented our LOR design in C++. We use this implementation to quantify the node degree and diameter in our LOR topology.

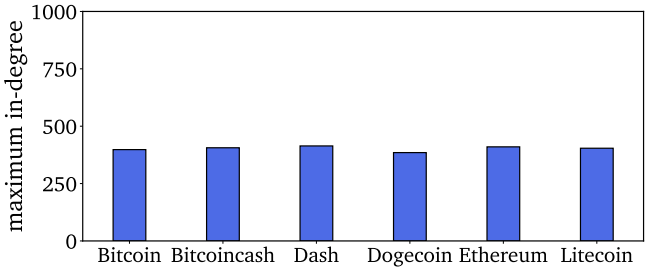

Fig. 5. Maximum in-degree in our LOR topology. The topology is for $n = 100000$, $f = 0.3$, $\epsilon = 0.1$ and $\delta = 0.01$, and under the same stake distributions as Figure 4(b).

Maximum node degree. Figure 4 presents the maximum node degree in our LOR topology, where the topology is configured to achieve the (ϵ, δ) guarantee with $\epsilon = 0.1$ and $\delta = 0.01$. To ensure a fair comparison, these ϵ and δ values are the same as in Figure 1, where we presented the maximum node degrees in existing designs [17], [18]. Figure 4 considers f ranging from 0.1 to 0.3. While blockchain designs theoretically can often tolerate f up to $\frac{1}{3}$ or $\frac{1}{2}$, in actual experiments, researchers usually consider $f = 0.2$ or $f = 0.25$, due to the difficulty of approaching the theoretical limit in practice [4], [6], [7], [10]. Hence we use these more practical f values. (Theoretically, our topology tolerate all constant $f < 1.0$.)

Figure 4 shows that the maximum node degree in our design is around 200 to 400, for f from 0.1 to 0.3. This is in sharp contrast to the results in Figure 1 for existing designs [17], [18], where a node may have a degree of roughly 25000 or larger (for $f = 0.2$). Figure 4 further shows that the maximum node degree does not increase much when n increases from 10000 to 100000. This is expected, since asymptotically, the maximum node degree in our design is $O(\log n)$.

Incoming degree and diameter. We further present experimental results on the maximum incoming degree (Figure 5) and the diameter (Figure 6), for $n = 100000$ and $f = 0.3$. We do not separately plot results for $n = 10000$ and $f = 0.1$ or 0.2, since those results are even better.

Figure 5 shows that the maximum incoming degree in our LOR topology is not far from the maximum outgoing degree

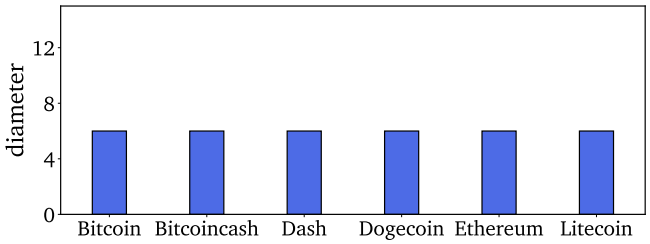

Fig. 6. Diameter of our LOR topology. The topology is for $n = 100000$, $f = 0.3$, $\epsilon = 0.1$ and $\delta = 0.01$, and under the same stake distributions as Figure 4(b).

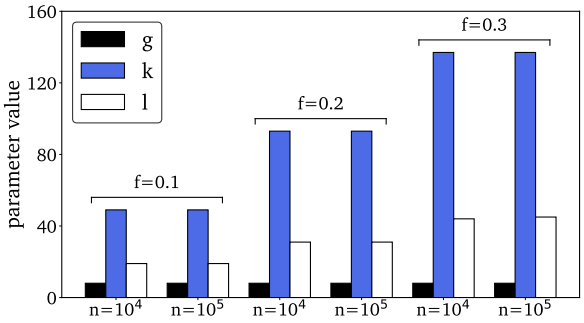

Fig. 7. Parameter values in our LOR topology, under the settings of Figure 4. These parameter values do not depend on the stake distribution.

in Figure 4. Figure 6 shows that the diameter of our LOR topology is small (below 10), even for $n = 100000$.

Parameter values. Finally, to gain deeper understanding into these results, Figure 7 further plots the (g, k, l) parameter values that are sufficient for our LOR topology to achieve the target (ϵ, δ)-guarantee, under the settings of Figure 4.

VII. Discussions

Node anonymity. In both our design and existing designs [17], [18], a node chooses its outgoing neighbors based on the stake distribution. To do so, the neighbors need to reveal their IP addresses. This may lead to privacy concern, since the association between the wallet address (from the stake distribution) and the corresponding node (at the given IP address) is revealed. Fortunately, such association can be hidden by using zero-knowledge proofs [32]. The wallet and the node can use separate public/private key pairs: The public key of the wallet is used for signing transactions, while the public key for the node is for forming the overlay. To enable the nodes to form the overlay, each wallet A delegates its s stake to the corresponding node A' using zero-knowledge proofs such as zk-cred [32], by publishing on the blockchain a commitment to delegate. The commitment will not reveal the identity of A', but A' can create a zero-knowledge proof [32] showing that there exists some commitment delegating at least s stake to it, without revealing the specific commitment it refers to. This then enables the nodes to form the overlay based on their stake.

Generalize to PoW. We has been using PoS blockchains as an example context. It is easy to generalize to PoW. For example, in typical PoW blockchains, the mining process aims to find x, such that $\texttt{hash}(x)$ has a certain number a of leading zeros. To use our design, each party can remember all the x's that it finds where $\texttt{hash}(x)$ has a' ($a' < a$) leading zeros. Intuitively, these are *partial-successes*. We can then view the number of partial-successes found by a node as its "stake", and then apply our design.

VIII. Related Works

Eclipse attacks. It is well-known that existing blockchains are vulnerable to *eclipse attacks* [33]–[36], where the adversary partitions honest nodes from the rest of the overlay network. Heilman et al. [33] exploit the neighbor selection mechanism in Bitcoin to eclipse nodes, while Marcus et al. [35]'s attack is on Ethereum. Apostolaki et al. [34] exploits the specifics of BGP routing in their eclipse attack. Tran et al. [36] describes a stealthier eclipse attack, using the power of AS on Internet routing. All these works [33]–[36] suggest patches to fix the problem. But these ad hoc patches are only designed for specific strategies of the adversary — they do not provide security guarantees under all strategies. In contrast, as long as Internet routing is intact, our design on secure flooding can provably defend against all eclipse attacks, regardless of the strategy of the adversary. Defending against BGP-level or AS-level adversaries (as those in [34], [36]), which can compromise Internet routing, is beyond the scope of this paper.

Researchers have also designed mechanisms [22]–[24] to enables eclipsed nodes to detect such attacks. Xu et al. [22] use random forests to detect eclipse attacks, based on packet meta-data. Zheng et al. [23] monitor block difficulty to detect eclipse attacks. Alangot et al. [24] detect based on excessively long block interval, and by asking web servers to help with state propagation. These mechanisms are complementary to our work. For example, they can be added to our design, to deal with adversaries that can compromise Internet routing [34], [36].

Secure flooding in blockchains. In blockchains, overlay topologies are often constructed in ad hoc fashion. For example, Ethereum's overlay [19], [35] is based on the Kademlia DHT, which is not designed to be robust against an adversary. Unsurprisingly, these ad hoc designs have been discovered to be insecure [33], [35]. Provably secure flooding has not attracted much attention from blockchain researchers, until recently [16]–[18]. Matt et al. [16] focus on showing that secure flooding is possible, against a δ-delayed adaptive adversary. Their key focus is on formally reasoning about δ-delayed corruption in the UC framework. Robust topology construction is not their focus: They simply use an Erdos-Renyi random graph [37], while assuming a certain fraction of the nodes to be honest. Such assumption unfortunately is incompatible with typical PoW/PoS blockchains, where the adversary is *resource-bounded* (see Section I). Matt et al. [16] suggest Liu et al. [17]'s design, as a possible solution under such resource-bounded adversaries. In contrast to [16] and similar to [17], our work targets resource-bounded adversaries.

Liu et al. [17] and Coretti et al. [18] are the most related prior efforts to our work. Both of them consider resource-

bounded adversaries as in typical PoW/PoS blockchains. We have thorough compared with them throughout the paper already.

Finally, Rohrer and Tschorsch [38] propose *Kadcast*, a protocol that adapts the Kademlia DHT to the blockchain setting and makes message flooding more efficient. It is unclear whether the topology in Kadcast [38] is secure against a byzantine adversary. In comparison, our design comes with provable security guarantees.

Secure flooding in byzantine agreement/broadcast and MPC protocols. Secure flooding has been used in byzantine agreement/ broadcast and MPC protocols, to enable better performance. Dwork et al. [39] propose an *almost-everywhere* agreement protocol. They use a low-degree network among nodes, to achieve byzantine agreement despite $O(\frac{n}{\log n})$ node failures. Upfal [40], by employing better graph construction techniques, improves upon [39] and can tolerate $O(n)$ node failures. Ben-Or and Goldriech [41] propose an agreement protocol, using a network of constant degree to tolerate statistical node failures. For byzantine broadcast, Tsimos et al. [42] use secure flooding on an Erdos-Renyi random topology, to reduce communication complexity. In MPC context, Boyle et al. [43] build overlay networks where each node only communicates with $O(\texttt{polylog}(n))$ other nodes. Chandran et al. [44] improves upon [43] by additionally tolerating adaptive corruptions and more malicious nodes. The topology used in [44] is an Erdos-Renyi graph.

Compared to our work, all the above efforts [39]–[44] assume that a certain fraction of the nodes are honest. In contrast, we focus on secure flooding in the blockchain context, where the adversary is *resource-bounded*. As explained earlier, this makes the problem fundamentally different, and these prior designs cannot be applied to our setting.

Network proximity. Kermarrec et al. [45] develop overlay topology that takes network proximity into account, so that a node is more likely to choose neighbors that are closer in the underlying Internet. To achieve this, their nodes form a hierarchical structure related to the underlying network structure. They consider a model where a certain fraction of the nodes can fail. In contrast, our design i) is not concerned with network proximity, and ii) considers resource-bounded adversaries.

IX. Conclusions

This paper has proposed a novel LOR design for robust overlay and secure flooding, against resource-bounded adversaries in typical permissionless blockchains. Our design is *the very first* such design with practically-feasible node degrees. Our design offers provable security guarantees against all possible adversarial strategies. To achieve it goals, our design avoids using virtual nodes, and exploits properties of bounded-disparity groups.

References

[1] S. Nakamoto, "Bitcoin whitepaper," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] P. Daian, R. Pass, and E. Shi, "Snow white: Robustly reconfigurable consensus and applications to provably secure proof of stake," in *FC*, 2019.

[3] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *CRYPTO*, 2017.

[4] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *SOSP*, 2017.

[5] H. Yu, I. Nikolić, R. Hou, and P. Saxena, "Ohie: Blockchain scaling made simple," in *Oakland*, 2020.

[6] V. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, "Prism: Deconstructing the blockchain to approach physical limits," in *CCS*, 2019.

[7] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *CCS*, 2016.

[8] P. Szalachowski, D. Reijsbergen, I. Homoliak, and S. Sun, "Strongchain: Transparent and collaborative proof-of-work consensus," in *USENIX Security*, 2019.

[9] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *CCS*, 2018.

[10] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in *Oakland*, 2018.

[11] C. Li, P. Li, D. Zhou, Z. Yang, M. Wu, G. Yang, W. Xu, F. Long, and A. C.-C. Yao, "A decentralized blockchain with high throughput and fast confirmation," in *USENIX Annual Technical Conference*, 2020.

[12] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *NSDI*, 2016.

[13] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *USENIX Security*, 2016.

[14] R. Hou, H. Yu, and P. Saxena, "Using throughput-centric byzantine broadcast to tolerate malicious majority in blockchains," in *Oakland*, 2022.

[15] R. Hou and H. Yu, "Optimistic fast confirmation while tolerating malicious majority in blockchains," in *Oakland*, 2023.

[16] C. Matt, J. B. Nielsen, and S. E. Thomsen, "Formalizing delayed adaptive corruptions and the security of flooding networks," in *CRYPTO*, 2022.

[17] C.-D. Liu-Zhang, C. Matt, U. Maurer, G. Rito, and S. E. Thomsen, "Practical provably secure flooding for blockchains," in *Asiacrypt*, 2022.

[18] S. Coretti, A. Kiayias, C. Moore, and A. Russell, "The generals' scuttlebutt: Byzantine-resilient gossip protocols," in *CCS*, 2022.

[19] T. Wang, C. Zhao, Q. Yang, S. Zhang, and S. C. Liew, "Ethna: Analyzing the underlying peer-to-peer network of ethereum blockchain," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, 2021.

[20] Source code for experiments in this paper: https://www.comp.nus.edu.sg/~sunyuch/projects/prdc24/prdc24.html.

[21] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*, 2002.

[22] G. Xu, B. Guo, C. Su, X. Zheng, K. Liang, D. S. Wong, and H. Wang, "Am i eclipsed? a smart detector of eclipse attacks for ethereum," *Computers & Security*, vol. 88, p. 101604, 2020.

[23] H. Zheng, T. Tran, and O. Arden, "Total eclipse of the enclave: detecting eclipse attacks from inside tees," in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2021.

[24] B. Alangot, D. Reijsbergen, S. Venugopalan, P. Szalachowski, and K. S. Yeo, "Decentralized and lightweight approach to detect eclipse attacks on proof of work blockchains," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1659–1672, 2021.

[25] Etherscan, Accessed: 2024-03-11. [Online]. Available: https://etherscan.io/accounts

[26] BitInfoCharts, Accessed: 2024-03-11. [Online]. Available: https://bitinfocharts.com

[27] J. A. Donet Donet, C. Pérez-Sola, and J. Herrera-Joancomartí, "The bitcoin p2p network," in *FC Workshops*, 2014.

[28] A. R. Sai, J. Buckley, and A. Le Gear, "Characterizing wealth inequality in cryptocurrencies," *Frontiers in blockchain*, vol. 4, 2021.

[29] S. Coretti, A. Kiayias, C. Moore, and A. Russell, "The generals' scuttlebutt: Byzantine-resilient gossip protocols," *Cryptology ePrint Archive, Paper 2022/541*, 2022. [Online]. Available: https://eprint.iacr.org/2022/541

[30] R. Pass and E. Shi, "Hybrid consensus: Efficient consensus in the permissionless model," in *DISC*, 2017.

[31] R. J. Serfling, "Probability inequalities for the sum in sampling without replacement," *The Annals of Statistics*, pp. 39–48, 1974.
[32] M. Rosenberg, J. White, C. Garman, and I. Miers, "zk-creds: Flexible anonymous credentials from zksnarks and existing identity infrastructure," in *Oakland*, 2023.
[33] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *USENIX security*, 2015.
[34] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *Oakland*, 2017.
[35] Y. Marcus, E. Heilman, and S. Goldberg, "Low-resource eclipse attacks on ethereum's peer-to-peer network," *Cryptology ePrint Archive*, 2018.
[36] M. Tran, I. Choi, G. J. Moon, A. V. Vu, and M. S. Kang, "A stealthier partitioning attack against bitcoin peer-to-peer network," in *Oakland*, 2020.
[37] B. Bollobás, "Random graphs," *Cambridge University Press, UK*, 2001.
[38] E. Rohrer and F. Tschorsch, "Kadcast: A structured approach to broadcast in blockchain networks," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, 2019.
[39] C. Dwork, D. Peleg, N. Pippenger, and E. Upfal, "Fault tolerance in networks of bounded degree," in *STOC*, 1986.
[40] E. Upfal, "Tolerating linear number of faults in networks of bounded degree," in *PODC*, 1992.
[41] M. Ben-Or and D. Goldriech, "Agreement in the presence of faults on networks of constant degree," 1992.
[42] G. Tsimos, J. Loss, and C. Papamanthou, "Gossiping for communication-efficient broadcast," in *CRYPTO*, 2022.
[43] E. Boyle, S. Goldwasser, and S. Tessaro, "Communication locality in secure multi-party computation: how to run sublinear algorithms in a distributed setting," in *TCC*, 2013.
[44] N. Chandran, W. Chongchitmate, J. A. Garay, S. Goldwasser, R. Ostrovsky, and V. Zikas, "The hidden graph model: Communication locality and optimal resiliency with adaptive faults," in *Conference on Innovations in Theoretical Computer Science*, 2015.
[45] A.-M. Kermarrec, L. Massoulié, and A. J. Ganesh, "Probabilistic reliable dissemination in large-scale systems," *IEEE Transactions on Parallel and Distributed systems*, vol. 14, no. 3, pp. 248–258, 2003.

APPENDIX A
LEMMA 7 AND ITS PROOF

Lemma 4 through 6 in the following are existing results, which we will invoke later to prove Lemma 7. As we construct a random intra-group topology for every group G, we overload the notation G, and use G to denote both the group and the intra-group topology of the group. Define $ER_{N,p}$ to be the undirected Erdos-Renyi random graph with N nodes, where each pair of nodes have an edge between them with probability p independently. Define $\vec{\mathrm{ER}}_{N,p}$ to be the directed version of that, where a directed edge from one node to another exists with probability p independently. Define $Diam(G)$ as the diameter of the graph G.

Lemma 4. (Adapted from Corollary 1.1 in [31].) *Consider sampling k balls uniformly at random without replacement from N balls (some of them are red). Let X denote the number of red balls sampled, then for all $t > 0$ and $t \in \mathbb{R}$,*

$$\Pr[X - E[X] \geq k \cdot t] \leq e^{-2kt^2}$$

Lemma 5. (Adapted from Corollary 5 in [17].) *For all positive integer N and $p \in [0,1]$, where $\frac{7}{N} \leq p \leq 1$, we have:*

$$\Pr_{G \leftarrow ER_{N,p}}[Diam(G) > 7\log(\frac{1}{2p}) + 2] \leq N \cdot \left(e^{-\frac{Np}{18}} + (6\log(\frac{1}{2p}) + 1) \cdot e^{-\frac{7Np}{108}}\right) + e^{-N(\frac{Np}{9} - 2)}$$

Lemma 6. (Adapted from Lemma 7 in [17].) *For all positive integer N and t, and $0 \leq p \leq 1$, let u be any node in the vertex set of the random graph distribution $\vec{ER}_{N,p}$, then:*

$$\Pr_{G_1 \leftarrow ER_{N,p}}[Diam(G_1) \leq t] \quad \leq \quad \Pr_{G_2 \leftarrow \vec{ER}_{N,p}}[Ecce(G_2, u) \leq t],$$

where $Ecce(G_2, u)$ is the eccentricity of u in G_2 (i.e., the maximum distance from u to any other node in $V(G_2)$).

Recall that for any given group G, $W(G)$ is the total weight of nodes in G, and $W_h(G)$ is the total weight of the honest nodes in G. Also recall that G is a *surviving group* if $\frac{W_h(G)}{W(G)} \geq w_0$, where $w_0 = \frac{\epsilon}{2.2f+\epsilon}$. Otherwise G is a *non-surviving* group. If a surviving group contains some honest leader, it is a *surviving group with surviving leader* (or *SGL*). Otherwise it is a *surviving group with no surviving leader* (or *SGN*).

Lemma 7. *Let $h_0 = \frac{w_0}{g - w_0 g + w_0}$, and $k = \frac{400 \cdot \log n + 27}{h_0^2 \cdot \delta} = O(\log n)$. Then except for $\delta/2$ probability, in all surviving groups, the honest nodes form a strongly connected component with a diameter $O(\log n)$.*

Proof: The lemma trivially holds if there is no surviving group. Hence we assume the existence of some surviving group. We ignore floor and ceiling for presentation clarity, and our proof easily extends to the general case. For a group G, use $H(G)$ to denote the induced subgraph formed by its honest nodes, after certain random graph is formed. By definition, in a surviving group, honest nodes constitute at least w_0 fraction of weight. Additionally, the weight of nodes in each group differs by at most a factor of g. Therefore, $\frac{|H(G)|}{|G|} \geq \frac{W_h(G)}{g(W(G) - W_h(G)) + W_h(G)} \geq \frac{1}{g(\frac{1}{w_0} - 1) + 1} = \frac{w_0}{g - w_0 g + w_0} = h_0$. In another word, h_0 is a lower bound on the fraction of honest nodes in any surviving group.

Define $\delta' = \frac{\delta}{2 \cdot z}$, where $z = \lfloor \log_g((1 + \frac{f}{10n})/(\frac{f}{10n})) \rfloor + 1$ is the total number of groups. We will show for each surviving group G, except with probability δ', $H(G)$ is a strongly connected component with a diameter $O(\log n)$. The lemma then trivially follows from a union bound across all surviving groups.

For any surviving group G with at most $k+1$ nodes, trivially, $H(G)$ is a clique and we are done. Hence we only need to consider surviving group G where $n \geq |G| > k + 1$. Observe that we always have $h_0 < 1$ and $\delta < 1$. Hence, by the definition of k, we have:

$$h_0(k+1) > 6 \quad (1)$$

$$n > k > 400 \quad (2)$$

Relating $H(G)$ to directed ER random graph. Consider a random directed graph G_1, where $|G_1| = |H(G)|$, formed in the following way: Each node in G_1 first samples a random integer $X \in [0, k]$ where $\Pr[X = x] = \frac{C(|H(G)|-1, x) \cdot C(|G| - |H(G)|, k - x)}{C(|G|-1, k)}$ and $C(\cdot, \cdot)$ is the combination function. Then, the node uniformly randomly sample X other $|G_1| - 1$ nodes without replacement, and form outgoing edges to them.

We claim that G_1 has exactly the same distribution as $H(G)$. Recall in our construction, each node in $H(G)$ samples without replacement k outgoing edges to the other $|G|-1$ nodes. Among those $|G|-1$ nodes, $|H(G)|-1$ nodes are in $H(G)$. Hence, for each node A in $H(G)$, the probability that it has exactly x outgoing edges in the subgraph $H(G)$ is exactly $\frac{C(|H(G)|-1,x)\cdot C(|G|-|H(G)|,k-x)}{C(|G|-1,k)}$. Moreover, all the cases where A forms exactly x outgoing edges in $H(G)$ are symmetric up to permutation of the other $H(G)-1$ honest nodes, and hence $H(G)$ has the same distribution as G_1.

Consider another random directed graph G_2, where $|G_2| = |H(G)|$, formed in the following way: For each node A in G_2, A samples X from a binomial distribution $Bi(|H(G)|-1, \frac{k}{5|G|})$. Then, A uniformly randomly X random nodes without replacement from the $|G_2|$ nodes and form outgoing edges to them. With a similar reasoning as above, it can be shown that G_2 has the same distribution as an ER random graph $\vec{\mathrm{ER}}_{|H(G)|,\frac{k}{5|G|}}$.

Next, we do a simple coupling of G_1 and G_2: As they have the same number of nodes, we compare the nodes with the same index. For any node A, when A establishes outgoing edges to random nodes without replacement in G_1 and G_2, we let A use the same randomness. Hence, either the set of outgoing edges of A in G_1 is a subset of the set of outgoing edges of A in G_2, or vice versa. Now, in the next we show that with probability at least $1-\frac{\delta'}{5}$, for every node, the edges in G_2 is always a subset of the edges in G_1. Define random event E_1: the minimum number of out-neighbors of any node in G_1 is at least $\frac{k\cdot|H(G)|}{2|G|}$. Also define random event E_2: the maximum number of out-neighbors of any node in G_2 is at most $\frac{k\cdot|H(G)|}{2|G|}$. It is clear that when both E_1 and E_2 holds, our claim must hold.

- We first reason about E_1. Fix some node in G_1. Consider the construction of G_1, and observe that the random number X can be interpreted as the number of red balls obtained by sampling k balls from $|G|-1$ balls without replacement, where $|H(G)|-1$ balls are red. Let $\overline{X}$ denote the number of non-red balls obtained in this process. Then, we have that $E[\overline{X}] = \frac{k(|G|-|H(G)|)}{|G|-1}$. Applying Lemma 4, we have:

$$
\begin{aligned}
&\Pr[X < \frac{k|H(G)|}{2|G|}] = \Pr[\overline{X} > k - \frac{k|H(G)|}{2|G|}] \\
&= \Pr\left[\overline{X} - E[\overline{X}] > k - \frac{k|H(G)|}{2|G|} - \frac{k(|G|-|H(G)|)}{|G|-1}\right] \\
&\le \Pr\left[\overline{X} - E[\overline{X}] > k - \frac{k|H(G)|}{2(|G|-1)} - \frac{k(|G|-|H(G)|)}{|G|-1}\right] \\
&= \Pr\left[\overline{X} - E[\overline{X}] > k\cdot\frac{|H(G)|-2}{2(|G|-1)}\right] \\
&\le \exp(-2k(\frac{|H(G)|-2}{2(|G|-1)})^2) \le \exp(-2k(\frac{h_0|G|-2}{2|G|})^2) \\
&\le \exp(-2k(\frac{h_0|G|}{3|G|})^2) = \exp(-2k\frac{h_0^2}{9}) \\
&\le \exp(-\frac{800\log n}{9\delta} - 3) \\
&\le \frac{1}{10}\cdot e^{-80\cdot\log n/\delta} = \frac{1}{10}(\frac{1}{n^{80}})^{\delta^{-1}} \\
&\le \frac{1}{10}\cdot\frac{\delta}{n^{80}} \le \frac{\delta'}{10n}
\end{aligned}
$$

  In the above deduction we have used Equation 1, which implies $h_0|G| \ge h_0(k+1) > 6$. A union bound over all nodes in G_1 will give $\Pr[E_1] \ge 1-\frac{\delta'}{10}$.
- For event E_2. Similarly, we fix a node in G_2 and consider the random number X it draws. We have that X follows a binomial distribution $Bi(|H(G)|-1, \frac{k}{5|G|})$. Hence, $E[X] = \frac{k(|H(G)|-1)}{5|G|} \ge k\frac{h_0|G|-1}{5|G|} \ge \frac{k}{6}$. The last step is because $h_0|G| \ge h_0 k \ge 6$. By Chernoff bound, $\Pr[X \ge \frac{k\cdot|H(G)|}{2}] \le \Pr[X \ge 2.5\cdot E[X]] \le \exp(-\frac{1.5^2}{3.5}\cdot\frac{k}{6}) \le \exp(-40\cdot\log n/\delta) = (\frac{1}{n^{40}})^{\delta^{-1}} \le \frac{\delta}{n^{40}} \le \frac{\delta'}{10n}$. A union bound over all nodes in G_2 gives $\Pr[E_2] \ge 1-\frac{\delta'}{10}$.

Now conditioned on E_1 and E_2, for any positive integer t, we have:

$$\Pr[Diam(G_1) > t] \le \Pr[Diam(G_2) > t] \tag{3}$$

Relating directed ER graphs to undirected ones. We have related $H(G)$ to a directed ER graph in terms of diameter. We next relate the directed ER graph with an undirected ER graph. Let $G_3 \leftarrow \mathrm{ER}_{|H(G)|,\frac{k}{5|G|}}$. Then by Lemma 6, for all positive integer t and for all $u \in G_2$, we have $\Pr[Ecce(G_2,u) > t] \le \Pr[Diam(G_3) > t]$. Then for all positive integer t:

$$
\begin{aligned}
\Pr[Diam(G_2) > t] &\le n\cdot\Pr[Ecce(G_2,u) > t] \\
&\le n\cdot\Pr[Diam(G_3) > t] \qquad (4)
\end{aligned}
$$

Here the first inequality is due to a union bound over all u in G_2 (there at most $|G_2| \le n$ of such u's).

By Equation 3 and Equation 4, to prove that $H(G)$ has $O(\log n)$ diameter (and hence is strongly connected) except for probability δ', it suffices to show $n\cdot\Pr[Diam(G_3) > c\cdot\log n] \le \frac{4\cdot\delta'}{5}$ (for some constant c). Then taking a union bound over this $\frac{4\cdot\delta'}{5}$ probability and the probability that either E_1 or E_2 does not happen will complete the proof.

Set $N = |H(G)|$ and $p = \frac{k}{5|G|}$. Notice $1 \ge p = \frac{k}{5|G|} \ge \frac{35/h_0}{5|G|} \ge \frac{7}{|H(G)|} = \frac{7}{N}$. Plugging such N and p into Lemma 5 eventually shows $\Pr[Diam(G_3) > 7\log(\frac{1}{2p})+2] \le n\cdot\left((\frac{1}{n^4})^{\delta^{-1}} + n\cdot(\frac{1}{n^5})^{\delta^{-1}}\right) + (\frac{1}{n^{100}})^{\delta^{-1}} \le \frac{3\delta}{n^3}$. Observe that $7\log(\frac{1}{2p})+2 \le 7\log(2.5|G|)+2 \le c\log n$ for some constant c. Hence we have $n\cdot\Pr[Diam(G_3) > c\cdot\log n] \le \frac{3\delta}{n^2} = \frac{6\cdot\delta'\cdot z}{n^2} \le \frac{4\delta'}{5}$. The deduction here uses Equation 2 and the fact that $z \le n$. ∎

APPENDIX B
FINDING SUITABLE PARAMETERS

We use an explicit algorithm to search for suitable values for the parameters (g,k,l), in a way similar to constraint optimization. Specifically, we use a *parameter testing algorithm* to do so. Recall that these parameters (g,k,l) only depend

on (n, f, ϵ, δ), and not on the stake distribution. For given (n, f, ϵ, δ) and given (g, k, l), our parameter testing algorithm will determine whether such (g, k, l) are *sufficient* to achieve the desired (ϵ, δ)-guarantee, under all possible adversarial strategies. Among those (g, k, l) tuples that are sufficient, we then pick a (g, k, l) tuple that leads to good (i.e., small) maximum node degree (as from Section IV-D) in our topology.

There is no need to exhaustively test all possible (g, k, l) tuples — testing more just enables us to potentially find better parameters that result in smaller node degree. The security of the topology is always guaranteed, once the parameter testing algorithm determines (g, k, l) to be sufficient. Appendix B.5 later will explain what tuples were tested for our experiments. Also note that such testing does not need to be run *online*. We can simply run the parameter testing algorithm offline, for various (n, f, ϵ, δ) values. The (g, k, l) values found can then be *hard-coded* into our topology generation process.

B.1 Some Basic Concepts and Properties

Before describing the parameter testing algorithm, we formalize a few concepts, and obtain a few basic properties regarding the adversary's strategies.

Categorizing strategies into types. We call the adversary's overall strategy as a *global strategy*. A global strategy consists of z *local strategies*, where the i-th local strategy describes the adversary's strategy on the i-th group ($1 \leq i \leq z$), namely, which nodes are corrupted in that group. We categorize local strategies into different *types*.

Definition 8. *A local strategy is of* $(m_{\text{malicious}}, m_{\text{honest}})$-type*, if under that local strategy, the corresponding group has exactly* $m_{\text{malicious}}$ *malicious nodes and* m_{honest} *honest nodes. Note that it is possible for two different local strategies on two different groups to belong to the same type.*

Localized damage. Consider the subgraph containing all the honest nodes in the i-th group ($1 \leq i \leq z$), and all the edges among them. Define the set $T_i = \{A \mid (A \text{ is a node in this subgraph}) \ \& \ (A \text{ is reachable to and from some leader in this subgraph})\}$. If the subgraph contains no leader, then T_i is defined to be $\emptyset$. Recall that all the leaders from all the groups form a clique. This implies that the union $\cup_{i=1}^{z} T_i$ must be a strongly-connected component. Hence, an honest node in the i-th group is eclipsed, iff it is not in T_i.

Now if the adversary corrupts more nodes in the i-th group, it can only affect T_i, and not T_j where $j \neq i$. This leads to *localized damage*: If the adversary corrupts nodes in a group, then nodes in other groups will never become eclipsed due to such corruption.

Cost-benefit ratio. Let f_i be the total weight corrupted in group i by the i-th local strategy, where $\sum_{i=1}^{z} f_i \leq 1.1f$. Here $1.1f$ is the total budget of the adversary in terms of weight (see Section IV-B). Under fixed f_i's, the adversary still has the freedom to choose which nodes to corrupt.

Algorithm 1 `Parameter_Testing`$(n, f, \epsilon, \delta, g, k, l)$.

1: $r_{\text{max}} \leftarrow \frac{\epsilon}{1.1f}$; $z \leftarrow \lfloor \log_g((1 + \frac{f}{10n})/(\frac{f}{10n})) \rfloor + 1$;
2: `sufficient` $\leftarrow$ `true`;
3: **foreach** $(m_{\text{malicious}}, m_{\text{honest}})$-type where $m_{\text{malicious}} + m_{\text{honest}} \leq n$ **do**
4: $w \leftarrow$ `GccSize`$(k, \frac{\delta}{2z}, m_{\text{malicious}}, m_{\text{honest}})$;
5: $\psi \leftarrow \frac{1}{(1+\frac{1}{r_{\text{max}}})(1+1/(r_{\text{max}} \cdot \frac{m_{\text{malicious}}}{g(m_{\text{honest}}-w)}-1))}$;
6: **if** $(\frac{g(m_{\text{honest}}-w)}{m_{\text{malicious}}} < r_{\text{max}})$ **then** `part1` $\leftarrow$ `true`; **else** `part1` $\leftarrow$ `false`; **endif**
7: **if** $(m_{\text{malicious}} + m_{\text{honest}} \leq l)$ or $((1-\psi)^l \leq \frac{\delta}{2z})$ **then** `part2` $\leftarrow$ `true`; **else** `part2` $\leftarrow$ `false`; **endif**
8: **if** (!`part1`) or (!`part2`) **then** `sufficient` $\leftarrow$ `false`; **endif**
9: **endforeach**
10: **return** `sufficient`;

Now consider the i-th group, and define S_{eclipsed} to be the total weight held by the honest nodes not in T_i. (Recall that these nodes are eclipsed.) We view S_{eclipsed} as the *benefit* achieved by the local strategy, and f_i as the local strategy's *cost*. Note that larger f_i does not always lead to larger S_{eclipsed}: When f_i reaches the total stake of the group, S_{eclipsed} will be 0, since there are no honest nodes left. Define the *benefit-cost ratio* (or *BCR*) of the local strategy to be S_{eclipsed}/f_i. Given localized damage, it is obvious that the BCR of the i-th local strategy depends only on that local strategy itself, and not on the local strategies for other groups. We can now prove the following simple property:

Lemma 9. *To achieve the* (ϵ, δ)*-guarantee, it suffices to ensure that for each of the* z *groups, regardless of the local strategy used, the BCR is at most* $r_{\text{max}} = \frac{\epsilon}{1.1f}$*, except for* $\frac{\delta}{z}$ *probability.*

Proof: First, a union bound tells us that except for probability δ, all the z local strategies get BCR of at most r_{max}. We condition upon such event. The total combined benefit across the z local strategies will be at most $r_{\text{max}} \cdot \sum_{i=1}^{z} f_i \leq r_{\text{max}} \times 1.1f = \epsilon$. This means that at most ϵ honest weight is eclipsed. In turn, at most ϵ honest stake is eclipsed. □

B.2 Pseudo-code of Parameter Testing Algorithm

Based on Lemma 9, the parameter testing algorithm exhaustively checks the BCR of all the $O(n^2)$ types of possible local strategies. For each type, it checks whether BCR $\leq r_{\text{max}}$ *always* holds (except for $\frac{\delta}{z}$ probability). Here, "always" refer to i) all possible stake distribution in the group, ii) all possible local strategies of the given type, and iii) allowing the local strategy to use all possible $f_i \in [0, 1.1f]$.

Algorithm 1 gives the complete pseudo-code of the parameter testing algorithm. Here Line 3 through Line 9 exhaustively check the $O(n^2)$ types. For each type, the algorithm determines whether the given parameters (g, k, l) suffice to ensure that BCR $\leq r_{\text{max}}$. This is done, *indirectly*, by the conditions at Line 6 and Line 7. If the given (g, k, l) suffices for all types, then the parameter testing algorithm declares the tuple

as sufficient. Line 4 in the algorithm needs to compute a function $\texttt{GccSize}()$. We explain that function later.

B.3 Correctness Proof

This section formally proves the correctness of the parameter testing algorithm. Ultimately, we will prove that if the parameter testing algorithm determines (g, k, l) as being sufficient, then the topology constructed using such (g, k, l) must achieve the (ϵ, δ)-guarantee.

Some notations. Consider a given $(m_{\text{malicious}}, m_{\text{honest}})$-type of local strategies. The group has total $m_{\text{malicious}} + m_{\text{honest}}$ nodes. Consider the intra-group topology. Let m_{gcc} be the number of nodes in the largest strongly-connected component (i.e., *Giant Connected Component* or *GCC*), among the m_{honest} honest nodes. Define $m_{\text{nongcc}} = m_{\text{honest}} - m_{\text{gcc}}$. Define $W_{\text{malicious}}$, W_{honest}, W_{gcc}, and W_{nongcc} to be the total weight held by these $m_{\text{malicious}}$, m_{honest}, m_{gcc}, and m_{nongcc} nodes, respectively. Define the function $\texttt{GccSize}(k, \frac{\delta}{2z}, m_{\text{malicious}}, m_{\text{honest}})$ to return the maximum v where $\Pr[m_{\text{gcc}} \geq v] \geq 1 - \frac{\delta}{2z}$. Let $w = \texttt{GccSize}(k, \frac{\delta}{2z}, m_{\text{malicious}}, m_{\text{honest}})$. Intuitively, this means that except for probability $\frac{\delta}{2z}$, the GCC has a size of at least w.

Theorem 10. *For any given $n \geq 1$, $f \in [0, 1)$, $\epsilon \in (0, 1]$, $\delta \in (0, 1]$, let (g, k, l) be any parameter values that the parameter testing algorithm determines to be sufficient. Then our* LOR *topology constructed with such (g, k, l) must achieve the (ϵ, δ)-guarantee, under all possible adversarial strategies and all possible stake distributions (including adversarially generated ones).*

Proof: By Lemma 9, it suffices to prove that for any given group, all possible local strategies for the group must have BCR $\leq r_{\text{max}}$, except for $\frac{\delta}{z}$ probability. Recall the definitions of $m_{\text{malicious}}$, m_{honest}, m_{gcc}, m_{nongcc}, $W_{\text{malicious}}$, W_{honest}, W_{gcc}, W_{nongcc}, and w. If $m_{\text{malicious}} + m_{\text{honest}} \leq l$, then every node in the group is a leader and we must have BCR $= 0 \leq r_{\text{max}}$.

If $m_{\text{malicious}} + m_{\text{honest}} > l$, then by Line 6 and Line 7 in Algorithm 1, we must have $\frac{g(m_{\text{honest}} - w)}{m_{\text{malicious}}} < r_{\text{max}}$ and $(1-\psi)^l \leq \frac{\delta}{2z}$. Next we consider two cases:

- $\frac{W_{\text{gcc}} + W_{\text{nongcc}}}{W_{\text{malicious}}} < r_{\text{max}}$. Note that the benefit achieved by the local strategy is at most $W_{\text{gcc}} + W_{\text{nongcc}}$. Hence the BCR is at most $\frac{W_{\text{gcc}} + W_{\text{nongcc}}}{W_{\text{malicious}}}$, which is less than r_{max}.
- $\frac{W_{\text{gcc}} + W_{\text{nongcc}}}{W_{\text{malicious}}} \geq r_{\text{max}}$. We condition our reasoning on the event $m_{\text{gcc}} \geq w$. By definition of w, this event happens with at least $1 - \frac{\delta}{2z}$ probability. Our topology chooses l leaders from each group. Consider the very first leader. The probability of this leader being in the GCC is $p = \frac{W_{\text{gcc}}}{W_{\text{gcc}} + W_{\text{nongcc}} + W_{\text{malicious}}}$. If this first leader is not in the GCC, then the probability of the second leader being in the GCC is at least p. Such an argument applies to all l leaders. Lemma 11 later proves that $p \geq \psi$. Hence except for $(1-\psi)^l$ probability, there must exist a leader among the m_{gcc} nodes.
  With a union bound and since $(1-\psi)^l \leq \frac{\delta}{2z}$ by Line 7, we know that with probability except $1 - \frac{\delta}{z}$, there exists a leader among the m_{gcc} nodes and $m_{\text{gcc}} \geq w$. Since there exists a leader among the m_{gcc} nodes, only W_{nongcc} can count toward the benefit to the adversary. This means that the BCR is at most $\frac{W_{\text{nongcc}}}{W_{\text{malicious}}}$. We further have $\frac{W_{\text{nongcc}}}{W_{\text{malicious}}} \leq \frac{g \cdot m_{\text{nongcc}}}{m_{\text{malicious}}} \leq \frac{g(m_{\text{honest}} - w)}{m_{\text{malicious}}} < r_{\text{max}}$. Here the last inequality follows from Line 6. Hence we must have BCR $\leq r_{\text{max}}$.

□

Lemma 11. *If $m_{\text{gcc}} \geq w$, $\frac{W_{\text{gcc}} + W_{\text{nongcc}}}{W_{\text{malicious}}} \geq r_{\text{max}}$, $\frac{g(m_{\text{honest}} - w)}{m_{\text{malicious}}} < r_{\text{max}}$, and $\frac{W_{\text{nongcc}}}{W_{\text{malicious}}} < r_{\text{max}}$, then:*

$$\frac{W_{\text{gcc}}}{W_{\text{gcc}} + W_{\text{nongcc}} + W_{\text{malicious}}} \quad \geq \quad \psi, \tag{5}$$

where

$$\psi \quad = \quad \frac{1}{(1 + \frac{1}{r_{\text{max}}})(1 + 1/(r_{\text{max}} \cdot \frac{m_{\text{malicious}}}{g(m_{\text{honest}} - w)} - 1))} \tag{6}$$

Proof: Since the weight of the nodes in the same group can differ by at most a factor of g, we immediately have $\frac{W_{\text{malicious}}}{W_{\text{nongcc}}} \geq \frac{m_{\text{malicious}}}{g(m_{\text{honest}} - m_{\text{gcc}})} \geq \frac{m_{\text{malicious}}}{g(m_{\text{honest}} - w)}$. Together with $\frac{W_{\text{gcc}} + W_{\text{nongcc}}}{W_{\text{malicious}}} \geq r_{\text{max}}$, $\frac{g(m_{\text{honest}} - w)}{m_{\text{malicious}}} < r_{\text{max}}$, and $\frac{W_{\text{nongcc}}}{W_{\text{malicious}}} < r_{\text{max}}$, we have:

$$\begin{aligned}
& \frac{W_{\text{gcc}}}{W_{\text{gcc}} + W_{\text{nongcc}} + W_{\text{malicious}}} \\
= \quad & \frac{1}{1 + \frac{W_{\text{malicious}} + W_{\text{nongcc}}}{W_{\text{gcc}}}} \geq \frac{1}{1 + \frac{W_{\text{malicious}} + W_{\text{nongcc}}}{r_{\text{max}} \cdot W_{\text{malicious}} - W_{\text{nongcc}}}} \\
= \quad & \frac{1}{1 + \frac{1}{r_{\text{max}}} + (1 + \frac{1}{r_{\text{max}}}) \frac{1}{r_{\text{max}} \cdot \frac{W_{\text{malicious}}}{W_{\text{nongcc}}} - 1}} \\
\geq \quad & \frac{1}{1 + \frac{1}{r_{\text{max}}} + (1 + \frac{1}{r_{\text{max}}}) \frac{1}{r_{\text{max}} \cdot \frac{m_{\text{malicious}}}{g(m_{\text{honest}} - w)} - 1}} = \psi
\end{aligned}$$

□

B.4 Computing $\texttt{GccSize}()$

We now explain how to compute $\texttt{GccSize}(k, \frac{\delta}{2z}, m_{\text{malicious}}, m_{\text{honest}})$. Intuitively, this function returns the maximum v such that the GCC size is at least v except for $\frac{\delta}{2z}$ probability. The key observation is that once $(k, \frac{\delta}{2z}, m_{\text{malicious}}, m_{\text{honest}})$ are fixed, the return value of $\texttt{GccSize}()$ is also fixed: This return value no longer depends on the stake distribution, or the local strategy, or which $m_{\text{malicious}}$ nodes the adversary chooses to corrupt. This is simply because in the intro-group topology, *all nodes are symmetric*.

With this observation, we can simply determine $\texttt{GccSize}()$ via numerical methods, such as Monte-Carlo sampling, and the results are guaranteed to be correct under all adversarial strategies. Specifically, in our Monte-Carlo sampling, we construct an intra-group topology with vertices 1 through $m_{\text{malicious}} + m_{\text{honest}}$. We then remove vertices 1 through $m_{\text{malicious}}$, and calculate the GCC size (or a lower bound of that) among the remaining nodes. Such Monte-Carlo sampling can be done *offline*, with the results being *hard-coded* into the parameter testing algorithm.

B.5 Parameter Values Tested for Experiments

Recall that there is no need to exhaustively test all possible (g, k, l) tuples — testing more just enables us to potentially find better parameters that result in smaller node degree. The security of our topology is always guaranteed, once the parameter testing algorithm determines (g, k, l) to be sufficient. For our experimental results in Figure 7: (i) We have tested $g = 2, 4, 6, 8, \ldots$, until the g value under which the total number of groups becomes 1. (ii) For each g, we have tested $k = x, 1.5x, 2x, 2.5x, \ldots$, until k reaches n. To save overhead, we do not test k values smaller than x. Here x is such that the expected number of outgoing edges from each honest node to other honest nodes in the same group is 1, when the adversary corrupts f' fraction of the nodes in that group, where $\frac{g(1-f')}{f'} = r_{\text{max}}$. Note that if the adversary corrupts more that f' fraction, then its BCR must be below r_{max}. We do not test k values smaller than x, because those smaller k values will likely not suffice, even under this example adversarial strategy. (iii) For each (g, k) tuple, we find the smallest possible l value, via a binary search, such that the parameter testing algorithm determines (g, k, l) as sufficient.