Cryptoeconomic Security for Data Availability Committees

Ertem Nusret Tas and Dan Boneh

Stanford University

Abstract. Layer 2 systems have received increasing attention due to their potential to scale the throughput of L1 blockchains. To avoid the cost of putting data on chain, these systems increasingly turn to off-chain data availability solutions such as data availability committees (DACs). However, placing trust on DACs conflicts with the goal of obtaining an L2 architecture whose security relies solely on the L1 chain. To eliminate such trust assumptions, we propose a DAC protocol that provides financial incentives to deter the DAC nodes from adversarial behavior such as withholding data upon request. We then analyze the interaction of rational DAC nodes and clients as a dynamic game, with a Byzantine adversary that can corrupt and bribe the participants. We also define a notion of optimality for the DAC protocols, inspired by fairness and economic feasibility. Our main result shows that our protocol is optimal and guarantees security with the highest possible probability under reasonable assumptions on the adversary.

1 Introduction

Layer 2 systems [1,2,3] are an important approach to scaling the throughput of Layer 1 blockchains such as Ethereum. One of the key challenges in securing an L2 system is *data availability*: how to ensure that the state of the L2 system is always available and can be reconstructed when needed? This data is needed to safely restart the L2 system after a failure, and for basic operations such as deposits and withdrawals. The data availability problem comes up in other contexts as well, such as in decentralized storage systems [4,5,6].
There are three general approaches to data availability in L2 systems:

- *On-chain data:* Rollup systems [3] store all transaction data on a Layer 1 *parent* chain, such as Ethereum. These systems rely on the security of the L1 nodes to ensure that the data is always available.
- *Off-chain data stored by a Data Availability Committee (DAC):* Other systems such as StarkEx [7], zkPorter [8] and EigenLayr [9] use a DAC to store data off-chain across a number of trusted nodes [10]. While the DAC provides a gas-efficient alternative to on-chain data, these systems rely on the correct operation of the DAC nodes to ensure that the data remains available.
- *Off-chain data with (repeated) Data Availability Sampling (Celestium [11]):* An enhancement to DACs employs data availability sampling [12,13,14,15]

so that light clients, such as rollup users, can identify unavailable blocks created by the DAC without attempting to download the full block. This approach is being used by modular blockchains such as Celestia [16] and Polygon Avail [17] that specialize in preserving other chains' data. However, DAS does not remove the trust assumption placed on the DAC nodes for data availability, since it requires DAC members to reply to DAS queries for data recovery. DAS also cannot ensure that data remains permanently available [18].

Providing a standalone data availability service, such as Celestia and others, reflects a general trend towards modularity in the design of blockchains.

In this paper, we focus on the security of Data Availability Committees (DAC), namely the last two bullets on the previous page. A DAC consists of multiple DAC members, which we call *nodes*, that store copies of the data that should be made available (*e.g.*, data sent by the rollup sequencer). These nodes are expected to provide the data to querying clients in a timely manner. Since malicious DAC members can withhold the data, DACs typically replicate the data on each DAC node for fault tolerance. Thus, as long as one member is honest, rollup clients would receive the data upon request. Although the storage requirement of the DAC scales linearly in the number of nodes due to replication, this redundancy can be reduced through the use of erasure codes and polynomial commitments. For instance, the semi-AVID-PR scheme [19] uses linear erasure-correcting codes and homomorphic vector commitments to guarantee data availability as long as over 2/3 of the nodes faithfully follow the protocol.

A major drawback of DACs is the need to trust the DAC members. Consider a compromised DAC, where the adversary can prevent the reconstruction of the data, for example, by controlling more than 1/3 of the DAC members. Such a DAC can evolve the rollup state using unavailable transaction data, and withhold this data from the rollup clients. This prevents clients from issuing transactions, and enables the adversary to steal client funds through ransom attacks [20]. Thus, using a DAC hinders the goal of realizing a trust-minimized scaling architecture that relies solely on the security of the L1 chain for the safety and liveness of the rollup[1]. Liveness signifies that the clients can submit new transactions to the rollup system, and the system processes these transactions.

Data availability sampling (DAS) does not improve the liveness guarantees over the basic DAC architecture. If the DAC is not compromised, then DAS helps rollup clients verify that the rollup data is available without downloading all the data from the DAC. However, if the DAC is compromised, DAS provides no guarantees for data availability. The compromised DAC can update the rollup state with unavailable transactions, and ignore all DAS queries from the clients. Hence, DAS needs the trust assumption placed on the DAC members for liveness.

[1] Although current rollup systems typically rely on a single honest sequencer to evolve the rollup state, as long as the rollup data is available (*e.g.*, on the L1 chain), any rollup full node can step up to fulfill the sequencer's role if it fails.

Incentive-based data availability. One way to strengthen the security of a DAC is to rely on financial incentives to deter the DAC members from adversarial behavior such as withholding data and lazy validation, where the DAC members *pretend* as if the data was stored. There are solutions such as Proofs of Custody [21] using financial disincentives (*e.g.*, slashing) to encourage the lazy DAC members to store the entrusted data. However, as withholding data is not a provable offense, it not clear how to enforce the slashing of the adversarial members' stake when they do indeed store the data, yet refuse to reveal it upon request (even if DAS is being used). Moreover, any incentive-based data availability proposal must be analyzed in the face of rational DAC members who may respond to bribes, and Byzantine adversaries who may offer bribes.

Our main contribution is a DAC protocol that introduces a *slashing* mechanism for malicious DAC nodes that withhold data. The bulk of the paper is a technical analysis of the protocol, and proves its security under certain assumptions on the adversary's power. Moreover, we show that our protocol is optimal in a rigorous sense. We define the security model and the optimality notions in Sections 2 and 4.

We model the interactions of the DAC as a dynamic game involving multiple parties:

- DAC members, *i.e.*, **nodes**, are denoted by $\mathcal{P}_1, \ldots, \mathcal{P}_N$, where N is the number of nodes. These nodes store the data provided by an external entity.
- A **client** $\mathcal{V}$ sends a sequence of data queries to the N nodes. Every node can either respond to $\mathcal{V}$ with the requested data, or not respond. We assume the data held by the nodes is signed by the data provider, so that integrity of the response is easily verified. If a response contains incorrect data, it is treated as a non-response.
- A **contract** running on the L1 chain is used to resolve disputes and punish misbehaving DAC nodes. In particular, all N nodes are staked, and the stake is held in the contract. If the nodes do not respond to $\mathcal{V}$ with the requested data, $\mathcal{V}$ can send its query to the contract. In this case, the nodes are obliged to post their responses to the contract. If a node provably fails to do so, the contract can slash that node by confiscating part of its stake. Part of the slashed stake is given to the client as compensation and the rest is burned. The size of the per-node stake and the behavior of the contract are the key design decisions for a DAC protocol.

Nodes and clients are rational agents that seek to maximize their utilities. An adversary $\mathcal{A}$ who fully controls f corrupt nodes may try to bribe the remaining $N - f$ nodes to cause a client query to fail. This will make the requested data unrecoverable. Our goal is to design a DAC, so that under reasonable assumptions on the size of f and on the adversary's budget, every query from the client will succeed with probability at least $1 - \epsilon$, for some small ϵ.

Queries from the client model data requests needed for normal operations such as withdrawals. For instance, in a rollup system, clients might have to prove their account balances with respect to the latest state root, and they do

so by presenting a Merkle proof for their account. A non-responsive DAC storing the latest state can delay withdrawals by refusing to provide these Merkle proofs. In this case, each client can post a query to the contract, and force the nodes to place the requested proof on the L1 chain. Our model for the DAC system and the incentivize mechanism enforced by the contract has applications beyond data availability, and can be used to incentivize the honest participation of nodes in any committee outside the L1 chain that provides a service (*e.g.* Decentralized Oracle Networks [22,23]). We discuss use cases for our DAC system in Section 2.

The DAC protocol. Suppose every query requires at least k nodes out of N to respond either directly to the client, or to the contract, for the client to obtain an answer to its query. If no erasure coding is used and the data is replicated across all nodes, then $k = 1$, otherwise k could be bigger than 1.

The protocol proceeds in four steps:

- *step 1:* the client $\mathcal{V}$ sends its query to all DAC nodes over the network.
- *step 2:* if k or more nodes respond, then the client obtains the requested data and the protocol terminates.
- *step 3:* if by a certain timeout the client does not receive k responses, it posts its query to the contract on chain. For this purpose, the client has to send a base payment to the contract, which is needed to deter spamming clients. We discuss the choice of client payment amount in Section 6.
- *step 4:* all N nodes are then asked to post their responses to the query on chain. The protocol terminates once a certain timeout is reached.

It remains to describe what the contract does once the timeout is reached in step 4. Every node that does not post its response to the contract by the timeout loses part or all of its stake. The precise *slashing function* is explained in Section 3. Moreover, if by the timeout in step 4 the client does not obtain an answer to its query through the responses, the client is compensated by the contract using the funds obtained from the slashed nodes.

The question is how to analyze the security and performance of a contract in comparison to other contracts. In Section 4 we present four desirable properties that a slashing function should satisfy. Informally, these properties are:

- *Symmetry.* Motivated by fairness, the slashing function does not depend on the identities of the nodes, only on their actions.
- *No Reward.* The slashing function does not pay out any rewards to the responsive nodes. This is motivated by economic feasibility as the contract should maintain a non-negative balance, and discourage the nodes from forcing an on-chain interaction for extra payoff rather than answering over the network. (No rewards rule does not rule out flat rewards by other means.)
- *Security Under No Attack.* The slashing function ensures that the client promptly learns the correct response to its query, if the adversary does not offer any bribes. This captures a minimal notion of security.

– *Minimal Punishment.* The slashing function keeps the slashed amounts of non-responsive nodes at a minimum when the client obtains an answer to its query. Thus, when most nodes are responsive, those that fail to respond due to benign failures, *e.g.*, crash faults, are not heavily penalized.

We then define a notion of optimality for these functions:

Definition 1 (Informal). *A slashing function is* **optimal** *with respect to a set of slashing functions $\mathcal{F}$, if the function satisfies the following two conditions: (i) upon sending its query to the contract, the client obtains an answer with the maximum probability from among all the functions in $\mathcal{F}$ given the worst adversary, and (ii) when the client obtains an answer, the function imposes the minimal punishment on non-responsive nodes from among the functions in $\mathcal{F}$.*

In Section 4, we show that our slashing function is optimal for both *risk-neutral* and *risk-averse* nodes among the set of all functions that satisfy the four desirable properties described above. We also analyze the security of a dynamic game among a rational client and the DAC nodes. We identify the conditions under which the client obtains an answer to its query without calling the contract. The analysis of Section 4 is the most technical part of the paper, and is our core contribution.

Evaluation. In Section 5, we evaluate the real-world performance of our optimal contract. To match the number of Ethereum validators and the minimum value that can be staked as an independent validator on Ethereum, we set the total number of DAC nodes to $N = 300,000$ and the amount staked per node to 32 ETH. Then, given risk-neutral nodes, the adversary has to offer a total bribe of $\approx 3.2 \cdot 10^3$ ETH ($\approx$ 3.9 million USD[2]) to the nodes, to reduce the security probability per query by a tiny amount, namely to reduce the probability that a client learns the answer to its query from 100% to 99.9%. To prevent clients from learning the answers over repeated queries, the adversary has to spend at least 3.9 million USD *for each query*. As our contract is optimal, no other contract can force the adversary to pay a higher bribe for the same security probability. The minimum bribe needed by the adversary to reduce the security probability increases as N or the collateral grows, or as the nodes become more risk-averse.

2 Model

Notation. We denote the security parameter by λ. We say that an event happens with negligible probability, if its probability, as a function of λ, is $o(1/\lambda^d)$ for all $d > 0$. We say that an event happens with overwhelming probability if it happens except with probability negligible in λ. If an event happens with probability $q + \text{negl}(\lambda)$ or $q - \text{negl}(\lambda)$, where q is a non-negligible constant, for simplicity, we say that the event happens with probability q. We assume that

[2] Ethereum to USD conversion rate, 1 ETH $\approx$ 1231.0 USD, is the average Ethereum price on July 15, 2022 [24].

except with probability negligible in λ, the contract implements the specified slashing function correctly, the underlying cryptographic primitives are secure, and messages can be posted to the contract within bounded time. We use the shorthand $[N]$ to denote the set $\{1, 2, \ldots, N\}$.

Environment and the Adversary. Time is slotted, and the clocks of the client and nodes are synchronized[3]. Messages, *e.g.*, queries and replies, can only be sent at the beginning of a slot, and are delivered to the recipient by the end of the same slot by the environment $\mathcal{Z}$.

Adversary $\mathcal{A}$ is a probabilistic polynomial time algorithm. Before the execution starts, $\mathcal{A}$ corrupts f nodes, which are subsequently called adversarial. These nodes can deviate from the protocol arbitrarily (Byzantine faults) under $\mathcal{A}$'s control, which has access to their internal states. The remaining $N - f$ nodes and the client are utility maximizing agents and can choose any action that gives them a higher utility. In the subsequent analysis, we will assume that $\mathcal{P}_i$, $i = N - f + 1, \ldots, N$ represent the adversarial nodes, and $f \leq N - k$. Otherwise, it is impossible to guarantee the recovery of the answer to a query as the adversarial nodes can withhold their responses from the client and the contract.

Before the protocol execution starts, the adversary can also offer *bribes* to the *remaining* nodes and the client subject to constraints. It has a supply of p_0 coins, which can be distribute to any subset of the nodes as additional payoff if the nodes adopt an adversarial action during the game. Similarly, the adversary can give up to p_1 coins to the client if it adopts an adversarial action. Such an adversary is called a (p_0, p_1)-adversary. When the bribe offered to the client is irrelevant, we use the notation p_0-adversary. (p_0 and p_1 are adversary's resources that are *beyond* the f nodes corrupted by the adversary.) Upon hearing an offer, each participant can independently choose to accept or reject the bribe depending on the expected utility. Once a participant accepts the bribe, the adversary can monitor through the environment and the contract if the specified action was taken. Although the action and the exchange of the bribe might not happen atomically, the adversary and nodes can ensure that no party deviates from its promise through a trusted third party, or repeated games (*cf.* Section 4.4).

Actions, Payoffs, and the Game. We next describe the dynamic game played by the client and the DAC nodes. Before the game starts, the client $\mathcal{V}$ and the nodes are input a single query by the environment $\mathcal{Z}$. Given a query, each node $\mathcal{P}_i$ can instantaneously generate a response c_i, called the *clue*. We assume that the correctness of these clues can be verified by the clients and the contract[4]. The contract accepts a clue by a node if and only if it is the first correct response by the node to a query posted to the contract. It records the time slots when each query or clue was received, in a contract state. At the beginning of each slot, the participants learn about the state recorded at the end of the previous slot.

[3] Bounded clock offsets can be captured by the network delay.

[4] For instance, correctness of the data shards in PoS Ethereum can be verified with respect to a KZG commitment on the blockchain [25,18,15].

Let p_s be the amount staked by a node to function as a DAC member. It costs p_c coins for the client to send a query to the contract, and p_w coins for each node to prepare and post the corresponding clue to the contract. It is free to send a clue to the client over the network. These parameters are summarized in Table 1. We assume that each node starts the game with a baseline payoff of $C = p_s + p_w$, as it has p_s coins staked in the contract, and is assumed to have enough funds to post clues to the contract during the game[5]. The client starts the game with an initial payoff of 0.

The actions available to the client $\mathcal{V}$ and a node $\mathcal{P}$ at any slot t are denoted as follows:

- $\mathcal{S}_{\mathsf{r}}$: $\mathcal{P}$ sends a correct clue to $\mathcal{V}$ over the network at slot t.
- $\mathcal{S}_{\mathsf{q}}$: $\mathcal{V}$ sends a query to the contract for the first time at slot t.
- $\mathcal{S}_{\mathsf{p}}$: Replying to a query, $\mathcal{P}$ sends a correct clue to the contract for the first time at slot t.

The notation $\neg(.)$ is used to denote the opposite of the specified action. At any time slot, a node can take an action (a, b), where $a \in \{\mathcal{S}_{\mathsf{r}}, \neg\mathcal{S}_{\mathsf{r}}\}$ and $b \in \{\mathcal{S}_{\mathsf{p}}, \neg\mathcal{S}_{\mathsf{p}}\}$. Similarly, the client can take an action from $\{\mathcal{S}_{\mathsf{q}}, \neg\mathcal{S}_{\mathsf{q}}\}$. Although the clients and nodes can exchange messages other than queries and clues, only the queries, clues or their absence can lead to a change in their payoffs. Since the participants play a dynamic game, the actions chosen at later slots can depend on the actions observed at the earlier ones.

The game ends, and the payoffs are realized at the beginning of slot T_{answer}. If $\mathcal{V}$ finds out the correct answer to its query through the clues, either posted to the contract or sent over the network, by slot T_{answer}, it receives a payoff of p_f coins. We set $T_{\mathsf{answer}} = 4$ though it can be any sufficiently large constant. In our model, T_{answer} should be at least 4 to guarantee any meaningful security. The payoffs of the participants depend on the bribes p_0 and p_1, the collateral p_s, the variables p_f, p_c, p_w selected by $\mathcal{Z}$, and the contract's *slashing function*.

Utility of a participant is given as a function $U(.)$ of the payoff obtained at the end of the game. In the subsequent sections, we will first consider risk-neutral nodes with a linear utility function $U(x) = x$, where x is the net payoff at the end. We will then analyze risk-averse nodes with a strictly concave utility function of the form $U(x) = (x)^\nu$, where $\nu \in (0, 1)$. We do not consider risk-seeking nodes with strictly convex utility functions, *e.g.*, $U(x) = (x)^\nu$, $\nu > 1$, as such a function violates the law of diminishing marginal utility for the payoffs.

We will later also consider a sub-game that focuses exclusively on the interaction between the nodes and the contract. In the game, a query appears in the contract at some slot t, and the nodes choose to post clues or not at slot $t + 1$, after which the payoffs are realized. These payoffs depend on the bribe p_0, the collateral p_s, the cost p_w, and the slashing function.

Security. We say T_{answer}-security is satisfied if the client receives k or more correct clues from the nodes either over the network or through the contract by the *beginning* of slot T_{answer} with overwhelming probability.

[5] For risk-neutral nodes, the baseline is normalized to be 0.

Application. The game above models the withdrawal of client funds from a blockchain or rollup. Each client has an account, represented as a key-value pair, and the balances of these accounts constitute the blockchain state. The hashes of the key-value pairs are organized in a vector commitment, *e.g.*, a sparse Merkle tree, with a constant size commitment, called the state root. The state data is preserved by the DAC nodes and state commitments are posted to the chain.

To prove its account balance, a client requests a witness from the nodes for the inclusion of its account within the latest state. If it does not receive a witness over the network, the client can complain on a *smart contract* by sending a query that contains the hash of the account's key-value pair. If the hash is a hiding commitment, the client can also ensure that no observer learns its balance. It can always prove its balance to a select third party by revealing the key-value pair at the pre-image of the hash, the latest state root on chain, and the witness.

Upon receiving a query, the contract expects a witness to be provided by the DAC nodes within a bounded time, *e.g.*, the chain's confirmation latency. Correctness of this witness can be verified by the contract and the client with respect to the state commitment on the chain. If the query is for an account not included in the latest state, the nodes can convince the contract of this fact via a proof of non-inclusion. If there are multiple queries, instead of sending the witness for each query, the nodes can compute a SNARK proof that verifies the inclusion of all the queried accounts within the state. Clients can then verify the inclusion of the queried accounts by checking the proof with respect to the latest state root, and the hashes of the queried accounts. Succinctness of the SNARK proof enables achieving bounded delay on the response time.

3 The Optimal Contract

Parameter	Explanation
N	Number of nodes
p_0	Total payoff the adversary can offer to the nodes
p_1	Total payoff the adversary can offer to the client
p_{comp}	Compensation for the client if reconstruction fails
p_f	Client's payoff from a valid reply within 4 slots
p_c	Cost of sending a query to the contract
p_w	Cost of constructing and sending a clue to the contract
p_s	Collateral per node

Table 1: Parameters in our model

A contract can reward or punish the nodes depending on whether it received clues from the nodes for a query within a timeout period. We normalize this timeout to be a single slot for all contracts[6]. Let $x_i = 1$ if the node $\mathcal{P}_i$ sends a

[6] In a network with temporary partitions, the timeout can be increased to guarantee the timely inclusion of the messages sent to the contract.

valid clue at slot $t+1$ in response to a query posted at some slot t, and $x_i = 0$ otherwise. We characterize a contract by a slashing function f that maps actions $\mathbf{x} = (x_1, \ldots, x_N) \in \{0,1\}^N$ to payoffs $(f_1(\mathbf{x}), \ldots, f_N(\mathbf{x})) \in \mathbb{R}^N$ for the nodes, and the payoff $f_{\mathcal{V}}(\mathbf{x}) \in \mathbb{R}$ for the client. Since the contract cannot punish the nodes more than the staked collateral, $f_i(\mathbf{x}) \geq -p_s$ for every action $\mathbf{x} \in \{0,1\}^N$. We will hereafter use slashing function and the contract interchangeably.

The proposed optimal contract and the associated slashing function is parameterized by a small number $\epsilon > 0$:

$$f_i(\mathbf{x}) = \begin{cases} 0 & \text{if } x_i = 1 \\ -p_s & \text{if } \sum_{j=1}^{N} x_j < k \text{ and } x_i = 0 \\ -p_w - \epsilon & \text{if } \sum_{j=1}^{N} x_j \geq k \text{ and } x_i = 0 \end{cases}$$

$$f_{\mathcal{V}}(\mathbf{x}) = \begin{cases} 0 & \text{if } \sum_{j=1}^{N} x_j \geq k \\ p_{\text{comp}} & \text{if } \sum_{j=1}^{N} x_j < k \end{cases}$$

Here, $p_{\text{comp}} < p_s, p_f$, and $p_{\text{comp}} > p_c$ to ensure that the client's net payoff stays above zero if it does not receive sufficiently many clues through the contract.

The contract burns, *i.e.*, slashes the collateral p_s put up by each node that has not sent a valid clue by the end of slot $t+1$, if there are less than k clues. In this case, the contract also awards p_{comp} of the slashed coins to $\mathcal{V}$. Otherwise, if there are k or more clues in the contract by slot $t+1$, it punishes the non-responsive nodes by a modest amount, namely $p_w + \epsilon$.

4 Analysis

In Section 4.1, we formalize the desirable properties and notions of optimality for slashing functions. In Section 4.2, we show that the slashing function of Section 3 is optimal for risk-neutral and risk-averse nodes. In Section 4.3, we generalize the analysis to a dynamic game with a rational client. In Section 4.4, we analyze a repeated game played between the nodes and the adversary.

4.1 Contract Properties

The desirable properties for a slashing function f, introduced in Section 1, are formalized below:

- *A1: Symmetry.* A slashing function f is symmetric if $f(\pi(\mathbf{x})) = \pi(f(\mathbf{x}))$ for every action $\mathbf{x} \in \{0,1\}^N$ and permutation π.
- *A2: No Reward.* A slashing function f offers no rewards if for every action $\mathbf{x} \in \{0,1\}^N$, $f_i(\mathbf{x}) \leq 0$, $\forall i \in [N]$, and $f_{\mathcal{V}}(\mathbf{x}) + \sum_{i \in [N]} f_i(\mathbf{x}) \leq 0$.
- *A3: Security Under No Attack.* A slashing function f guarantees security under no attack if for all $(0,0)$-adversaries, it achieves T_{answer}-security with overwhelming probability in all Nash equilibria of the game.

– *A4: B-Minimal punishment.* A slashing function f offers B minimal punishment if for every action $\mathbf{x} \in \{0,1\}^N$ such that $\sum_{i=1}^{N} x_i \geq k$, we have that $f_i(\mathbf{x}) \geq -B$ for all $i \in [N]$.

Definition 2. *A slashing function f is said to be* **compliant** *if it satisfies the axioms A1–A3, and the axiom A4 for some constant $B \in \mathbb{R}^+$.*

Definition 3. *A compliant slashing function f is said to be (p_0, q)-***tolerant** *if for all p_0-adversaries, when a query is received by the contract at some slot t, there are k or more correct clues in the contract at slot $t+1$, with probability at least q, in all Nash equilibria.*

The value q of a (p_0, q)-tolerant contract can be interpreted as the minimum probability for security given that the client received no responses over the network and sent its query to the contract.

We next introduce two notions of optimality for the contract. A security-optimal function ensures that for any p_0, security is violated with the minimum possible probability in the equilibrium with the largest failure probability.

Definition 4. *A compliant slashing function f is said to be* **security-optimal** *if for all $p_0 \geq 0$, there exists a $q_0 \in [0,1]$ such that f is (p_0, q_0)-tolerant, and there does not exist any compliant, (p_0, q)-tolerant function f', where $q > q_0$.*

A punishment-optimal contract imposes the minimum punishment on the unresponsive nodes (*e.g.*, due to benign errors) if security was not compromised.

Definition 5. *A compliant slashing function f is said to be ϵ-***punishment-optimal** *if it satisfies B-minimal punishment, and no compliant slashing function f' can satisfy B'-minimal punishment for some $B' < B - \epsilon$.*

Finally, we combine the two notions of optimality in a single definition:

Definition 6. *A family of slashing functions f_ϵ, parameterized by ϵ, is said to be* **optimal** *if each member f_ϵ of the family is compliant, security-optimal and* ϵ-punishment-optimal.

4.2 Analysis of The Optimal Contract

We prove the following theorem for risk-neutral and risk-averse nodes.

Theorem 1. *The family of slashing functions described in Section 3 is optimal.*

Theorem 1 follows from Theorems 2, 3, and 4. Their proofs for risk-neutral and risk-averse nodes are given in Appendices A and B respectively.

We first showing that the slashing function is compliant:

Theorem 2. *Each slashing function from Section 3, parameterized by $\epsilon > 0$, satisfies symmetry (A1), no reward (A2), security under no attack (A3), and $(p_w + \epsilon)$-minimal punishment (A4).*

The axioms A1, A2 and A4 follow by inspection, whereas A3 is shown by Lemma 1. Proof of Lemma 1 is given in Appendices A and B for risk-neutral and risk-averse nodes respectively.

Lemma 1. *Given the slashing function of Section 3, for any* $(0,0)$*-adversary* $\mathcal{A}$*,* 4*-security is satisfied with overwhelming probability in all Nash equilibria.*

When $p_{\text{comp}} > p_c$, the client is incentivized to send its query to the contract if it receives less than k clues over the network. Then, the nodes post their clues to the contract to avoid slashing of their stakes, and the contract ensures security with overwhelming probability.

Remark 1. If $p_{\text{comp}} \leq p_c$, for any contract that offers no rewards to the nodes, and for any $(\epsilon, 0)$-adversary where $\epsilon \geq 0$, there exists a Nash equilibrium such that 4-security is violated with overwhelming probability. Consider the action profile, where the nodes do not send their clues to the client $\mathcal{V}$ over the network, and do not post their clues to the contract. Given these actions, if $p_{\text{comp}} \leq p_c$, $\mathcal{V}$'s payoff can at most be 0, and the maximum payoff is achieved if $\mathcal{V}$ does not send a query to the contract, even when it does not receive clues over the network. In this case, the normalized payoff of each node becomes 0 as well, which is the maximum payoff attainable by any node. Hence, the nodes do not have any incentive to deviate from the action profile above, which constitutes a Nash equilibrium.

We next show that the slashing function is ϵ-punishment optimal.

Theorem 3. *Consider a slashing function that is symmetric (A1), offers no rewards (A2), and satisfies* B*-minimal punishment for some* $B < p_w$ *(A4). Then, for* $k > 1$*, there exists a* $(0,0)$*-adversary* $\mathcal{A}$ *and a Nash equilibrium, where* 4*-security is violated with non-negligible probability. Thus, no compliant slashing function can satisfy* B*-minimal punishment for some* $B < p_w$*.*

When $B < p_w$, punishment for a node that does not post its clue to the contract while the other nodes send their clues is smaller than the cost of posting the clue. This leads to a free-rider problem, and results in an equilibrium with a non-negligible failure probability for security, where each non-adversarial node trusts the others to send clues to the contract.

Finally, we prove security-optimality:

Theorem 4. *The slashing function of Section 3 is security optimal.*

Consider the sub-game, where the contract receives a query at some slot t. For a given contract and utility function $U(x) = x^\nu$, let $q_{\text{v}}^{\mathcal{A}}$ denote the probability that given a p_0-adversary $\mathcal{A}$, there are less than k valid clues in the contract at slot $t+1$ in the Nash equilibrium with the largest probability of failure. Then, the proof of Theorem 4 for risk-neutral nodes follow from Theorem 5:

Theorem 5. *Suppose $p_0 < (N - f - k + 1)(p_s - p_w)$ and the nodes are risk-neutral with the utility function $U(x) = x$. Then, for any p_0-adversary $\mathcal{A}$, the slashing function of Section 3 satisfies*

$$q_{\mathrm{v}}^{\mathcal{A}} \leq q^* = \frac{p_0}{(N - f - k + 1)(p_s - p_w)}$$

Moreover, there exists a p_0-adversary $\mathcal{A}$ such that for any compliant slashing function, $q_{\mathrm{v}}^{\mathcal{A}} \geq q^*$.

The p_0-adversary $\mathcal{A}$ of Theorem 5 offers a bribe of $\frac{p_0}{N-f-k+1}$ to $N-f-k+1$ non-adversarial nodes, *e.g.*, $\mathcal{P}_i$, $i \in [N - f - k + 1]$. In return, it requests these nodes to collectively withhold their clues from the contract with probability q^*.

Remark 2. If $p_0 \geq (N - f - k + 1)(p_s - p_w)$, there exists a Nash equilibrium, where 4-security is violated with overwhelming probability. Adversary offers a payoff of $p_s - p_w$ to each of the $N - f - k + 1$ nodes, and requests them to withhold their clues from the contract. In the equilibrium, the offer is accepted and the nodes do not post their clues to the contract.

Remark 3. Sending repeated queries to the contract does not reduce the failure probability by more than a linear factor in latency. Suppose the client $\mathcal{V}$ is allowed to send the same query to the contract up to ℓ times. Then, if there are less than k valid clues in the contract at slot $t + 1$, $\mathcal{V}$ might want to repeat the sub-game up to ℓ times with the hope of eventually learning the answer to its query. In this case, the adversary $\mathcal{A}$ can offer a payoff of $\frac{p_0}{N-f-k+1}$ to the nodes $\mathcal{P}_i$, $i \in [N - f - k + 1]$, and in return, ask them to collectively withhold their clues in *all* of the games with probability q^*/ℓ. As in the proof of Theorem 5, this adversary ensures $q_{\mathrm{v}}^{\mathcal{A}} \geq q^*/\ell$ for any compliant slashing function.

Finally, we characterize the failure probability for the optimal contract. Suppose the contract of Section 3 is $(p_0, 1 - q^*_{p_0,\nu})$-tolerant per Definition 3, where the failure probability $q^*_{p_0,\nu}$ is depends on the total bribe p_0 and the nodes' utility function $U(x) = x^\nu$, *e.g.*, $q^*_{p_0,1} = q^*$ by Theorem 5. Although Theorem 4 proves that the contract of Section 3 is security optimal, unlike Theorem 5, its proof does not provide an explicit expression for $q^*_{p_0,\nu}$ when $\nu < 1$, *i.e.*, for risk-averse nodes. Instead, in Appendix C, we identify an optimization problem whose solution gives $q^*_{p_0,\nu}$. As the optimization problem is not convex for $\nu < 1$, in lieu of solving the problem, we provide bounds on $q^*_{p_0,\nu}$ that characterize its asymptotic behavior in terms of ν, p_0 and N.

4.3 Analysis of the Dynamic Game

In this section, we analyze the interaction among a rational client and the nodes during the dynamic game. For a specified slashing function, let $q(p_0, \nu)$ denote the maximum probability that 4-security is violated in the Nash equilibrium with the largest probability of failure, across all p_0-adversaries.

Theorem 6 shows that when $k > 1$, the slashing function of Section 3 achieves the minimum $q(p_0, \nu)$ among all compliant slashing functions, and this probability equals $q^*_{p_0,\nu}$. Proofs of the subsequent theorems are presented in Appendix D.

Theorem 6. *Consider (p_0, p_1)-adversaries such that $p_1 < p_{\text{comp}} - p_c$ and $p_0 < (N - f - k + 1)(p_s - p_w)$. Then, for the slashing function of Section 3, it holds that $q(p_0, \nu) \leq q^*_{p_0,\nu}$.*

Moreover, given any compliant slashing function, if $k > 1$, then, there exists a $(p_0, 0)$-adversary and a subgame perfect equilibrium such that 4-security is violated in the equilibrium with probability $q^*_{p_0,\nu}$.

Theorem 6 proves that even if the adversary does not offer any bribe to the client, *i.e.*, $p_1 = 0$, if $k > 1$, there exists a subgame perfect equilibrium where security is violated with the maximum probability $q^*_{p_0,\nu}$.

Remark 4. If $p_1 > p_{\text{comp}} - p_c$, there exists a Nash equilibrium, where 4-security is violated with overwhelming probability. Suppose the adversary $\mathcal{A}$ asks the nodes to *not* send their clues to $\mathcal{V}$ or to the contract, and requests $\mathcal{V}$ to *not* post its query to the contract. If $\mathcal{V}$ never sends its query to the contract, nodes achieve a strictly better utility by accepting the adversary's offer. Similarly, $\mathcal{V}$ cannot increase its utility by deviating from the adversarial action. This is because, if $\mathcal{V}$ rejects its bribe and sends a query to the contract, given the nodes' actions, its payoff becomes $p_{\text{comp}} - p_c$, less than the bribe p_1. Hence, given $\mathcal{A}$, the specified actions indeed constitute a Nash equilibrium.

Theorem 7 analyzes the game when $k = 1$.

Theorem 7. *Consider any compliant slashing function and (p_0, p_1)-adversaries such that $p_1 < p_{\text{comp}} - p_c$ and $p_0 < (N - f - k + 1)(p_s - p_w)$. Suppose there are N nodes, and $k = 1 \leq N - f$. Then, if p_1 satisfies*

$$(1 - q^*_{p_0,\nu})(p_f - p_c + p_1)^\nu + q^*_{p_0,\nu}(p_f - p_c + p_1 + p_{\text{comp}})^\nu \geq (p_f)^\nu, \qquad (1)$$

there exists a (p_0, p_1)-adversary and a subgame perfect equilibrium such that 4-security is violated with probability at least $q^*_{p_0,\nu}$, i.e., *$q(p_0, \nu) \geq q^*_{p_0,\nu}$.*

Via Theorems 6 and 7, for all values of k and all (p_0, p_1)-adversaries with a sufficiently large p_1, the slashing function of Section 3 achieves the minimum possible failure probability for 4-security among all compliant slashing functions. If p_1 satisfies formula (1), then the adversary can incentivize $\mathcal{V}$ to send a query to the contract regardless of whether $\mathcal{V}$ received clues over the network. This in turn discourages the nodes from sending clues over the network, and helps sustain an equilibrium where security rests solely on the clues sent to the contract. In this context, slashing function of Section 3 minimizes the failure probability for security, which becomes $q^*_{p_0,\nu}$.

On the other hand, if p_1 is too small to satisfy formula (1), p_0 is sufficiently small (but non-zero) and $k = 1$, given the optimal slashing function of Section 3, 4-security can be satisfied, without any query sent to the contract, with probability exceeding $q^*_{p_0,\nu}$. This prevents the adversary from making the contract the default method for retrieving the data and bloating the blockchain.

Theorem 8. *Consider the slashing function of Section 3 and (p_0, p_1)-adversaries such that $p_1 < p_{\text{comp}} - p_c$, $p_0 < (N-f)p_w$, and p_1 satisfies*

$$(1 - q^*_{p_0,\nu})(p_f - p_c + p_1)^\nu + q^*_{p_0,\nu}(p_f - p_c + p_1 + p_{\text{comp}})^\nu < (p_f)^\nu.$$

Suppose there are N nodes, and $k = 1 \leq N - f$. Then, 4-security is satisfied with overwhelming probability in all Nash equilibria, without the client sending its query to the contract.

When $\nu = 1$, *i.e.*, for risk-neutral nodes, formula (1) implies $p_1 \geq p_c$. As p_c can be as small as the gas cost of sending a query, for most (p_0, p_1)-adversaries, we expect p_1 to exceed p_c, *i.e.* to satisfy formula (1).

4.4 Repeated Games

Although the adversary can offer any bribe and specify any action in return, exchange of the bribe and the execution of the action do not necessarily happen atomically. This might discourage cooperation between the nodes and the adversary as they can renege on their promises.

One way the parties can ensure atomicity is through a *trusted third party*. It can take the custody of the nodes' internal states, along with the adversary's bribe, and adjust the payoffs after the game. Alternatively, the adversary and nodes can sustain a cooperative equilibrium over the repeated instances of the single-stage query-response game analyzed in Section 4.2, with the help of a common random coin. For repeated games to be feasible, we assume that the nodes have more coins than the collateral p_0, enabling them to absorb occasional losses due to slashing in return for long-term profit. Similarly, we assume that the adversary can continue to offer bribes each new game. Let δ denote the discount rate. We consider the optimal contract of Section 3 in the following analysis.

Suppose $p_0 < (N - f - k + 1)(p_s - p_w)$, and there is a query at the contract at slot t. By Section 4.2, for any p_0 and $\nu \in (0, 1]$, there exists a p_0-adversary and Nash equilibrium, where less than k nodes send their clues to the contract at slot $t + 1$ with probability at least $q^*_{p_0,\nu}$. The utilities of the nodes in the equilibrium are feasible and strictly individually rational as they are at least as large as the maximum utility, $(C - p_w)^\nu$, any node can get without cooperating with the adversary. Thus, by the Nash folk theorem, we can state the following:

Theorem 9. *There exists a discount rate $\delta^* < 1$ such that for all $\delta > \delta^*$, there is a subgame perfect equilibrium of the repeated game with the same expected utilities for the nodes per game as the single-stage game. Moreover, at each game, less than k clues are posted to the contract with probability at least $q^*_{p_0,\nu}$.*

By the proof of Theorem 4, no adversary can guarantee less than k clues to be posted to the contract with probability larger than $q^*_{p_0,\nu}$ without making the utility of a node less than $(C - p_w)^\nu$, its minimax utility. Consequently, $q^*_{p_0,\nu}$ is the maximum failure probability that can be sustained through repeated games.

To maintain the aforementioned equilibrium, the adversary and nodes can use a grim trigger strategy: Consider the adversary of Theorem 5. The common random coin is flipped before each game, and obtains the value 0 with probability $q^*_{p_0,\nu}$ and the value 1 with probability $1 - q^*_{p_0,\nu}$. Before each game, the adversary offers its bribe. Then, the coin is flipped. If the outcome is 0, the bribed nodes are asked to withhold their clues from the contract. At this point, if any of the bribed nodes sends its clue to the contract, the adversary stops offering any future payoff to that node. Similarly, if the adversary fails to offer a sufficient bribe before the coin is flipped, the nodes stop cooperating with the adversary.

5 Evaluation

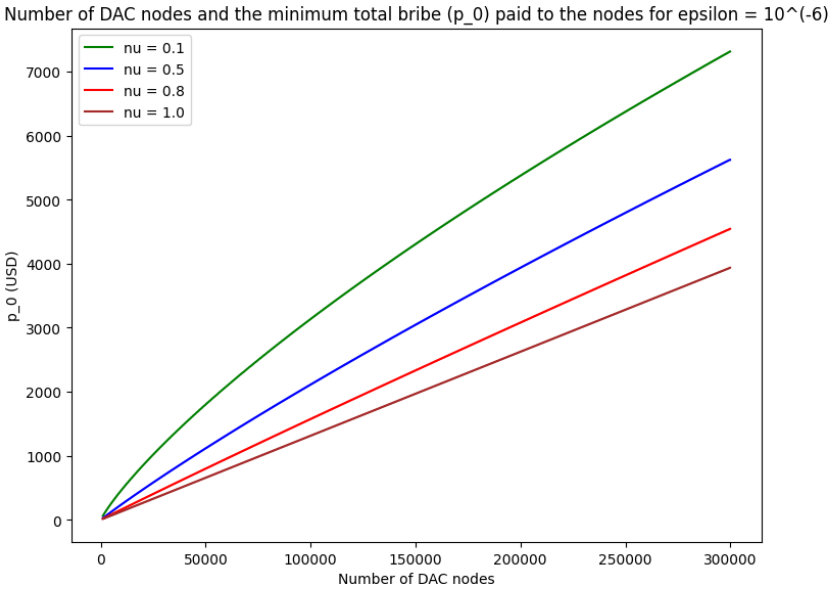

Fig. 1: Lower bounds on the total bribe, p_0, needed to ensure that the probability the client does not obtain the answer to its query is $\epsilon = 10^{-6}$, as a function of the number of DAC nodes N, and the utility functions $U(x) = x^\nu$, $\nu = 0.1, 0.5, 0.8, 1.0$.

We next calculate the bribe p_0 needed to violate security in the equilibrium with the largest failure probability, when a query is sent to the optimal contract of Section 3 on Ethereum. When the clues are SNARK proofs as argued in Section 2, assuming that sending and verifying a SNARK proof on Ethereum requires 650000 gas [26], and the gas cost is 34.77 Gwei[7], we estimate the cost of posting a clue to the contract as $p_w \approx 0.0226$ ETH. We set the collateral p_s

[7] The gas cost is the average gas price for July 15, 2022 [27].

to be 32 ETH to match the minimum amount that can be staked in Ethereum by an independent node. Assuming that the adversary can control up to 1/3 of the N DAC nodes, and clues from 1/3 of the nodes are sufficient to recover the answer to the client queries, we set $N-f-k+1$ to be $N/3$. The 1/3 bound for the adversarial DAC nodes matches the maximum tolerable adversary fraction shown for the security of Casper FFG [28], the finality gadget of PoS Ethereum. We consider $N < 300,000$, which has the same magnitude as the number of validators on PoS Ethereum [29].

Let ϵ denote the maximum failure probability for DAC security that the clients are willing to tolerate. Suppose $\epsilon = 0.1\%$. For risk-neutral nodes, Theorem 5 implies that $\epsilon = \min(1, \frac{1}{N-f-k+1}\frac{p_0}{p_s-p_w})$. For risk-averse nodes with the utility function $U(x) = x^\nu$, ϵ is the solution to the optimization problem in Theorem 10, which is upper and lower by Theorem 11 in Appendix C. Using the parameters identified above, the formula for ϵ for risk-neutral nodes and the bounds for risk-averse nodes, we calculate the following bounds[8] for the minimum bribe p_0 needed to violate security with probability $\epsilon = 10^{-3}$ (0.1%), as a function of the utility parameter ν (Details are presented in Appendix C).

ν	Lower bound on p_0	Upper bound on p_0
0.1	3197.9 ETH (3.9 Million USD)	13257.7 ETH (16.3 Million USD)
0.5	3197.9 ETH (3.9 Million USD)	6082.5 ETH (7.5 Million USD)
0.8	3197.9 ETH (3.9 Million USD)	3977.5 ETH (4.9 Million USD)
1.0	3197.9 ETH (3.9 Million USD)	3197.9 ETH (3.9 Million USD)

Table 2: Lower and upper bounds on p_0 in ETH and USD as a function of the utility parameter ν, where 1 ETH $\approx$ 1231.0 USD, the number of DAC nodes N is $300,000$, and the failure probability is $\epsilon = 0.1\%$.

The exact value of p_0 increases as ν decays, *i.e.*, as the nodes become more risk averse. This increase becomes more stark at small values of the maximum failure probability ϵ. To illustrate this point, we plot the lower bound on p_0 as a function of $N \in [1, 300000]$, for $\nu = 0.1, 0.5, 0.8, 1.0$ and $\epsilon = 10^{-6}$ (as opposed to $\epsilon = 0.1\%$) on Figure 1, where the lower bound curve increases as ν becomes smaller. Since Table 2 considers $\epsilon = 10^{-3}$, unlike the case with $\epsilon = 10^{-6}$, the lower bound expression for p_0 does not increase as ν becomes smaller.

6 Discussion and Future Work

Preventing centralization of storage. DAC members have an incentive to pool their resources and pay for a central data repository, *e.g.*, a cloud provider. They then answer the client queries by querying the central repository, and split

[8] Ethereum to USD conversion rate, 1 ETH $\approx$ 1231.0 USD, is the average Ethereum price on July 15, 2022 [24].

the cost of the repository among themselves. However, if this single repository loses the data, then all is lost. Thus, a DAC protocol should discourage data centralization, and this can be done using a cryptographic Proofs of Replication (POR) [30] that forces every node to store a different incompressible version of the data. However, POR introduces a significant computation overhead. Interestingly, the data centralization problem is not addressed by data availability or storage systems such as Celestia [16] and Arweave [4].

While our protocol does not solve the problem, arguably, it discourages data centralization. A node that participates in a centralization scheme is putting its trust in the repository to preserve the data. However, the repository has little to lose if the data is lost, while the node will lose its entire stake. Hence, the node is incentivized to store the data locally rather than to trust a third party.

Preventing client DoS attack. Clients can send queries to the contract frequently, at the cost of p_c coins per query. Although p_c can be as low as the gas cost of posting an account information on chain (*cf.* Application in Section 2), which implies a potential DoS vector, the contract can increase this cost to disincentivize DoS attacks. The value of p_c can even be adaptively chosen as a function of the number of queries to reduce congestion. Then, as long as p_c is not subsidized by the bribe p_1, no rational client would send a query unless the nodes withhold their clues. However, p_c should not be too high as that would hurt the contract balance by requiring a high p_{comp} (*cf.* Remark 4), and discourage rational clients from sending queries for accounts with smaller balances (*i.e.*, p_f). An interesting future work is to determine the optimal p_c that would not impose a high burden on most accounts while making spamming attacks costly.

Utility functions. The analysis in Section 5 demonstrates how risk-aversion implies a higher bribe for the adversary to violate security. However, the exact shape of the utility function depends on the marginal utility for the coin in which the payoffs are provided. Quantifying this marginal utility and identifying the correct function is important future work to accurately assess the affect of bribery on security.

Bribery. Besides external offers, bribery quantifies the nodes' incentives to withhold data, *e.g.*, to prevent a competitor from learning about a transaction. In general, bribery captures all incentives (*e.g.*, MEV) that might cause the rational nodes to deviate from the prescribed consensus and data availability protocols.

Collusion. Collusions among nodes can help mitigate the free-rider problem in Section 4.2. When the nodes are non-colluding and $p_w = 0$, there exists an equilibrium with non-negligible probability of failure, where the nodes trust each other to send sufficiently many clues to the contract. By colluding, rational nodes can post exactly k clues after each query, thus obviating the punishment p_w for non-responsive nodes when security is satisfied. This minimizes the number of clues posted to the contract. Analyzing collusions among subsets of nodes can also shed light on games, where multiple nodes are controlled by the same entity.

Acknowledgments. This work was supported by NSF, ONR, the Simons Foundation, NTT Research, and a grant from Ripple. Additional support was provided by the Stanford Center for Blockchain Research.

References

1. Bennet Yee, Dawn Song, Patrick McCorry, and Chris Buckland. Shades of finality and layer 2 scaling. *arXiv:2201.07920*, 2022.
2. Lewis Gudgeon, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry, and Arthur Gervais. Sok: Layer-two blockchain protocols. In *Financial Cryptography*, volume 12059 of *Lecture Notes in Computer Science*, pages 201–226. Springer, 2020.
3. Patrick McCorry, Chris Buckland, Bennet Yee, and Dawn Song. Sok: Validating bridges as a scaling solution for blockchains. *Cryptology ePrint Archive:2021/1589*, 2021.
4. Sam Williams, Viktor Diordiiev, Lev Berman, India Raybould, and Ivan Uemlianin. Arweave: A protocol for economically sustainable information permanence. Yellow paper, 2019. https://www.arweave.org/yellow-paper.pdf.
5. Yiannis Psaras and David Dias. The interplanetary file system and the filecoin network. In *DSN (Supplements)*, page 80. IEEE, 2020.
6. Andrew Miller, Ari Juels, Elaine Shi, Bryan Parno, and Jonathan Katz. Permacoin: Repurposing bitcoin work for data preservation. In *IEEE Symposium on Security and Privacy*, pages 475–490. IEEE Computer Society, 2014.
7. StarkEx v4. https://docs.starkware.co/starkex-v4/.
8. zkPorter: a breakthrough in L2 scaling. https://blog.matter-labs.io/zkporter-a-breakthrough-in-l2-scaling-ed5e48842fbf, 2021.
9. Eigenlayer. https://www.layrlabs.com/.
10. Aditi Sriram and John Adler. The Ethereum Off-Chain Data Availability Landscape. https://blog.celestia.org/ethereum-off-chain-data-availability-landscape/, 2022.
11. Aditi Sriram, John Adler, and Mustafa Al-Bassam. Quantum gravity bridge: Secure off-chain data availability for ethereum l2s with celestia. https://blog.celestia.org/celestiums/, 2022.
12. Mustafa Al-Bassam, Alberto Sonnino, Vitalik Buterin, and Ismail Khoffi. Fraud and data availability proofs: Detecting invalid blocks in light clients. In *Financial Cryptography (2)*, volume 12675 of *Lecture Notes in Computer Science*, pages 279–298. Springer, 2021.
13. Mingchao Yu, Saeid Sahraei, Songze Li, Salman Avestimehr, Sreeram Kannan, and Pramod Viswanath. Coded merkle tree: Solving data availability attacks in blockchains. In *Financial Cryptography*, volume 12059 of *Lecture Notes in Computer Science*, pages 114–134. Springer, 2020.
14. Vitalik Buterin. 2d data availability with kate commitments. https://ethresear.ch/t/2d-data-availability-with-kate-commitments/8081, 2020.
15. Dankrad Feist. New sharding design with tight beacon and shard block integration. https://notes.ethereum.org/@dankrad/new_sharding, 2022.
16. Mustafa Al-Bassam. Lazyledger: A distributed data availability ledger with client-side smart contracts. *arXiv:1905.09274*, 2019.
17. Polygon. Avail - the data availability blockchain. https://github.com/maticnetwork/data-availability, 2021.

18. Vitalik Buterin. Proto-danksharding faq. https://notes.ethereum.org/@vbuterin/proto_danksharding_faq#If-data-is-deleted-after-30-days-how-would-users-access-older-blobs, 2022.
19. Kamilla Nazirkhanova, Joachim Neu, and David Tse. Information dispersal with provable retrievability for rollups. *arXiv:2111.12323*, 2021. To appear in ACM Advances in Financial Technologies - AFT 2022.
20. Justin Drake. Starkex validium ransom attack. https://notes.ethereum.org/DD7GyItYQO2d0ax_X-UbWg?view, 2020.
21. Dankrad Feist. Proofs of custody, 2021.
22. Steve Ellis, Ari Juels, and Sergey Nazarov. Chainlink a decentralized oracle network. Whitepaper, 2017. https://research.chain.link/whitepaper-v1.pdf.
23. Lorenz Breidenbach, Christian Cachin, Benedict Chan, Alex Coventry, Steve Ellis, Ari Juels, Farinaz Koushanfar, Andrew Miller, Brendan Magauran, Daniel Moroz, Sergey Nazarov, Alexandru Topliceanu, Florian Tramer, and Fan Zhang. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. Whitepaper, 2021. https://research.chain.link/whitepaper-v2.pdf.
24. Ethereum historical data. https://www.investing.com/crypto/ethereum/historical-data. Accessed: 2022-07-15.
25. Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 177–194. Springer, 2010.
26. Vitalik Buterin. On-chain scaling to potentially $\tilde{5}00$ tx/sec through mass tx validation. https://ethresear.ch/t/on-chain-scaling-to-potentially-500-tx-sec-through-mass-tx-validation/3477, 2018.
27. Ethereum average gas price. https://ycharts.com/indicators/ethereum_average_gas_price. Accessed: 2022-07-15.
28. Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. *arXiv:1710.09437*, 2019.
29. Steve Anderrson. ETH 2.0 crosses 300,000 validators, Ether deposits worth 28.9B already locked. https://www.thecoinrepublic.com/2022/03/05/eth-2-0-crosses-300000-validators-ether-deposits-worth-28-9b-already-locked/, 2022.
30. Ben Fisch. Poreps: Proofs of space on useful data. *Cryptology ePrint Archive:2018/678*, 2018.

A Proofs of Lemma 1, and Theorems 3 and 5 for risk-neutral nodes

Proof of Lemma 1 for risk-neutral nodes and clients. Consider a game played among the client and the DAC nodes. At the beginning of slot 2 the client $\mathcal{V}$ either received k or more valid clues over the network from the nodes, or it did not.

- Let $Q_{\geq k}$ denote the event that $\mathcal{V}$ takes the action $\mathcal{S}_{\mathsf{q}}$, *i.e.*, sends query to the contract, by slot 2 even if it receives k or more valid clues over the network by the end of slot 3.
- Let $Q_{<k}$ denote the event that $\mathcal{V}$ takes the action $\mathcal{S}_{\mathsf{q}}$ by slot 2 if it does not receive k or more valid clues over the network by the end of slot 3.
- Let R denote the event that k or more nodes take the action $\mathcal{S}_{\mathsf{r}}$, *i.e.* send clues to the client over the network, by slot 3.

– Let P denote the event k or more nodes take the action $\mathcal{S}_{\mathsf{p}}$, *i.e.*, post clues to the contract, at slot $t+1 \leq 3$ if a query is received by the contract by the end of some slot $t \leq 2$.

Given the events above, we can summarize the $\mathcal{V}$'s payoff by the following table:

	$Q_{\geq k} \wedge Q_{<k}$	$Q_{\geq k} \wedge \overline{Q}_{<k}$	$\overline{Q}_{\geq k} \wedge Q_{<k}$	$\overline{Q}_{\geq k} \wedge \overline{Q}_{<k}$
$R \wedge P$	$p_f - p_c$	$p_f - p_c$	p_f	p_f
$\overline{R} \wedge P$	$p_f - p_c$	0	$p_f - p_c$	0
$R \wedge \overline{P}$	$p_f + p_{\text{comp}} - p_c$	$p_f + p_{\text{comp}} - p_c$	p_f	p_f
$\overline{R} \wedge \overline{P}$	$p_{\text{comp}} - p_c$	0	$p_{\text{comp}} - p_c$	0

Table 3: Payoff of a risk-neutral client

Towards contradiction, suppose there exists a Nash equilibrium, where the probability $\Pr[\overline{R} \wedge \overline{P}]$ is non-zero. If $\Pr[\overline{R} \wedge \overline{P}] > 0$, then the client never takes the actions $Q_{\geq k} \wedge \overline{Q}_{<k}$ and $\overline{Q}_{\geq k} \wedge \overline{Q}_{<k}$ with positive probability in the equilibrium, as it can increase its expected payoff by reducing the probability of $Q_{\geq k} \wedge \overline{Q}_{<k}$ in favor of $Q_{\geq k} \wedge Q_{<k}$, and by reducing the probability of $\overline{Q}_{\geq k} \wedge \overline{Q}_{<k}$ in favor of $\overline{Q}_{\geq k} \wedge Q_{<k}$. Hence, in the equilibrium, either $\mathcal{V}$ receives k or more valid clues over the network by the end of slot 3, *i.e.*, the event R happens, or $\mathcal{V}$ sends a query to the contract by slot 2. In the latter case, $\mathcal{V}$'s query is received by the contract by the end of slot 2.

If a valid query appears in the contract by the end of some slot t, and the node $\mathcal{P}_i$, $i \in [N-f]$, sends its clue to the contract at slot $t+1$, its payoff becomes $-p_w$. On the other hand, if a valid query is received by the contract by the end of some slot t, and $\mathcal{P}_i$ does not send its clue to the contract at slot $t+1$, its payoff can at most be $-p_w - \epsilon$. Since $-p_w > -p_w - \epsilon$, in the equilibrium, if $\mathcal{V}$'s query is received by the contract by the end of slot 2, $\mathcal{P}_i$ sends its clue to the contract by slot 3, which is then observed by the client by the beginning of slot 4. As this holds for all nodes $\mathcal{P}_i$, $i \in [N-f]$, if a query is received by the contract by the end of slot 2, all non-adversarial nodes send their clues to the contract by slot 3, *i.e.*, the event P happens. However, this implies $\Pr[R \vee P] = 1$, and $\Pr[\overline{R} \wedge \overline{P}] = 0$, which is a contradiction. Consequently, 4-security is satisfied with overwhelming probability in all Nash equilibria. □

Proof of Theorem 3 for risk-neutral nodes. Suppose a query is received by the contract at some slot $t \in \mathbb{N}$. Consider the adversary $\mathcal{A}$ that makes every adversarial node take the action $\neg\mathcal{S}_{\mathsf{p}}$ (not post clues to the contract) at all slots. Suppose there exists a Nash equilibrium of this subgame, where each node $\mathcal{P}_i$, $i \in [N-f]$, independently decides to take the action $\mathcal{S}_{\mathsf{p}}$ at slot $t+1$, with probability $r^* \in [0,1)$, and to take the action $\neg\mathcal{S}_{\mathsf{p}}$ at slot $t+1$, with probability

$1-r^*$. Then, in the equilibrium, the expected payoff of each node $\mathcal{P}_i$, $i \in [N-f]$, becomes

$$r^*(-p_w + \mathbb{E}[f_i(\mathbf{X})|X_i = 1]) + (1 - r^*)\mathbb{E}[f_i(\mathbf{X})|X_i = 0]$$

Here, $\mathbb{E}[f_i(\mathbf{X})|X_i = 1]$ and $\mathbb{E}[f_i(\mathbf{X})|X_i = 0]$ denote the expected payoff of $\mathcal{P}_i$ given all other nodes' actions and conditioned on the fact that $\mathcal{P}_i$ takes the action $\mathcal{S}_\mathsf{p}$ and $\neg\mathcal{S}_\mathsf{p}$ at slot $t+1$ respectively. As the slashing function is symmetric and offers no rewards, there exist $e_0, e_1 \in [-p_s, 0]$ such that $\mathbb{E}[f_i(\mathbf{X})|X_i = 1] = \mathbb{E}[f_j(\mathbf{X})|X_j = 1] = e_1$, and $\mathbb{E}[f_i(\mathbf{X})|X_i = 0] = \mathbb{E}[f_j(\mathbf{X})|X_j = 0] = e_0$ for all $i, j \in [N-f]$.

For the action profile described above to be a Nash equilibrium, it must be the case that

$$r^*(-p_w + e_1) + (1 - r^*)e_0 \geq r(-p_w + e_1) + (1 - r)e_0$$

for all $r \in [0, 1]$. For each $i \in [N-f]$, the inequality above is satisfied by

- $r^* = 1$, if $-p_w + e_1 > e_0$,
- $r^* = 0$, if $-p_w + e_1 < e_0$,
- any $r^* \in [0, 1]$, if $-p_w + e_1 = e_0$.

Let $r^*_{-i} \in [0, 1]$ denote the probability that among the $N - f - 1$ non-adversarial nodes other than $\mathcal{P}_i$, at most $k-1$ nodes take the action $\mathcal{S}_\mathsf{p}$ (send clues to the contract) at slot $t+1$ in the equilibrium. As the slashing function is symmetric, $r^*_{-i} = r^*_{-j} = r^*_{-1}$ for all $i, j \in [N-f]$, and

$$-(1 - r^*_{-1})B - r^*_{-1}p_s \leq e_0 \leq 0$$
$$-p_w - p_s \leq -p_w + e_1 \leq -p_w,$$

where $B < p_w$. This implies a value $r^*_{-1} \in (0, 1]$ such that $-p_w + e_1 \leq e_0$, and $r^* < 1$.

If $-p_w + e_1 < e_0$, then $r^* = 0$. On the other hand, if $-p_w + e_1 = e_0$, as $B < p_w$, it must be the case that $r^*_{-1} \in (0, 1]$, which implies $r^* \in [0, 1)$. Consequently, if $B < p_w$, there indeed exists a $(0, 0)$-adversary $\mathcal{A}$ and a Nash equilibrium, where each non-adversarial node independently decides to not post its clue to the contract at the slot after a query is received by the contract, with probability $r^* > 0$.

Finally, suppose there exists a subgame perfect equilibrium of the dynamic game, where none of the nodes send a valid clue to the client $\mathcal{V}$ over the network by slot 3, and $\mathcal{V}$ sends its query to the contract by slot 2. Once the query appears in the contract, each nodes chooses to withhold its clue from the contract in the next slot with probability r^*.

As shown above, if there is a query in the contract, none of the nodes can increase its expected payoff by deviating from the specified action given the other nodes' actions. Similarly, $\mathcal{V}$ cannot increase its expected payoff by not sending a query to the contract, when none of the nodes sends a valid clue over the

network by slot 3. Finally, none of the nodes can increase its expected payoff by sending a valid clue to $\mathcal{V}$ over the network by slot 2, since this does not affect $\mathcal{V}$'s behavior. This is because $k > 1$ and all other nodes refuse to send their clues over the network. Hence, the claimed action profile indeed constitutes a subgame perfect equilibrium. However, in this case, there are less than k valid clues in the contract by slot 4 with probability at least $(1-r^*)^{N-f} > 0$. Thus, there exists a $(0,0)$-adversary and a subgame perfect equilibrium, where 4-security is violated with positive probability. □

Proof of Theorem 5. We first show that given the slashing function of Section 3, $q_{\mathrm{v}}^{\mathcal{A}} \leq q^*$ for any p_0-adversary $\mathcal{A}$. Let F denote the event that there are less than k valid clues in the contract at slot $t+1$. Towards contradiction, suppose there exists a Nash equilibrium such that $\Pr[F] > q^*$ in the equilibrium. Then, with probability greater than q^*, at least $N-f-k+1$ non-adversarial nodes take the action $\neg\mathcal{S}_{\mathsf{p}}$, *i.e.*, do not post their clues to the contract, at slot $t+1$ in the equilibrium. Let U_i^* denote the realization of the node $\mathcal{P}_i$'s payoff, $i \in [N-f]$, in the equilibrium. In the case of event F, at least $N-f-k+1$ nodes incur a penalty of $-p_s$, and given the total bribe p_0, the total expected payoff of the non-adversarial nodes can at most be

$$\sum_{i=1}^{N-f} \mathbb{E}[U_i^* | F] \leq p_0 - (k-1)p_w - (N-f-k+1)p_s$$

in the equilibrium. Conversely, if the event F does not happen, the total expected payoff can at most be

$$\sum_{i=1}^{N-f} \mathbb{E}[U_i^* | \overline{F}] \leq p_0 - (N-f)p_w$$

in the equilibrium. Using the two inequalities and the assumed lower bound on $\Pr[F]$, we derive the following upper bound on the total expected payoff of the non-adversarial nodes in the equilibrium:

$$\begin{aligned}\sum_{i=1}^{N-f} \mathbb{E}[U_i^*] &\leq p_0 - (N-f)p_w - \Pr[F](N-f-k+1)(p_s - p_w) \\ &< p_0 - (N-f)p_w - q^*(N-f-k+1)(p_s - p_w) \\ &= -(N-f)p_w\end{aligned}$$

The inequality above implies the existence of at least one node $\mathcal{P}_{i^*}$, $i^* \in [N-f]$, such that $\mathbb{E}[U_{i^*}^*] < -p_w$. However, sending a valid clue to the contract at slot $t+1$ gives a payoff of $-p_w$ for $\mathcal{P}_{i^*}$ regardless of the actions of all other nodes, implying that there exists an action that strictly dominates the one taken by $\mathcal{P}_{i^*}$ in the equilibrium. Thus, we have reached a contradiction.

***

We next construct a p_0-adversary $\mathcal{A}$ such that for any compliant slashing function, there exists a Nash equilibrium, where $q_{\mathsf{v}}^{\mathcal{A}} \geq q^*$. Let X_i denote the indicator random variable for the event that $\mathcal{P}_i$ takes the action $\mathcal{S}_{\mathsf{p}}$, *i.e.* sends its query to the contract, at slot $t+1$. Then, any adversary $\mathcal{A}$ can be characterized as follows:

- Bribes offered to the non-adversarial nodes $\mathcal{P}_i$, $i \in [N-f]$: p_b^i, $i \in [N-f]$. Bribes must satisfy the equation $\sum_{i\in[N-f]} p_b^i \leq p_0$. A non-adversarial node is called *corrupt* if it accepts the bribe.
- The probability distribution over the actions adopted by the adversarial and corrupt nodes. For each Q, $\{N-f+1,\ldots,N\} \subseteq Q \subseteq [N]$, representing the set of adversarial and corrupt nodes, $\mathcal{A}$ proposes the following action profile: $\Pr[X_i = x_i \in \{0,1\}, i \in Q] = q_{(x_i, i\in Q)}$.

Similarly, given any adversary $\mathcal{A}$, each Nash equilibrium can be described by the following variables: $\{\tilde{r}_i^*\}_{i\in[N-f]}$ and $\{r_i^*\}_{i\in[N-f]}$. Here, $\tilde{r}_i^*$ denotes the probability that $\mathcal{P}_i$ accepts the bribe in the equilibrium, whereas r_i^* denotes the probability that $\mathcal{P}_i$ takes the action $\mathcal{S}_{\mathsf{p}}$ in the event that it does not accept the bribe. We allow r_i^* to be undefined (shown as $-$) if $\tilde{r}_i^* = 1$, *i.e.*, if $\mathcal{P}_i$ accepts the adversary's bribe, in which case it will take the action $\mathcal{S}_{\mathsf{p}}$ as dictated by the adversary. In the equilibrium, each node $\mathcal{P}_j$, $j \in [N-f]$, chooses the values $\tilde{r}_j^*$ and r_j^* to maximize its expected payoff given the adversary $\mathcal{A}$, and the other nodes' equilibrium actions, *i.e.*, $\{\tilde{r}_i^*\}_{i\in[N-f]/\{j\}}$ and $\{r_i^*\}_{i\in[N-f]/\{j\}}$.

Consider the p_0-adversary $\mathcal{A}$ that offers a bribe of $p_b^i = p_0/(N-f-k+1)$ to the nodes $\mathcal{P}_i$, $i \in [N-f-k+1]$, and for the set $Q \subseteq [N-f-k+1] \cup \{N-f+1,\ldots,N\}$, specifies

$$\Pr[X_i = 0, i \in Q] = \frac{p_0}{(N-f-k+1)(p_s - p_w)},$$
$$\Pr[X_i = 1, i \in Q] = 1 - \frac{p_0}{(N-f-k+1)(p_s - p_w)}.$$

We will show that given $\mathcal{A}$, for any compliant slashing function, there exists a Nash equilibrium such that $q_{\mathsf{v}}^{\mathcal{A}} \geq q^*$. In the equilibrium, for each $i \in [N-f-k+1]$, either $(\tilde{r}_i^*, r_i^*) = (1,-)$ or $(\tilde{r}_i^*, r_i^*) = (0,0)$, and for each $i \in [N-f]/[N-f-k+1]$, $(\tilde{r}_i^*, r_i^*) = (0,0)$. In other words, nodes that are offered bribes either accept the bribe and become corrupted, or do not post their clues to the contract in the equilibrium, whereas those that are not offered bribes post their clues to the contract.

Since the considered slashing functions are compliant, they offer no rewards and satisfy B minimal punishment for some $B \geq p_w$ by Theorem 3. Thus, for all $i \in [N]$, it must be the case that $f_i(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in \{0,1\}^N$, and $f_i(\mathbf{x}) \leq -p_w$ for all $\mathbf{x} \in \{0,1\}^N$ such that $x_i = 0$. Then, if a node $\mathcal{P}_i$, $i \in [N-f-k+1]$, rejects the bribe, its expected payoff becomes

$$(-p_w + \mathbb{E}[f_i(\mathbf{X})|X_i = 1])r_i^* + \mathbb{E}[f_i(\mathbf{X})|X_i = 0](1 - r_i^*) \leq \mathbb{E}[f_i(\mathbf{X})|X_i = 1]r_i^* - p_w.$$

Here, $\mathbb{E}[f_i(\mathbf{X})|X_i = 1]$ and $\mathbb{E}[f_i(\mathbf{X})|X_i = 0]$ denote the expected payoff of $\mathcal{P}_i$ given all other nodes' actions in the claimed equilibrium and conditioned on

the fact that $\mathcal{P}_i$ takes the actions $\mathcal{S}_\mathsf{p}$ and $\neg\mathcal{S}_\mathsf{p}$ at slot $t+1$ respectively. As the slashing function is symmetric, for any $i, j \in [N-f-k+1]$, $\mathbb{E}[f_i(\mathbf{X})|X_i = 1] = \mathbb{E}[f_j(\mathbf{X})|X_j = 1] = e_1 \leq 0$ and $\mathbb{E}[f_i(\mathbf{X})|X_i = 0] = \mathbb{E}[f_j(\mathbf{X})|X_j = 0] = e_0 \leq -B \leq -p_w$.

As each node $\mathcal{P}_i$, $i \in [N-f-k+1]$, maximizes its payoff given all other nodes' payoffs in the equilibrium, if $\mathcal{P}_i$ rejects the bribe, it must be the case that

- $r_i^* = 1$, if $-p_w + e_1 > e_0$. In this case, $\mathcal{P}_i$'s expected payoff becomes $-p_w + e_1 \leq -p_w$.
- $r_i^* = 0$, if $-p_w + e_1 < e_0$.
- r_i^* can be any value in $[0,1]$, if $-p_w + e_1 = e_0$. In this case, $\mathcal{P}_i$'s expected payoff becomes $e_0 \leq -B \leq -p_w$

On the other hand, if $\mathcal{P}_i$ accepts the bribe, its expected payoff becomes at least

$$\begin{aligned}
&\frac{p_0}{N-f-k+1} - p_s \frac{p_0}{(N-f-k+1)(p_s-p_w)} \\
&+ (-p_w + \mathbb{E}[f_i(\mathbf{X})|X_i = 1]) \left(1 - \frac{p_0}{(N-f-k+1)(p_s-p_w)}\right) \\
&= -p_w + e_1 \left(1 - \frac{p_0}{(N-f-k+1)(p_s-p_w)}\right).
\end{aligned}$$

Since $e_1 \leq 0$, if $-p_w + e_1 \geq e_0$, then the expected payoff of node $\mathcal{P}_i$ when it accepts the bribe is at least as large as its expected payoff when it rejects the bribe. Hence, if $-p_w + e_1 \geq e_0$, for the nodes $\mathcal{P}_i$, $i \in [N-f-k+1]$, there does not exist any action that strictly dominates $(\tilde{r}_i^*, r_i^*) = (1, -)$. In this case, the action profile specified by the adversary constitutes a Nash equilibrium, and $q_\mathrm{v}^\mathcal{A}$ becomes at least

$$\Pr[X_i = 0, i \in Q] = \frac{p_0}{(N-f-k+1)(p_s-p_w)} = q^*,$$

where $Q = [N-f-k+1] \cup \{N-f+1, \ldots, N\}$. Conversely, if $-p_w + e_1 < e_0$, then either the nodes $\mathcal{P}_i$, $i \in [N-f-k+1]$, all reject the bribe, and set $r_i^* = 0$, which implies $q_\mathrm{v}^\mathcal{A} = 1$, or they all accept the bribe, which implies $q_\mathrm{v}^\mathcal{A} \geq q^*$. In both cases, $q_\mathrm{v}^\mathcal{A} \geq q^*$, thus concluding the proof. □

B Proofs of Lemma 1, and Theorems 3, and 4 for risk-averse nodes

Proof of Lemma 1 for risk-averse nodes and clients. Consider a game played among the client and the DAC nodes. At the beginning of slot 2, the client $\mathcal{V}$ either received k or more valid clues over the network from the nodes, or it did not. Recall the definitions of the events $Q_{\geq k}$, $Q_{<k}$, R and P from the proof of Lemma 1 for risk-neutral nodes in Appendix A. Given these events, we can summarize the $\mathcal{V}$'s payoff by the following table:

Towards contradiction, suppose there exists a Nash equilibrium, where the probability $\Pr[\overline{R} \wedge \overline{P}]$ is non-zero. If $\Pr[\overline{R} \wedge \overline{P}] > 0$, then the client never takes the

	$Q_{\geq k} \wedge Q_{<k}$	$Q_{\geq k} \wedge \overline{Q}_{<k}$	$\overline{Q}_{\geq k} \wedge Q_{<k}$	$\overline{Q}_{\geq k} \wedge \overline{Q}_{<k}$
$R \wedge P$	$(p_f - p_c)^\nu$	$(p_f - p_c)^\nu$	$(p_f)^\nu$	$(p_f)^\nu$
$\overline{R} \wedge P$	$(p_f - p_c)^\nu$	0	$(p_f - p_c)^\nu$	0
$R \wedge \overline{P}$	$(p_f + p_{\text{comp}} - p_c)^\nu$	$(p_f + p_{\text{comp}} - p_c)^\nu$	$(p_f)^\nu$	$(p_f)^\nu$
$\overline{R} \wedge \overline{P}$	$(p_{\text{comp}} - p_c)^\nu$	0	$(p_{\text{comp}} - p_c)^\nu$	0

Table 4: Utility of a risk-averse client

actions $Q_{\geq k} \wedge \overline{Q}_{<k}$ and $\overline{Q}_{\geq k} \wedge \overline{Q}_{<k}$ with positive probability in the equilibrium, as it can increase its expected payoff by reducing the probability of $Q_{\geq k} \wedge \overline{Q}_{<k}$ in favor of $Q_{\geq k} \wedge Q_{<k}$, and by reducing the probability of $\overline{Q}_{\geq k} \wedge \overline{Q}_{<k}$ in favor of $\overline{Q}_{\geq k} \wedge Q_{<k}$. Hence, in the equilibrium, either $\mathcal{V}$ receives k or more valid clues over the network by the end of slot 3, *i.e.*, the event R happens, or $\mathcal{V}$ sends a query to the contract by slot 2. In the latter case, $\mathcal{V}$'s query is received by the contract by the end of slot 2.

If a valid query appears in the contract by the end of some slot t, and the node $\mathcal{P}_i$, $i \in [N-f]$, sends its clue to the contract at slot $t+1$, its payoff becomes $(C - p_w)^\nu$. On the other hand, if a valid query is received by the contract by the end of some slot t, and $\mathcal{P}_i$ does not send its clue to the contract at slot $t+1$, its payoff can at most be $(C - p_w - \epsilon)^\nu$. Since $(C - p_w)^\nu > (C - p_w - \epsilon)^\nu$, in the equilibrium, if $\mathcal{V}$'s query is received by the contract by the end of slot 2, $\mathcal{P}_i$ sends its clue to the contract by slot 3, which is then observed by the client by the beginning of slot 4. As this holds for all nodes $\mathcal{P}_i$, $i \in [N-f]$, if a query is received by the contract by the end of slot 2, all non-adversarial nodes send their clues to the contract by slot 3, *i.e.*, the event P happens. However, this implies $\Pr[R \vee P] = 1$, and $\Pr[\overline{R} \wedge \overline{P}] = 0$, which is a contradiction. Consequently, 4-security is satisfied with overwhelming probability in all Nash equilibria. □

Theorem 2 follows from Lemma 1, which shows security under no attack (A3) for the slashing function of Section 3.

Proof of Theorem 3 for risk-averse nodes. Suppose a query is received by the contract at some slot $t \in \mathbb{N}$. Consider the adversary $\mathcal{A}$ that makes every adversarial node take the action $\neg\mathcal{S}_\mathsf{p}$ (not post clues to the contract) at all slots. Suppose there exists a Nash equilibrium of this subgame, where each node $\mathcal{P}_i$, $i \in [N-f]$, independently decides to take the action $\mathcal{S}_\mathsf{p}$ at slot $t+1$, with probability $r^* \in [0,1)$, and to take the action $\neg\mathcal{S}_\mathsf{p}$ at slot $t+1$, with probability $1-r^*$. Then, in the equilibrium, the expected payoff of each node $\mathcal{P}_i$, $i \in [N-f]$, becomes

$$r^*\mathbb{E}[(C - p_w + f_i(\mathbf{X}))^\nu | X_i = 1] + (1 - r^*)\mathbb{E}[(C + f_i(\mathbf{X}))^\nu | X_i = 0],$$

where $C = p_s + p_w$. Here, $\mathbb{E}[(C - p_w + f_i(\mathbf{X}))^\nu | X_i = 1]$ and $\mathbb{E}[(C + f_i(\mathbf{X}))^\nu | X_i = 0]$ denote the expected payoff of $\mathcal{P}_i$ given all other nodes' actions and conditioned on the fact that $\mathcal{P}_i$ takes the action $\mathcal{S}_\mathsf{p}$ and $\neg\mathcal{S}_\mathsf{p}$ at slot $t+1$ respectively. As the slashing function is symmetric and offers no rewards, there exist $e_0, e_1 \in [-p_s, 0]$

such that $\mathbb{E}[(C-p_w+f_i(\mathbf{X}))^\nu|X_i=1]=\mathbb{E}[(C-p_w+f_j(\mathbf{X}))^\nu|X_j=1]=e_1$, and $\mathbb{E}[(C+f_i(\mathbf{X}))^\nu|X_i=0]=\mathbb{E}[(C+f_j(\mathbf{X}))^\nu|X_j=0]=e_0$ for all $i,j\in[N-f]$.

For the action profile described above to be a Nash equilibrium, it must be the case that

$$r^*e_1+(1-r^*)e_0\geq re_1+(1-r)e_0$$

for all $r\in[0,1]$. For each $i\in[N-f]$, the inequality above is satisfied by

- $r^*=1$, if $e_1>e_0$,
- $r^*=0$, if $e_1<e_0$,
- any $r^*\in[0,1]$, if $e_1=e_0$.

Let $r^*_{-i}\in[0,1]$ denote the probability that among the $N-f-1$ non-adversarial nodes other than $\mathcal{P}_i$, at most $k-1$ nodes take the action $\mathcal{S}_\mathsf{p}$ (send clues to the contract) at slot $t+1$ in the equilibrium. As the slashing function is compliant, $r^*_{-i}=r^*_{-j}=r^*_{-1}$ for all $i,j\in[N-f]$, and

$$(1-r^*_{-1})(C-B)^\nu-r^*_{-1}(C-p_s)^\nu\leq e_0\leq(C)^\nu$$
$$(C-p_w-p_s)^\nu\leq e_1\leq(C-p_w)^\nu,$$

where $B<p_w$. This implies a value $r^*_{-1}\in(0,1]$ such that $e_1\leq e_0$, and $r^*<1$.

If $e_1<e_0$, then $r^*=0$. On the other hand, if $e_1=e_0$, as $B<p_w$, it must be the case that $r^*_{-1}\in(0,1]$, which implies $r^*\in[0,1)$. Consequently, if $B<p_w$, there indeed exists a $(0,0)$-adversary $\mathcal{A}$ and a Nash equilibrium, where each non-adversarial node independently decides to not post its clue to the contract at the slot after a query is received by the contract, with probability $r^*>0$.

Finally, suppose there exists a subgame perfect equilibrium of the dynamic game, where none of the nodes send a valid clue to the client $\mathcal{V}$ over the network by slot 3, and $\mathcal{V}$ sends its query to the contract by slot 2. Once the query appears in the contract, each nodes chooses to withhold its clue from the contract in the next slot with probability r^*.

As shown above, if there is a query in the contract, none of the nodes can increase its expected payoff by deviating from the specified action given the other nodes' actions. Similarly, $\mathcal{V}$ cannot increase its expected payoff by not sending a query to the contract, when none of the nodes sends a valid clue over the network by slot 3. Finally, none of the nodes can increase its expected payoff by sending a valid clue to $\mathcal{V}$ over the network by slot 2, since this does not affect $\mathcal{V}$'s behavior. This is because $k>1$ and all other nodes refuse to send their clues over the network. Hence, the claimed action profile indeed constitutes a subgame perfect equilibrium. However, in this case, there are less than k valid clues in the contract by slot 4 with probability at least $(1-r^*)^{N-f}>0$. Thus, there exists a $(0,0)$-adversary and a subgame perfect equilibrium, where 4-security is violated with positive probability. □

Proof of Theorem 4. Consider the subgame, where a query is received by the contract of Section 3 at some slot t. Let $q_\mathsf{v}^\mathcal{A}$ denote the probability that given

an adversary $\mathcal{A}$, there are less than k valid clues in the contract at slot $t+1$ in the Nash equilibrium with the largest probability of failure. We first show that there exists a function $q^*(x) : [0, (N-f-k+1)(p_s-p_w)) \to (0,1)$ such that given the slashing function of Section 3, for any p_0-adversary $\mathcal{A}$, $0 \leq p_0 < (N-f-k+1)(p_s-p_w)$, it holds that $q_{\mathrm{v}}^{\mathcal{A}} \leq q^*(p_0)$. By Remark 2, if $p_0 \geq (N-f-k+1)(p_s-p_w)$, $q_{\mathrm{v}}^{\mathcal{A}} = 1$ for any compliant slashing function, including the function of Section 3.

Given a p_0-adversary with $p_0 < (N-f-k+1)(p_s-p_w)$, consider the Nash equilibrium with the maximum failure probability. Let q_i denote the probability that in the equilibrium, $\mathcal{P}_i$, $i \in [N-f]$, does not take the action $\mathcal{S}_{\mathsf{p}}$, *i.e.*, does not post its clue to the contract, and there are less than k valid clues in the contract at slot $t+1$. Let p_b^i denote the bribe offered to $\mathcal{P}_i$ by $\mathcal{A}$. As $\mathcal{P}_i$ maximizes its utility given all other nodes' actions in the equilibrium, any $q_i \in [0,1]$ satisfies the following inequality; otherwise, $\mathcal{P}_i$ can increase its utility by posting its clue to the contract at slot $t+1$:

$$(C-p_w)^\nu \leq q_i(p_b^i + C - p_s)^\nu + (1-q_i)(p_b^i + C - p_w)^\nu,$$

which further implies

$$q_i \leq \frac{(p_b^i + C - p_w)^\nu - (C-p_w)^\nu}{(p_b^i + C - p_w)^\nu - (p_b^i + C - p_s)^\nu},$$

where $C = p_s + p_w$.

Let $\mathcal{G}$ denote the set of subsets of $[N-f]$ with $N-f-k+1$ elements. Let E_G, $G \in \mathcal{G}$, denote the event that the nodes in G do not take the action $\mathcal{S}_{\mathsf{p}}$ at slot $t+1$. By definition of q_i,

$$q_i = \Pr[\cup_{G \in \mathcal{G}: i \in G} E_G] = \Pr[\cup_{G \in G_i} E_G],$$

where $G_i = \{G \in \mathcal{G} : i \in G\}$. Similarly, by definition of $q_{\mathrm{v}}^{\mathcal{A}}$,

$$q_{\mathrm{v}}^{\mathcal{A}} = \Pr[\cup_{G \in \mathcal{G}} E_G].$$

Hence, the function $q^*(p_0)$ is upper bounded by the solution $\tilde{q}^*(p_0)$ to the following optimization problem:

$$\begin{aligned}
&\max_{G \in \mathcal{G}} \ \Pr[\cup_{G \in \mathcal{G}} E_G] \\
&\text{s.t. } \Pr[\cup_{G \in G_i} E_G] = q_i \leq \frac{(p_b^i + C - p_w)^\nu - (C-p_w)^\nu}{(p_b^i + C - p_w)^\nu - (p_b^i + C - p_s)^\nu} \quad \forall i \in [N-f] \\
&\quad \sum_{i=1}^{N-f} p_b^i \leq p_0 \\
&\quad p_b^i \geq 0 \quad \forall i \in [N-f]
\end{aligned}$$

Let $\{\tilde{E}_G : G \in \mathcal{G}\}$, denote one set of events for which the value of $\Pr[\cup_{G \in \mathcal{G}} E_G]$ is maximized. Let $\tilde{p}_b^i$ and $\tilde{q}_i$, $i \in [N-f]$, denote the optimal values of the parameters p_b^i and q_i, $i \in [N-f]$, associated with this set of events.

We next construct a p_0-adversary $\mathcal{A}$ such that for any compliant slashing function, there exists a Nash equilibrium, where $q_{\mathsf{v}}^{\mathcal{A}} \geq \tilde{q}^*(p_0) \geq q^*(p_0)$. This would show that the slashing function of Section 3 is security optimal:

- Bribes offered to the nodes $\mathcal{P}_i$, $i \in [N-f]$ are given by $\tilde{p}_b^i$.
- The probability distribution over the actions adopted by the adversarial and corrupt nodes is determined by the events $\tilde{E}_G$, which satisfy the equation $\Pr[\cup_{G \in G_i} \tilde{E}_G] = \tilde{q}_i$.

By definition of the optimization problem, it holds that

$$\tilde{q}_i \leq \frac{(p_b^i + C - p_w)^\nu - (C - p_w)^\nu}{(p_b^i + C - p_w)^\nu - (p_b^i + C - p_s)^\nu} \quad \forall i \in [N-f] \tag{2}$$

Recall that given any adversary $\mathcal{A}$, each Nash equilibrium can be described by the following variables: $\{\tilde{r}_i^*\}_{i \in [N-f]}$ and $\{r_i^*\}_{i \in [N-f]}$. Here, $\tilde{r}_i^*$ denotes the probability that $\mathcal{P}_i$ accepts the bribe in the equilibrium, whereas r_i^* denotes the probability that $\mathcal{P}_i$ takes the action $\mathcal{S}_{\mathsf{p}}$, *i.e.*, posts a valid clue to the contract, at slot $t+1$ in the event that it does not accept the bribe. We allow r_i^* to be undefined if $\mathcal{P}_i$ accepts the bribe, in which case it will take the action $\mathcal{S}_{\mathsf{p}}$ as dictated by the adversary.

If a node $\mathcal{P}_i$, $i \in [N-f]$, rejects the bribe, its expected payoff becomes

$$\mathbb{E}[(C + f_i(\mathbf{X}) - p_w)^\nu | X_i = 1] r_i^* + \mathbb{E}[(C + f_i(\mathbf{X}))^\nu | X_i = 0](1 - r_i^*)$$

Here the expectation is over the actions of the other nodes in the equilibrium.

As the slashing function satisfies symmetry, for any $i, j \in [N-f]$, $\mathbb{E}[(C + f_i(\mathbf{X}) - p_w)^\nu | X_i = 1] = \mathbb{E}[(C + f_j(\mathbf{X}) - p_w)^\nu | X_j = 1] = e_1$ and $\mathbb{E}[(C + f_i(\mathbf{X}))^\nu | X_i = 0] = \mathbb{E}[(C + f_j(\mathbf{X}))^\nu | X_j = 0] = e_0$. As $\mathcal{P}_i$ maximizes its utility given all other nodes' actions in the equilibrium, it must be the case that

- $r_i^* = 1$, if $e_1 > e_0$. In this case, $\mathcal{P}_i$'s expected utility becomes e_1.
- $r_i^* = 0$, if $e_1 < e_0$.
- r_i^* can be any value in $[0,1]$, if $e_1 = e_0$. In this case, $\mathcal{P}_i$'s expected utility becomes e_1.

Here, e_1 is upper bounded as shown by the following lemma:

Lemma 2. *For any $i \in [N-f]$, $p_b^i \geq 0$ and $r_i^* \in [0,1]$,*

$$\begin{aligned}&(C - p_w)^\nu + (1 - r_i^*)(\mathbb{E}[(C - p_w + f_i(\mathbf{X}) + p_b^i)^\nu | X_i = 1] - (C + p_b^i - p_w)^\nu) \\ &\geq \mathbb{E}[(C + f_i(\mathbf{X}) - p_w)^\nu | X_i = 1] = e_1\end{aligned}$$

Proof. For any fixed $a, b, c \in \mathbb{R}^+ \cup \{0\}$ such that $a \geq b \geq c$, it holds that $(a-c)^\nu - (b-c)^\nu \geq a^\nu - b^\nu$, which implies $b^\nu - (b-c)^\nu \geq a^\nu - (a-c)^\nu$. Moreover, for any compliant slashing function, $-p_s \leq f_i(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in \{0,1\}^N$ and all

$i \in [N]$. Since $C = p_s + p_w$, it holds that $C - p_w + f_i(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \{0,1\}^N$ and all $i \in [N]$. Then, for any $\mathbf{x} \in \{0,1\}^{N-f}$ and $i \in [N-f]$,

$$\begin{aligned}
&(C - p_w)^\nu - (C - p_w + f_i(\mathbf{x}))^\nu \geq (C - p_w + p_b^i)^\nu - (C - p_w + f_i(\mathbf{x}) + p_b^i)^\nu \\
\implies &(C - p_w)^\nu - ((C - p_w + p_b^i)^\nu - (C - p_w + f_i(\mathbf{x}) + p_b^i)^\nu) \geq (C - p_w + f_i(\mathbf{x}))^\nu \\
\implies &(C - p_w)^\nu - (1 - r_i^*)((C - p_w + p_b^i)^\nu - (C - p_w + f_i(\mathbf{x}) + p_b^i)^\nu) \\
&\geq (C - p_w + f_i(\mathbf{x}))^\nu,
\end{aligned}$$

since $1 - r_i^* \in [0,1]$ and $(C - p_w + p_b^i)^\nu - (C - p_w + f_i(\mathbf{x}) + p_b^i)^\nu \geq 0$. Hence, by linearity of expectation,

$$\begin{aligned}
&(C - p_w)^\nu + (1 - r_i^*)(\mathbb{E}[(C - p_w + f_i(\mathbf{X}) + p_b^i)^\nu | X_i = 1] - (C + p_b^i - p_w)^\nu) \\
&\geq \mathbb{E}[(C - p_w + f_i(\mathbf{X}))^\nu | X_i = 1].
\end{aligned}$$

□

On the other hand, if $e_1 > e_0$ and $\mathcal{P}_i$ accepts the bribe, its expected payoff becomes at least

$$\begin{aligned}
&\left(C + p_b^i - p_s\right)^\nu \tilde{q}_i + \mathbb{E}[(C - p_w + f_i(\mathbf{X}) + p_b^i)^\nu | X_i = 1](1 - \tilde{q}_i) \\
&\geq (C - p_w)^\nu + (1 - \tilde{q}_i)(\mathbb{E}[(C - p_w + f_i(\mathbf{X}) + p_b^i)^\nu | X_i = 1] - (C + p_b^i - p_w)^\nu) \\
&\geq e_1.
\end{aligned}$$

Here, the first inequality follows from the equation (2), and the last inequality follows from Lemma 2. Hence, if $e_1 \geq e_0$, the expected utility of node $\mathcal{P}_i$ when it accepts the bribe is at least as large as its expected utility when it rejects the bribe. Thus, for the nodes $\mathcal{P}_i$, $i \in [N-f]$, there does not exist any action that dominates $(\tilde{r}_i^*, r_i^*) = (1, -)$, implying that they accept the adversary's bribe when any bribe offer is made. In this case, the action profile specified by the adversary constitutes a Nash equilibrium, and in the equilibrium, $q_{\mathrm{v}}^{\mathcal{A}}$ becomes at least $\tilde{q}^*(p_0) \geq q^*(p_0)$.

Finally, if $e_1 < e_0$, then either the nodes $\mathcal{P}_i$, $i \in [N-f]$, all reject the bribe, and set $r_i^* = 0$, or they all accept the bribe. The first case happens if for the nodes $\mathcal{P}_i$, $i \in [N-f-k+1]$, there does not exist any action that dominates $(\tilde{r}_i^*, r_i^*) = (0, 0)$. Then, in the equilibrium, $q_{\mathrm{v}}^{\mathcal{A}}$ becomes 1. In the latter case, $q_{\mathrm{v}}^{\mathcal{A}} \geq q^*(p_0)$ as argued above, thus concluding the proof. □

C Probability of security failure for risk-averse nodes

Theorem 10. *For any p_0, $0 \leq p_0 < (N-f-k+1)(p_s - p_w)$, and $\nu \in (0,1]$, $q_{p_0,\nu}^*$ is the solution to the following optimization problem: Let $\mathcal{G}$ denote the set of subsets of $[N-f]$ with $N-f-k+1$ elements. Let x_G denote variables indexed*

by the sets $G \in \mathcal{G}$.

$$\begin{aligned}
&\max_{j\in[|\mathcal{G}|]} \sum_{j=1}^{|\mathcal{G}|} x_G \\
&\quad s.t. \sum_{G\in\mathcal{G}:i\in G} x_G \leq \frac{(p_b^i+p_s)^\nu-(p_s)^\nu}{(p_b^i+p_s)^\nu-(p_b^i+p_w)^\nu} \quad \forall i\in[N-f] \\
&\quad \sum_{i=1}^{N-f} p_b^i \leq p_0 \\
&\quad p_b^i \geq 0 \quad \forall i\in[N-f] \\
&\quad x_G \in [0,1] \quad \forall G\in\mathcal{G}
\end{aligned} \tag{3}$$

By Remark 2, $q^*_{p_0,\nu} = 1$ if $p_0 \geq (N-f-k+1)(p_s-p_w)$.

Proof of Theorem 10. From the proof of Theorem 4, we know that $q^*_{p_0,\nu}$ is the solution to the following optimization problem, where E_G is the event that the nodes in G do not post their clues to the contract at slot $t+1$ when a query is received by the contract at slot t:

$$\begin{aligned}
&\max_{G\in\mathcal{G}} \Pr[\cup_{G\in\mathcal{G}} E_G] \\
&\text{s.t. } \Pr[\cup_{G\in\mathcal{G}:i\in G} E_G] = q_i \leq \frac{(p_b^i+C-p_w)^\nu-(C-p_w)^\nu}{(p_b^i+C-p_w)^\nu-(p_b^i+C-p_s)^\nu} \quad \forall i\in[N-f] \\
&\quad \sum_{i=1}^{N-f} p_b^i \leq p_0 \\
&\quad p_b^i \geq 0 \quad \forall i\in[N-f]
\end{aligned} \tag{4}$$

Let $\{G^j\}_{j\in[|\mathcal{G}|]}$ denote a total order across the events $G\in\mathcal{G}$. Defining the disjoint events $\hat{E}_{G^1} := E_{G^1}$, and $\hat{E}_{G^i} := E_{G^i}/(\cup_{j\in[i-1]} E_{G^j})$ for $i = 2,\ldots,|\mathcal{G}|$, we observe that the solution of the following optimization is at least as large as the solution to the problem presented by Formula (4), *i.e.*, $q^*_{p_0,\nu}$:

$$\begin{aligned}
&\max_{j\in[|\mathcal{G}|]} \sum_{j=1}^{|\mathcal{G}|} \Pr[\hat{E}_{G^j}] \\
&\quad \text{s.t. } \sum_{G\in\mathcal{G}:i\in G} \Pr[\hat{E}_G] \leq \frac{(p_b^i+C-p_w)^\nu-(C-p_w)^\nu}{(p_b^i+C-p_w)^\nu-(p_b^i+C-p_s)^\nu} \quad \forall i\in[N-f] \\
&\quad \sum_{i=1}^{N-f} p_b^i \leq p_0 \\
&\quad p_b^i \geq 0 \quad \forall i\in[N-f] \\
&\quad \Pr[\hat{E}_G] \in [0,1] \quad \forall G\in\mathcal{G}
\end{aligned} \tag{5}$$

This is because by definition of $\hat{E}_{G^j}$, the optimal values are the same

$$\Pr[\cup_{G\in\mathcal{G}} E_G] = \Pr[\cup_{G\in\mathcal{G}} \hat{E}_G] = \sum_{j=1}^{|\mathcal{G}|} \Pr[\hat{E}_{G^j}]$$

in both formulas whereas the constraints are relaxed in Formula (5):

$$\sum_{G\in\mathcal{G}:i\in G} \Pr[\hat{E}_G] = \Pr[\cup_{G\in\mathcal{G}:i\in G} \hat{E}_G] \leq \Pr[\cup_{G\in\mathcal{G}:i\in G} E_G]$$

Note that by setting all of E_G, $G \in \mathcal{G}$, to be disjoint events, which implies $\hat{E}_G = E_G$, we can ensure that $q^*_{p_0,\nu}$ is the same as the solution to Formula (5). In this case, using $C = p_s + p_w$ and defining $x_G := \hat{E}_G$, we can re-write Formula (4) as Formula (3). □

Theorem 11. *Suppose* $p_0 < (N - f - k + 1)(p_s - p_w)$. *Then,*

$$\frac{(p_s + p_b)^\nu - (p_s)^\nu}{(p_s + p_b)^\nu - (p_w + p_b)^\nu} \leq q^*_{p_0,\nu}, \text{ and}$$
$$q^*_{p_0,\nu} \leq \frac{1}{N - f - k + 1} \min\left(\frac{p_0}{p_s - p_w}, \frac{(p_s + p_0)^\nu - (p_s)^\nu}{(p_s + p_0)^\nu - (p_w + p_0)^\nu}\right),$$

where $p_b = \frac{p_0}{N-f-k+1}$.

Proof of Theorem 11. We first prove the upper bound. Recall that the maximum value for $q^*_{p_0,\nu}$ is given by the solution to Formula (3), where

$$q^*_{p_0,\nu} = \sum_{G\in\mathcal{G}} x_G = \frac{1}{N - f - k + 1} \sum_{i=1}^{N-f} \sum_{G\in\mathcal{G}:i\in G} x_G \tag{6}$$
$$\leq \frac{1}{N - f - k + 1} \sum_{i=1}^{N-f} \frac{(p_b^i + C - p_w)^\nu - (C - p_w)^\nu}{(p_b^i + C - p_w)^\nu - (p_b^i + C - p_s)^\nu} \tag{7}$$

Since $\sum_{i=1}^{N-f} p_b^i \leq p_0$ and the function

$$f(x) = \frac{(x + C - p_w)^\nu - (C - p_w)^\nu}{(x + C - p_w)^\nu - (x + C - p_s)^\nu}$$

is convex in x, the sum in equation (7) is maximized when one variable, *e.g.*, p_b^1 is set to be p_0 and the rest becomes 0. Thus, given $C = p_s + p_w$, we obtain

$$q^*_{p_0,\nu} \leq \frac{1}{N - f - k + 1} \frac{(p_s + p_0)^\nu - (p_s)^\nu}{(p_s + p_0)^\nu - (p_w + p_0)^\nu}.$$

We next observe that for any p_b^i such that $0 \leq p_b^i \leq p_s - p_w$[9], and any $\nu \in (0, 1)$, it holds that

$$\frac{(p_s + p_b^i)^\nu - (p_s)^\nu}{(p_s + p_b^i)^\nu - (p_w + p_b^i)^\nu} \leq \frac{p_b^i}{p_s - p_w}$$

[9] Offering more bribes to a node is a waste of coins for the adversary. Without loss of generality, we can assume $p_b^i \leq p_s - p_w$.

Thus, if we relax the constraints for $\sum_{G\in\mathcal{G}:i\in G} x_G$ by replacing their upper bounds with $p_b^i/(p_s-p_w)$ in Formula (3), and maximize the objective, which is possible since the problem has now become convex, we obtain

$$\frac{1}{N-f-k+1}\frac{p_0}{p_s-p_w}.$$

(Note that this is the solution to the optimization problem when $\nu = 1$.)

Finally, we prove the lower bound on $q^*_{p_0,\nu}$. For this purpose, we consider the p_0-adversary $\mathcal{A}$ that offers a bribe of $p_b^i = p_0/((N-f-k+1)(p_s-p_w)) = p_b$ to the nodes $\mathcal{P}_i$, $i \in [N-f-k+1]$, and for the set $Q \subseteq [N-f-k+1]\cup\{N-f+1,\ldots,N\}$, specifies

$$\Pr[X_i = 0, i \in Q] = \tilde{q} = \frac{(C+p_b-p_w)^\nu - (C-p_w)^\nu}{(C+p_b-p_w)^\nu - (C+p_b-p_s)^\nu},$$
$$\Pr[X_i = 1, i \in Q] = 1-\tilde{q} = 1 - \frac{(C+p_b-p_w)^\nu - (C-p_w)^\nu}{(C+p_b-p_w)^\nu - (C+p_b-p_s)^\nu}.$$

We will show that given $\mathcal{A}$, for any compliant slashing function, there exists a Nash equilibrium such that $q_{\mathrm{v}}^{\mathcal{A}} \geq \tilde{q}$, which would imply $q^*_{p_0,\nu} \geq \tilde{q}$. Recall the variables $\tilde{r}_i^*$ and r_i^* from the proof of Theorem 5. In the equilibrium, either $(\tilde{r}_i^*, r_i^*) = (1,-)$ for all $i \in [N-f-k+1]$, or $(\tilde{r}_i^*, r_i^*) = (0,0)$ for all $i \in [N-f-k+1]$. In other words, each node that is offered a bribe either accepts the bribe and becomes corrupted, or does not send a valid clue to the contract at slot $t+1$.

If a node $\mathcal{P}_i$, $i \in [N-f-k+1]$, rejects the bribe, its expected payoff becomes

$$\mathbb{E}[(C+f_i(\mathbf{X})-p_w)^\nu | X_i = 1] r_i^* + \mathbb{E}[(C+f_i(\mathbf{X}))^\nu | X_i = 0](1-r_i^*)$$

As the slashing function satisfies symmetry, for any $i,j \in [N-f-k+1]$, $\mathbb{E}[(C+f_i(\mathbf{X})-p_w)^\nu|X_i=1] = \mathbb{E}[(C+f_j(\mathbf{X})-p_w)^\nu|X_j=1] = e_1$ and $\mathbb{E}[(C+f_i(\mathbf{X}))^\nu|X_i=0] = \mathbb{E}[(C+f_j(\mathbf{X}))^\nu|X_j=0] = e_0$. As $\mathcal{P}_i$ maximizes its utility given all other nodes' actions in the equilibrium, it must be the case that

- $r_i^* = 1$, if $e_1 > e_0$. In this case, $\mathcal{P}_i$'s expected utility becomes e_1.
- $r_i^* = 0$, if $e_1 < e_0$.
- r_i^* can be any value in $[0,1]$, if $e_1 = e_0$. In this case, $\mathcal{P}_i$'s expected utility becomes e_1.

On the other hand, if $e_1 > e_0$ and $\mathcal{P}_i$ accepts the bribe, its expected payoff becomes at least

$$\begin{aligned}
&(C+p_b-p_s)^\nu\,\tilde{q} + \mathbb{E}[(C+p_b+f_i(\mathbf{X})-p_w)^\nu|X_i=1](1-\tilde{q}) \\
&\geq (C-p_w)^\nu + (1-\tilde{q})(\mathbb{E}[(C+p_b+f_i(\mathbf{X})-p_w)^\nu|X_i=1] - (C+p_b-p_w)^\nu) \\
&\geq e_1,
\end{aligned}$$

where the first equality follows from the value of $\tilde{q}$ specified by the adversary, and the last inequality follows from Lemma 2 of Theorem 4. Thus, if $e_1 \geq e_0$,

then the expected payoff of node $\mathcal{P}_i$ when it accepts the bribe is at least as large as its expected payoff when it rejects the bribe. Hence, for the nodes $\mathcal{P}_i$, $i \in [N-f-k+1]$, there does not exist any action that dominates $(\tilde{r}_i^*, r_i^*) = (1, -)$. Then, the action profile specified by the adversary constitutes a Nash equilibrium, and in the equilibrium, $q_{\mathrm{v}}^{\mathcal{A}}$ becomes at least $\tilde{q}$:

$$\frac{(C+p_b-p_w)^\nu-(C-p_w)^\nu}{(C+p_b-p_w)^\nu-(C+p_b-p_s)^\nu}=\frac{(p_b+p_s)^\nu-(p_s)^\nu}{(p_b+p_s)^\nu-(p_b+p_w)^\nu},$$

where $C = p_s + p_w$.

Finally, if $e_1 < e_0$, then either the nodes $\mathcal{P}_i$, $i \in [N-f]$, all reject the bribe, and set $r_i^* = 0$, or they all accept the bribe. The first case happens if for the nodes $\mathcal{P}_i$, $i \in [N-f-k+1]$, there does not exist any action that dominates $(\tilde{r}_i^*, r_i^*) = (0, 0)$. Then, in the equilibrium, $q_{\mathrm{v}}^{\mathcal{A}}$ becomes 1. In the latter case, $q_{\mathrm{v}}^{\mathcal{A}} \geq \tilde{q}$ as argued above, thus concluding the proof. □

Lower and upper bounds on p_0 for risk-averse nodes Let $\tilde{p}_0 = p_0/p_w$ and $\tilde{p}_s = p_s/p_w \approx 1416$ for the given values of $p_s = 32$ ETH and $p_w = 0.0226$ ETH. Using the formula of Theorem 11, we can bound the value of $\tilde{p}_0$ as a function of N, ν and ϵ: $\tilde{p}_{0,\min} \leq \tilde{p}_0 \leq \tilde{p}_{0,\max}$, where $\tilde{p}_{0,\max}$ and $\tilde{p}_{0,\min}$ satisfy the following expressions respectively:

$$\begin{aligned}
\epsilon &= \frac{(1416+3\frac{\tilde{p}_{0,\max}}{N})^\nu-(1416)^\nu}{(1416+3\frac{\tilde{p}_{0,\max}}{N})^\nu-(1+3\frac{\tilde{p}_{0,\max}}{N})^\nu} \\
\epsilon &= \frac{3}{N}\min\left(\frac{\tilde{p}_{0,\min}}{1415}, \frac{(1416+\tilde{p}_{0,\min})^\nu-(1416)^\nu}{(1416+\tilde{p}_{0,\min})^\nu-(1+\tilde{p}_{0,\min})^\nu}\right) \\
&= \frac{3}{N}\frac{(1416+\tilde{p}_{0,\min})^\nu-(1416)^\nu}{(1416+\tilde{p}_{0,\min})^\nu-(1+\tilde{p}_{0,\min})^\nu} \text{ for } N < 300,000
\end{aligned}$$

Solving for $\tilde{p}_{0,\min}$ and $\tilde{p}_{0,\max}$ at different values of N, ν and ϵ, we can calculate the lower and upper bounds on p_0 for different $N < 300,000$, utility functions of the form $U(x) = x^\nu$ and ϵ. These bounds are presented by Table 2 and Figure 1. We observe that when $p_0\nu/p_s << 1$, the upper and lower bounds differ by at most a constant factor for all values of ν.

D Proofs of Theorems 6, 7 and 8

Proof of Theorem 6. We first show that for any (p_0, p_1)-adversary, $q(p_0, \nu) \leq q^*_{p_0,\nu}$ for the contract of Section 3. Consider a game played among the client and the DAC nodes. At the beginning of slot 2, the client $\mathcal{V}$ either received k or more valid clues over the network from the nodes, or it did not. Recall the definitions of the events $Q_{\geq k}$, $Q_{<k}$, R and P from the proof of Lemma 1. Recall Table 4 summarizing $\mathcal{V}$'s payoff under different actions taken by $\mathcal{V}$.

Since 4-security is violated only in the event $\overline{R} \wedge \overline{P}$, given the slashing function of Section 3, $q(p_0, \nu) = \Pr[\overline{R} \wedge \overline{P}]$. If $\Pr[\overline{R} \wedge \overline{P}] = 0$ in all Nash equilibria, it

trivially holds that $0 = q(p_0, \nu) \leq q^*_{p_0,\nu}$. Thus, we next assume that there exists a Nash equilibrium where $\Pr[\overline{R} \wedge \overline{P}] > 0$.

Given that the event R happens, *i.e.*, k or more nodes send clues to $\mathcal{V}$ over the network by slot 3, the adversary cannot distinguish between the actions $Q_{\geq k} \wedge Q_{<k}$ and $Q_{\geq k} \wedge \overline{Q}_{<k}$, and between the actions $\overline{Q}_{\geq k} \wedge Q_{<k}$ and $\overline{Q}_{\geq k} \wedge \overline{Q}_{<k}$. Moreover, the utility $(p_f - p_c)^\nu$ of the events $Q_{\geq k} \wedge Q_{<k}$ and $\overline{Q}_{\geq k} \wedge Q_{<k}$ exceeds the maximum utility $(p_{\text{comp}} - p_c)^\nu$ of the events $Q_{\geq k} \wedge \overline{Q}_{<k}$ and $\overline{Q}_{\geq k} \wedge \overline{Q}_{<k}$ with the bribe. Thus, even if the adversary offers an additional payoff of $p_1 < p_{\text{comp}} - p_c$ for the actions $Q_{\geq k} \wedge \overline{Q}_{<k}$ or $\overline{Q}_{\geq k} \wedge \overline{Q}_{<k}$, if $\Pr[\overline{R} \wedge \overline{P}] > 0$, then $\mathcal{V}$ can increase its expected utility by reducing the probability of taking $Q_{\geq k} \wedge \overline{Q}_{<k}$ in favor of $Q_{\geq k} \wedge Q_{<k}$, and reducing the probability of taking $\overline{Q}_{\geq k} \wedge \overline{Q}_{<k}$ in favor of $\overline{Q}_{\geq k} \wedge Q_{<k}$. Hence, $\mathcal{V}$ never takes the actions $Q_{\geq k} \wedge \overline{Q}_{<k}$ and $\overline{Q}_{\geq k} \wedge \overline{Q}_{<k}$ with positive probability in any equilibrium where $\Pr[\overline{R} \wedge \overline{P}] > 0$. In other words, if $\mathcal{V}$ does not receive k or more clues over the network by slot 3, it sends a query to the contract by slot 2 in any equilibrium with a positive failure probability for 4-security.

Suppose a query appears in the contract of Section 3 at some slot t. By the optimality of the contract, for any given p_0 and ν, no compliant contract with a different slashing function can ensure that there are less than k valid clues in the contract at slot $t+1$ with probability less than $q^*_{p_0,\nu}$ in the equilibrium with the maximum failure probability.

Finally, if the event R happens, *i.e.*, the nodes send k or more clues over the network by slot 3, then 4-security is satisfied with probability 1. If R does not happen, then the client sends a query to the contract by slot 2, in which case there are k or more clues in the contract by slot 4 except with probability $q^*_{p_0,\nu}$ in the equilibrium with the maximum failure probability. Consequently, 4-security is violated with probability at most $q^*_{p_0,\nu}$, implying that $q(p_0, \nu) \leq q^*_{p_0,\nu}$ for the contract of Section 3.

For the achievability claim, consider a compliant contract and the adversary $\mathcal{A}$ from Theorem 4. Suppose there exists a subgame perfect equilibrium, where none of the nodes sends a valid clue over the network by slot 3, and $\mathcal{V}$ takes the action $\mathcal{S}_{\mathsf{q}}$, *i.e.*, sends its query to the contract, by slot 2. Once the query appears in the contract, the nodes follow the actions specified by $\mathcal{A}$ in the proof of Theorem 4.

Observe that if there is a query in the contract, none of the nodes can increase its expected utility by deviating from the action specified by the adversary $\mathcal{A}$ given the other nodes' actions. Similarly, $\mathcal{V}$ cannot increase its expected utility by taking the action $\neg\mathcal{S}_{\mathsf{q}}$ when none of the nodes sends a valid clue over the network by slot 3. Finally, none of the nodes can increase its expected utility by sending a valid clue to $\mathcal{V}$ over the network by slot 2, since this does not affect $\mathcal{V}$'s behavior. This is because $k > 1$ and all other nodes withhold their clues. Hence, the claimed action profile indeed constitutes a subgame perfect equilibrium. However, in this case, the p_0-adversary $\mathcal{A}$ ensures that less than k valid clues are sent to the contract by slot 4 with probability at least $q^*_{p_0,\nu}$ for any compliant contract. Consequently, there exists a $(p_0, 0)$-adversary and

a subgame perfect equilibrium such that 4-security is violated with probability $q^*_{p_0,\nu}$. □

Proof of Theorem 7. We first augment the adversary $\mathcal{A}$ from Theorem 4 to also offer a bribe of p_1 to the client $\mathcal{V}$ such that $p_1 \leq p_{\text{comp}} - p_c$ and p_1 satisfies formula (1). In return, $\mathcal{A}$ asks $\mathcal{V}$ to take the action $\mathcal{S}_{\mathsf{q}}$, *i.e.* to send its query to the contract, by slot 2 in addition to the actions specified for the nodes. We show that the action profile, where $\mathcal{V}$ and the nodes accept their bribes, and none of the nodes sends valid clues over the network to $\mathcal{V}$ by slot 3, constitutes a subgame perfect equilibrium, where security is violated with probability at least $q^*_{p_0,\nu}$.

Suppose a query appears in the contract at some slot t. By the optimality of the slashing function of Section 3, for any given p_0 and ν, no contract with a different slashing function can ensure that there are less than k valid clues in the contract at slot $t+1$ with probability less than $q^*_{p_0,\nu}$ in the Nash equilibrium with the maximum failure probability.

Suppose $\mathcal{V}$ receives no valid clues over the network by slot 3. Then, if $\mathcal{V}$ does not take the action $\mathcal{S}_{\mathsf{q}}$ by slot 2, $\mathcal{V}$'s expected utility becomes at most 0. Conversely, if $\mathcal{V}$ takes the action $\mathcal{S}_{\mathsf{q}}$ by slot 2, it obtains a utility of at least $(p_{\text{comp}} - p_c + p_1)^\nu$. Hence, if none of the nodes sends a valid clue to $\mathcal{V}$ by slot 3, since $(p_{\text{comp}} - p_c + p_1)^\nu > 0$, $\mathcal{V}$ chooses to take the action $\mathcal{S}_{\mathsf{q}}$ by slot 2.

On the other hand, consider an action profile where one or more of the nodes sends a valid clue to $\mathcal{V}$ over the network by slot 2. In this case, if $\mathcal{V}$ accepts the bribe and takes the action $\mathcal{S}_{\mathsf{q}}$ by slot 2, its utility becomes $(p_f - p_c + p_1)^\nu$ with probability at most $1 - q^*_{p_0,\nu}$ and $(p_f - p_c + p_1 + p_{\text{comp}})^\nu$ with probability at least $q^*_{p_0,\nu}$. Thus, its expected utility becomes at least $(1 - q^*_{p_0,\nu})(p_f - p_c + p_1)^\nu + q^*_{p_0,\nu}(p_f - p_c + p_1 + p_{\text{comp}})^\nu$. However, if $\mathcal{V}$ rejects the bribe, its expected payoff can at most be $(p_f)^\nu$. Since this is less than $(1 - q^*_{p_0,\nu})(p_f - p_c + p_1)^\nu + q^*_{p_0,\nu}(p_f - p_c + p_1 + p_{\text{comp}})^\nu$ by formula (1), $\mathcal{V}$ cannot increase its expected utility by rejecting the bribe even if one or more of the nodes sends a valid clue over the network by slot 2. Hence, regardless of whether the nodes send valid clues to $\mathcal{V}$ by slot 2 or not, $\mathcal{V}$ sends a query to the contract by slot 2, implying that the nodes cannot increase their expected utility by deviating from the action profile claimed to be a subgame perfect equilibrium.

Finally, the action profile, where $\mathcal{V}$ accepts the bribe and sends a query to the contract by slot 2, and the nodes do not send valid clues over the network by slot 3, indeed constitutes a subgame perfect equilibrium. However, in this case, the p_0-adversary $\mathcal{A}$ ensures that less than k valid clues are sent to the contract by slot 4 with probability at least $q^*_{p_0,\nu}$ for any compliant contract. Hence, 4-security is violated with probability at least $q^*_{p_0,\nu}$. □

Proof of Theorem 8. For the sake of contradiction, suppose there exists a (p_0, p_1)-adversary and a subgame perfect equilibrium, where all of the nodes withhold their clues from the client $\mathcal{V}$ with some positive probability. By the proof of Theorem 6, if $\mathcal{V}$ receives no valid clues over the network by slot 2, it takes the action $\mathcal{S}_{\mathsf{q}}$, *i.e.* sends query to the contract, by slot 2. Since $p_0 < (N - f)p_w$, given

any distribution of bribes to the nodes, there exists a non-adversarial node $\mathcal{P}$, which receives a bribe less than p_w. Thus, in the subgame, where $\mathcal{V}$ takes the action $\mathcal{S}_{\mathsf{q}}$ by slot 2, the maximum expected utility of $\mathcal{P}$ becomes less than $(C)^\nu$ if it takes the action $\mathcal{S}_{\mathsf{p}}$ at slot 3, *i.e.*, sends its clue to the contract, as its bribe cannot compensate for the cost p_w of sending a clue to the contract. Similarly, $\mathcal{P}$'s expected utility becomes less than $(C)^\nu$ if it does not take the action $\mathcal{S}_{\mathsf{p}}$ as its bribe cannot compensate for the minimum punishment $p_w + \epsilon$ of not sending a clue to the contract. Thus, in this subgame perfect equilibrium, $\mathcal{P}$'s expected utility is less than $(C)^\nu$.

Given the action profile, where one of the nodes sends a valid clue to $\mathcal{V}$ over the network by slot 2, thus enabling $\mathcal{V}$ to recover the response to its query, if $\mathcal{V}$ takes the action $\mathcal{S}_{\mathsf{q}}$, its utility becomes $(p_f - p_c + p_1)^\nu$ with probability at least $1 - q^*_{p_0,\nu}$ and $(p_f - p_c + p_1 + p_{\text{comp}})^\nu$ with probability at most $q^*_{p_0,\nu}$. In this case, $\mathcal{V}$'s expected utility is upper bounded by $(1 - q^*_{p_0,\nu})(p_f - p_c + p_1)^\nu + q^*_{p_0,\nu}(p_f - p_c + p_1 + p_{\text{comp}})^\nu$. Conversely, if $\mathcal{V}$ does not take the action $\mathcal{S}_{\mathsf{q}}$, its payoff becomes $(p_f)^\nu$. As

$$(1 - q^*_{p_0,\nu})(p_f - p_c + p_1)^\nu + q^*_{p_0,\nu}(p_f - p_c + p_1 + p_{\text{comp}})^\nu < (p_f)^\nu,$$

if at least one node sends a valid clue to $\mathcal{V}$ over the network by slot 2, $\mathcal{V}$ does not take the action $\mathcal{S}_{\mathsf{q}}$ at any slot.

Finally, if $\mathcal{P}$ deviates from the claimed equilibrium and sends a valid clue to $\mathcal{V}$ over the network by slot 2, $\mathcal{V}$ does not take the action $\mathcal{S}_{\mathsf{q}}$ before the game ends. In this case, $\mathcal{P}$ obtains a payoff of at least $(C)^\nu$. However, this is larger than $\mathcal{P}$'s claimed equilibrium payoff, implying a contradiction. Thus, in all subgame perfect equilibria, there is at least one node that sends a valid clue to $\mathcal{V}$ by slot 2 over the network, in which case $\mathcal{V}$ does not take the action $\mathcal{S}_{\mathsf{q}}$ before the game ends. Consequently, all subgame perfect equilibria of this game satisfies 4-security without the use of the contract. □