



Optimal Multilevel Slashing for Blockchains

Kenan Wood  

Davidson College, NC, USA

Hammurabi Mendes  

Davidson College, NC, USA

Jonad Pulaj  

Davidson College, NC, USA

Abstract

We present the notion of *multilevel slashing*, where proof-of-stake blockchain validators can obtain gradual levels of assurance that a certain block is bound to be finalized in a global consensus procedure, unless an increasing and optimally large number of Byzantine processes have their staked assets *slashed* – that is, deducted – due to provably incorrect behavior. Our construction is a highly parameterized generalization of combinatorial intersection systems based on finite projective spaces, with asymptotic high availability and *optimal* slashing properties. Even under weak conditions, we show that our construction has asymptotically optimal slashing properties with respect to message complexity and validator load; this result also illustrates a fundamental trade off between message complexity, load, and slashing. In addition, we show that *any* intersection system whose ground elements are disjoint *subsets* of nodes (e.g. “committees” in committee-based consensus protocols) has asymptotic high availability under similarly weak conditions. Finally, our multilevel construction gives the flexibility to blockchain validators to decide how many “levels” of finalization assurance they wish to obtain. This functionality can be seen either as (i) a form of an early, slashing-based block finalization; or (ii) a service to support reorg tolerance.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Mathematics of computing → Discrete mathematics

Keywords and phrases Blockchains, Finality, Slashability, Committees, Availability

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2024.8

1 Introduction

Blockchains are distributed systems with the task of (i) collecting concurrent *transactions* that originate from its users; (ii) order these transactions into a history, which is often expressed as a linear sequence of *blocks*, with each block containing an internal sequence of transactions; and (iii) maintain such history stored into a permanent, distributed ledger that reflects a global system state given the history. The ledger state could represent monetary balances or even the state of replicated programs (smart contracts) that execute in a virtual machine collectively simulated by participant nodes [26, 6, 33, 16, 2, 19]. Throughout this paper, we may refer to participant nodes, processes, and blockchain validators interchangeably.

Deciding how to group transactions into blocks and how to order blocks into the blockchain ledger is an application of Byzantine consensus, typically with some other properties and requirements in place, including: (i) participants always use digital signatures when they interact with the system; (ii) participation is dynamic, meaning that nodes join and leave the system at undetermined times; and (iii) in certain cases, participation is *permissionless*, meaning that certain system actions (say, performing transactions or even participating in the consensus itself) require no previous global registration or identity-based approval. For example, Bitcoin participation is permissionless as any node that can produce a token demonstrating the completion of a certain computationally-expensive hashing task is allowed to participate in the consensus protocol, regardless of that node’s identity. This is called a



© Kenan Wood, Hammurabi Mendes, and Jonad Pulaj;
licensed under Creative Commons License CC-BY 4.0

28th International Conference on Principles of Distributed Systems (OPODIS 2024).

Editors: Silvia Bonomi, Letterio Galletta, Etienne Rivière, and Valerio Schiavoni; Article No. 8; pp. 8:1–8:18



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

proof-of-work blockchain. Note that any two nodes could produce the tokens mentioned above roughly at the same time, so the ledger can *fork* during the system execution, and participants then heuristically define the longest chain of blocks as the authoritative one. In contrast, systems such as [6, 33, 16, 2, 19] work overall as follows. Discrete time *slots* are defined, each associated with a *committee* of participants. At each slot, the corresponding committee produces or acknowledges the next block of transactions by having some well-defined fraction of its participants to *attest* – that is, vote – for such next block. If a Byzantine committee participant votes for two conflicting blocks (on different branches), which is prohibited by the voting mechanics, the participant is subject to *slashing*: the participant’s pre-deposited assets (called its *stake*) are penalized, and the deducted penalty is distributed to other participants in the system. Having such pre-deposited assets is a prerequisite for participating in committees (and thus in the consensus protocol), and slashing is the incentive for nodes to behave correctly. This setting characterizes a *proof-of-stake* blockchain.

An operational advantage of using committees rather than having a global vote procedure is that committees can initially restrict expensive communication primitives [37, 4, 7] to its own participants, and later generate a compact *committee signature* that indicates internal agreement on a certain next block v (say, using a threshold signature scheme such as [3, 36]). Those compact committee signatures can then be communicated globally with the intention to reach global consensus on v .

Just as in proof-of-work systems, the ledger in proof-of-stake systems can also fork because committees can be temporarily or permanently isolated from the network due to technical outages, making attestations arrive asynchronously in different parts of the network, and thus creating multiple descendants of a given block. Hence, it is common to have a heuristic-based *fork-choice rule* that constantly defines the “best chain” of ongoing operations¹, along with a separate *finalization gadget*, which chooses one unique, canonical chain to be “final” [29, 5, 38]. The blockchain literature often refers to the fork-choice’s “best chain” as the *available chain*, because its relatively simple heuristics allow applications to identify it quickly, thus settle quickly on the current state of the system. That is in contrast with what is often referred as the *final chain*, which has been subject to the finalization gadget, and often depends on partially-synchronous assumptions for progress ([9, 40] among many others).

The problem we solve is that blockchain applications often need to know whether a transaction is “confirmed” (perhaps to settle a sale), which requires (a) observing a ledger block that includes such transaction; and (b) obtaining some guarantee that this block will be finalized later. However, currently, applications can either (i) *quickly* identify the transaction in the available chain, but have no guarantees that this chain will eventually be finalized; or (ii) *slowly* identify the transaction in the finalized chain, but be subject to an infeasible wait time that might completely break the application’s functional requirements (user satisfaction, quick response to financial events, etc).

Our solution creates a “sliding window” in the latency-trust spectrum in settling transactions, and the applications can tradeoff speed and certainty according to their very particular functional requirements. In other words, applications will not only have the fork-choice rule (a temporary “accounting mechanism”) or the finalization event (the permanent but slowly-moving global consensus) to deem that a particular block (or a particular transaction therein) is “confirmed”. Specifically, we create a distributed mechanism where increasing *levels* of trust can be obtained at increasing latency costs by querying other

¹ Just like Bitcoin does, except that Bitcoin’s heuristic is simply defined as the longest chain.

participant nodes as well as passively observing network events. Importantly, our design does not introduce any central control or “hotspots”, as it is defined using the highly symmetrical mathematical structure of projective spaces over finite fields.

In essence, our construction can be interpreted as an intermediary, flexible *trust* phase between the initial attestation tallying and the finalization. In this phase, a participant can obtain information from multiple sets of validators – which we call *quorums* – indicating that a block v is about to be finalized. If another participant obtains the same information for a different block v' , the intersection property of our structural construction will result in a significant amount of funds being slashed from the adversary. Importantly, we give the applications the flexibility to choose the balance of trust and potential adversarial slashing with our construction.

We note that this functionality, if further integrated into a blockchain system, can also be seen as (i) a form of an early, slashing-based *block finalization*, as the block confirmation guarantees are now much more continuous between the quick, yet unreliable, fork-choice rule and the slow, yet reliable, finalization; or (ii) a service for *reorg tolerance*. Reorgs [35, 30] are situations where applications consider some chain v as the logical continuation of the blockchain ledger because a fraction of nodes attest v , but later are forced to consider v' as such because a (typically larger) fraction of nodes became visible while attesting v' . In our system, once applications obtain a certain number of levels of assurance that v is the next block to be finalized, a different block v' that takes its place will incur significant adversarial slashing, optimal with respect to the magnitude of assurance levels originally obtained.

Our technical contributions are described below at a high level, but with pointers to the sections where we present our concrete constructions and proofs.

1. We design a distributed architecture to support multilevel slashing by applying projective spaces over finite fields to committee-based consensus (Section 3), a generalization of a previous approach that only used projective *planes* [32] in a context where slashing was irrelevant.
2. We define and analyze *slashability* – the relation of the query size/time and the magnitude of slashing associated with a level of trust. In particular, we show that our construction is *optimal* with respect to worst-case message complexity and validator load, demonstrating a fundamental trade off between slashability, message complexity, and load (Section 4).
3. We prove that a *general* class of similarly-designed intersection systems based on disjoint subsets of elements achieve asymptotic high availability under reasonable conditions (Section 5).

Our construction creates an intersection pattern among sets of blockchain nodes in a manner that is reminiscent of quorum systems [27, 23, 24, 8, 14] (among others, discussed in Section 6), but our purpose – and design – are not the same. Specifically, the mathematical intersections among sets of nodes intentionally uses a projective-space-based construction in order to define an *additional* level of transaction confirmation on top of *existing* blockchains that follow the availability-finality paradigm of [29]. We note that obtaining the higher-dimensional structures used to define our quorums is expensive, but can be done *a priori* for reasonable parameters, and later mapped to a running system, which we consider viable in practice. We discuss some practical scenarios in Section 3.

In addition to the core technical sections pointed to above, we include background on intersection systems and projective spaces in Section 2; we present our system design concretely in Section 3; we discuss related work in Section 6; and we conclude with final remarks in Section 7.

2 Intersection Systems and Projective Spaces

In this section, we present basic definitions on intersection systems and projective spaces over finite fields, used in our intersection system construction, provided in Section 3. In this paper we use the following standard combinatorial notation: for a positive integer m , we denote by $[m]$ the set $\{1, \dots, m\}$. For probability computations, $\mathbf{P}(A)$ denotes the probability of an event A and $\mathbf{E}X$ denotes an expectation of a random variable X .

2.1 Intersection Systems

Let us start with a definition of intersection system, below:

► **Definition 1 (Intersection System).** *An intersection system is simply a nonempty finite collection of finite sets \mathcal{Q} such that any two sets $A, B \in \mathcal{Q}$ have a nonempty intersection. The sets in \mathcal{Q} are called quorums, and we refer to $\bigcup \mathcal{Q}$ as the ground set of \mathcal{Q} .*

Intersection systems provide a framework for ensuring trust among decided or finalized blocks. Let \mathbb{P} be a set of n processes. Suppose the ground set of an intersection system \mathcal{Q} is the set of processes \mathbb{P} . Then if all processes in some quorum $A \in \mathcal{Q}$ attest to a block v and all processes in some quorum $B \in \mathcal{Q}$ attest to $v' \neq v$, all processes will eventually learn this information (since every message is attached with a digital signature and the network is partially synchronous). We will therefore be able to deduce that every process in the nonempty set $A \cap B$ attested to different blocks and can slash these processes' staked assets. It is important to note that honest validators (those that do not attest to different blocks) never have their stake slashed with this protocol, even if they attest to a block that is not finalized.

Now, observe that if every quorum contains an adversarial process, these processes can simply be silent forever, which means that no decision can ever be made with this protocol, even if every correct process attested to the same block. This motivates Definition 2.

Note that Definitions 2 and 6 are similar to concepts in [27], but their context is on replicated databases, not blockchain applications.

► **Definition 2 (Availability).** *Let \mathcal{Q} be an intersection system. Give each element of $\bigcup \mathcal{Q}$ a fixed probability of availability p , so the elements are independently non-faulty with probability p and faulty (Byzantine) with probability $1 - p$. Let $F_p(\mathcal{Q})$ denote the probability that every quorum in \mathcal{Q} has at least one faulty element. The quantity $A_p(\mathcal{Q}) := 1 - F_p(\mathcal{Q})$ is called the availability of \mathcal{Q} with respect to p .*

Another potential problem is that it is possible for two processes to trust different blocks if every process in $A \cap B$ is Byzantine, for quorums $A, B \in \mathcal{Q}$. Thus if $\min_{A, B \in \mathcal{Q}} |A \cap B|$ is small, then it is possible for an adversary to make processes finalize different blocks with only a small amount of its stake being slashed. Thus, a desirable property of intersection systems is that $|A \cap B|$ should be large for all $A, B \in \mathcal{Q}$.

► **Definition 3 (Slashability).** *For an intersection system \mathcal{Q} , define the slashability of \mathcal{Q} to be the quantity $\min_{A, B \in \mathcal{Q}} |A \cap B|$. The slashability of \mathcal{Q} is denoted $\text{slash}(\mathcal{Q})$.*

This definition of slashability is most relevant when validators have uniform stake. In practice, for heterogeneous validator stakes, committee-based constructions like ours in Section 3 could be adapted using techniques such as *node virtualization*, where high-stake nodes “simulate” multiple nodes proportional to their stake.

In our design, the elements of the quorums are disjoint *committees*, which are sets of processes in \mathbb{P} .

► **Definition 4.** If \mathbb{Q} is an intersection system whose elements are disjoint committees and $r \in (\frac{1}{2}, 1)$, we denote by $\mathbb{P}_r(\mathbb{Q})$ the intersection system

$$\mathbb{P}_r(\mathbb{Q}) = \{S \subseteq \mathbb{P} : \exists Q \in \mathbb{Q}, \forall Q' \in \mathbb{Q}, |Q \cap S| \geq r|Q|\}.$$

That is, a set of processes forms a quorum in $\mathbb{P}_r(\mathbb{Q})$ if it contains at least an r -fraction of every committee inside a quorum $Q \in \mathbb{Q}$. The number r is said to be the threshold of $\mathbb{P}_r(\mathbb{Q})$.

The following definitions have fundamental connections to slashability, as seen in Section 4. In our system, we assume that processes *actively* participate in obtaining quorums to reduce message complexity. In particular, committees select quorums uniformly at random to which they query messages. Thus, having small quorums is necessary, motivating the following definition.

► **Definition 5 (Message Complexity).** Given an intersection system \mathbb{Q} , let the maximum size of a quorum in \mathbb{Q} be called the message complexity of \mathbb{Q} , denoted $\text{msg}(\mathbb{Q})$.

Additionally, with this system of actively obtaining quorums, we would like to ensure that no committee is overly busy handling queries, motivating another concept:

► **Definition 6 (Load).** Given an intersection system \mathbb{Q} , the load of some $C \in \bigcup \mathbb{Q}$, denoted $\text{load}_{\mathbb{Q}}(C)$, is the probability that a quorum of \mathbb{Q} selected uniformly at random contains C . The load of \mathbb{Q} is defined to be the maximum load of any element of $\bigcup \mathbb{Q}$: $\text{load}(\mathbb{Q}) := \max_{C \in \bigcup \mathbb{Q}} \text{load}_{\mathbb{Q}}(C)$.

When \mathbb{Q} is clear from context, we simply write $\text{load}(C)$ instead of $\text{load}_{\mathbb{Q}}(C)$, where $C \in \bigcup \mathbb{Q}$. There is a connection between the load of an element of an intersection system and the degree of that element (using terminology from graph theory).

► **Definition 7 (Degree).** Given an intersection system \mathbb{Q} , the degree of some $C \in \bigcup \mathbb{Q}$ in \mathbb{Q} , written $\text{deg}_{\mathbb{Q}}(C)$, is the number of quorums of \mathbb{Q} containing C . The maximum degree of any element of $\bigcup \mathbb{Q}$ is denoted $\Delta(\mathbb{Q})$.

When \mathbb{Q} is clear from context, we write $\text{deg}(C)$ instead of $\text{deg}_{\mathbb{Q}}(C)$, where $C \in \bigcup \mathbb{Q}$. The following result should be clear from the definition of load and uniform selection.

► **Observation 8.** If \mathbb{Q} is an intersection system and $C \in \bigcup \mathbb{Q}$, then

$$\text{load}(C) = \frac{\text{deg}(C)}{|\mathbb{Q}|} \quad \text{and} \quad \text{load}(\mathbb{Q}) = \frac{\Delta(\mathbb{Q})}{|\mathbb{Q}|}.$$

2.2 Projective Spaces

Projective geometry provides a rich source of intersection systems that are highly symmetric (having a transitive automorphism group). Most of the definitions and notation in this section are similar to those presented in [11]. To begin, it is known that finite fields have prime power order, and for each prime power q , there exists a unique finite field of order q , up to isomorphism. Thus, given a prime power q , we may let \mathbb{F}_q denote the finite field of order q . For the following definitions, let V be a vector space over a field \mathbb{F} .

► **Definition 9 (Projective Space).** The projective space of V , denoted $\text{PG}(V)$, is the set of 1-dimensional vector subspaces of V . In the case when $\mathbb{F} = \mathbb{F}_q$ for a prime power q and $V = \mathbb{F}^{k+1}$, we may write $\text{PG}(k, q)$ instead of $\text{PG}(V)$. If V is finite-dimensional, then the projective dimension of $\text{PG}(V)$ is $\dim \text{PG}(V) = \dim V - 1$.

8:6 Optimal Multilevel Slashing for Blockchains

► **Definition 10** (Projective Subspace). *If U is a vector subspace of V , then $\text{PG}(U)$ is a projective subspace of $\text{PG}(V)$.*

► **Definition 11.** *If $d \geq 0$, let $\text{PG}_d(V)$ be the set of all projective subspaces of $\text{PG}(V)$ with (projective) dimension d . Just as before, if $\mathbb{F} = \mathbb{F}_q$ and $V = \mathbb{F}^{k+1}$, we write this as $\text{PG}_d(k, q)$.*

The following result will be useful for analyzing the slashability of our system design.

► **Proposition 12.** *Let $k \geq d \geq 0$ and let q be a prime power. Then for any $S, T \in \text{PG}_d(k, q)$, then $S \cap T$ is a projective subspace of $\text{PG}(k, q)$ of dimension at least $2d - k$. If $2d \geq k$, this bound is sharp for some $S, T \in \text{PG}_d(k, q)$.*

Proof. Please refer to Appendix A. ◀

► **Corollary 13.** *If $2d > k$ and q is any prime power, then $\text{PG}_d(k, q)$ is an intersection system.*

The following is a known result. Given nonnegative integers r, s and $q \geq 2$, recall the known q -Gaussian binomial coefficient by the following equality, which yields Proposition 14.

$$\binom{s}{r}_q = \frac{(q^s - 1)(q^s - q) \cdots (q^s - q^{r-1})}{(q^r - 1)(q^r - q) \cdots (q^r - q^{r-1})}.$$

► **Proposition 14.** *For all $k \geq d \geq 0$ and prime powers q , we have*

$$|\text{PG}_d(k, q)| = \binom{k+1}{d+1}_q \quad \text{and} \quad |\text{PG}(k, q)| = \frac{q^{k+1} - 1}{q - 1}.$$

3 System Design

Consider a system of n processes $\mathbb{P} = \{P_1, \dots, P_n\}$. We assume that processes are non-faulty independently with probability p (and faulty with probability $1 - p$); similar failure models have been used in [27, 32]. For practical applications, we use values p of the form $\frac{2}{3} + \epsilon$ for a small ϵ , so that the probability that at least $1/3$ of nodes display Byzantine behavior is negligible (so basic network primitives such as [4, 37] work). We assume authenticated channels, that is, every message sent has a digital signature for which it is computationally infeasible for an adversary to forge.

We now describe a multilevel intersection system where, each level has an intersection system of its own with ground set composed of committees in \mathbb{P} . Obtaining a quorum asserting block v within each level will increase the assurance that a v is bound to be finalized in the global consensus – that is, increase the associated slashing in case v is not finalized. We do this while allowing small quorums relative to system size (thus less communication complexity) and very high slashing relative to the size of the quorums. Our system also ensures a small load, as every committee is in the same number of quorums and no committee is particularly over-represented. **Our construction is defined as follows.**

1. Assume a set-up procedure to generate $|\text{PG}(k, q)|$ committees that equitably partition² the set of processes \mathbb{P} , where $k \geq 0$ is an integer and q is a prime power such that $n \geq |\text{PG}(k, q)|$. Note that when q is a prime power, this is always well-defined. Denote the set of these committees by \mathcal{C} , so that $|\mathcal{C}| = |\text{PG}(k, q)|$.

² An *equitable* partition of a finite set S is a partition of S such that the sizes of the sets in the partition differ by at most 1.

2. Next, define a one-to-one correspondence $\text{com} : \text{PG}(k, q) \rightarrow \mathcal{C}$, and two weakly increasing sequences with length ℓ , the total number of levels:

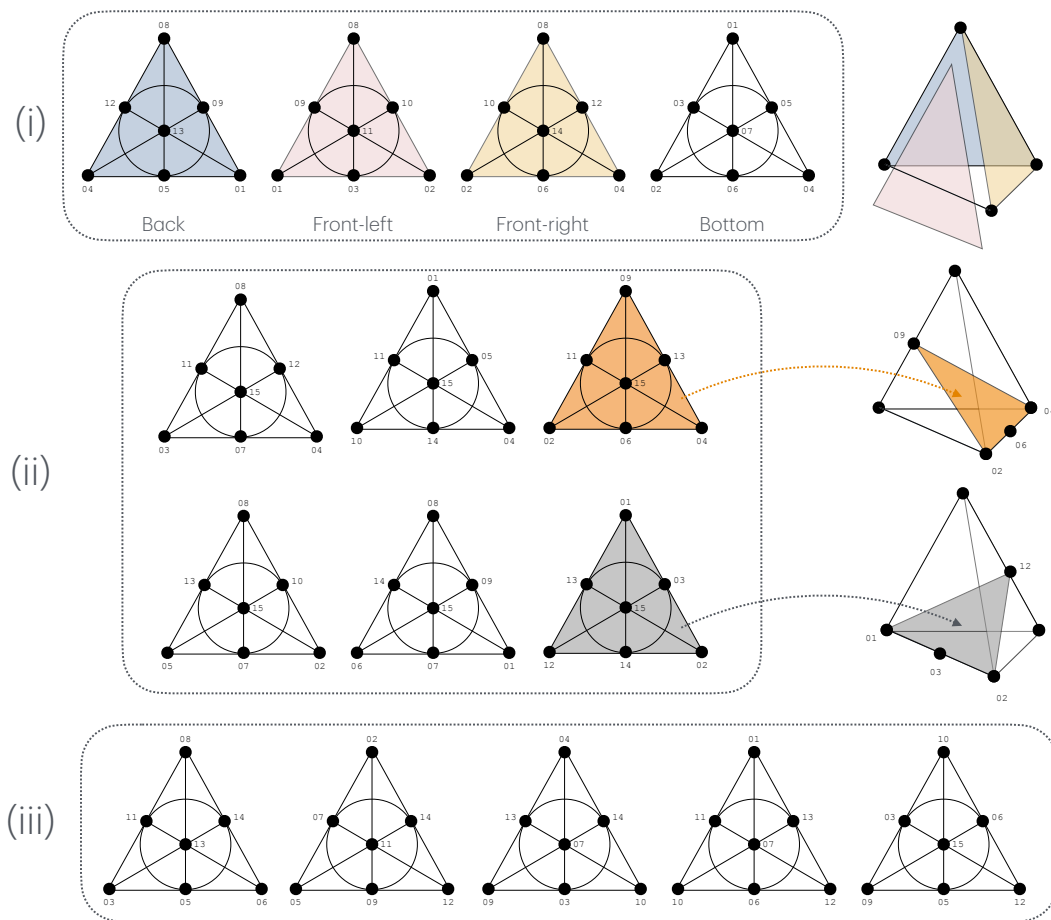
- $(d_j)_{j \in [\ell]}$ of integers in $(\frac{k}{2}, k)$;
- $(r_j)_{j \in [\ell]}$ of real numbers in $(\frac{1}{2}, p)$.

3. For each level $j \in [\ell]$, the j th level committee intersection system is defined to be

$$\mathbb{Q}_j = \{\text{com}(S) : S \in \text{PG}_{d_j}(k, q)\}.$$

4. Finally, for all $j \in [\ell]$, the j th level process intersection system is defined as

$$\mathcal{Q}_j = \mathbb{P}_{r_j}(\mathbb{Q}_j).$$



■ **Figure 1** A tetrahedral visualization of $\text{PG}(3, 2)$, which contains 15 points, 15 planes, and 35 lines (circles are viewed as lines in projective geometry). More details in the text.

Notice that each \mathbb{Q}_j is indeed an intersection system as $\text{PG}_{d_j}(k, q)$ is an intersection system by Corollary 13. In addition, observe that for all $1 \leq i < j \leq \ell$, every quorum in \mathbb{Q}_j contains a quorum in \mathbb{Q}_i ; since every d_j -dimensional projective subspace of $\text{PG}(k, q)$ contains a d_i -dimensional projective subspace (this follows from $d_1 \leq d_2 \leq \dots \leq d_\ell$). Thus, we say that a process *trusts a block v with j degrees of assurance* if that process obtains a quorum accepting v from \mathbb{Q}_j .

Visualization. To motivate the geometry of our construction with an example, in Figure 1 we give a tetrahedral visualization of $PG(3, 2)$, the smallest three-dimensional projective space. While the dimensions are low enough to allow a visualization, they only allow for a single level in our construction (with $d_1 = 2$). Nevertheless, they should be useful to comprehend the system in higher dimensions. For the sake of simplicity, we disregard r and focus on \mathbb{Q}_1 . By definition $PG_2(3, 2)$ is the set of planes in $PG(3, 2)$, where each plane is isomorphic to the Fano plane. In (i), the outer faces of the tetrahedron are each a Fano plane. In (ii), the internal “wedge” planes are represented, with two planes highlighted. In (iii), we show Fano planes that are isomorphic to the additional planes inside the tetrahedron. Mapping the points in each Fano plane to corresponding points in the tetrahedron and preserving the incidence structure recovers the original plane. Our intersection system \mathbb{Q}_1 is set of all the visualized Fano planes. It is straightforward to check that any two distinct quorums intersect in a line, in other words they share three committees.

Choice of Parameters (Example). It is crucial that parameters are chosen carefully, or the number of quorums at each level can quickly become too large. In that case, even the idea of precomputing quorums and later have them mapped to committees at runtime would be impractical. But many reasonable choices might exist: for example, consider a network of 2 million nodes, with committee sizes of about 8000 nodes. We can set $k = 7, q = 2$ and have $d_1 = 4, d_2 = 5$, and $d_3 = 6$, which creates a 3-level intersection system with sizes $|\mathbb{Q}_1| = \binom{8}{5}_2 = 97155$, $|\mathbb{Q}_2| = \binom{8}{6}_2 = 10795$ and $|\mathbb{Q}_3| = \binom{8}{7}_2 = 255$ (Proposition 14). In the first level, applications need to obtain assurances only from 4 committees forming a quorum ($d_1 = 4$) out of 97155 quorums available ($|\mathbb{Q}_1|$). Any conflicting quorum intersects in $2^{2 \cdot 4 - 7 + 1} - 1 = 3$ committees in common (Lemma 16). If we assume that applications get threshold signatures from committees representing a fraction of $r = 60\%$ of their size, this must expose $3 \cdot (0.2 \cdot 8000)$ Byzantine nodes to slashing³. In the second level, applications get *one* extra committee ($d_2 = 5$), essentially choosing one quorum out of 10795 options ($|\mathbb{Q}_2|$). Now, the number of committees in common jumps to $2^{2 \cdot 5 - 7 + 1} - 1 = 15$, thus exposing $15 \cdot (0.2 \cdot 8000)$ Byzantine nodes. In the third level, once again, applications get *one* extra committee ($d_3 = 6$), essentially choosing one quorum out of 255 options ($|\mathbb{Q}_3|$). Now, the number of committees in common jumps to $2^{2 \cdot 6 - 7 + 1} - 1 = 63$, thus exposing $63 \cdot (0.2 \cdot 8000)$ Byzantine nodes. Note how applications have many quorum choices at each level.

Implementation. While the system as described above is mathematically elegant, and achieves asymptotically optimal slashing (Section 4) and high availability (Section 5), the above example points out that the number of quorums per level may become too large. As we (reasonably) assume that there is an operational network cost to keep track of quorums (for instance, joining gossip channels in [39]), the applicability is compromised. In addition, calculating such large sets would be expensive. A probabilistic solution for reducing the number of quorums that works for *reasonable* choices of parameters is the following (reasonable meaning that committee size is large enough so that k, q are small enough).

The first two steps from the above construction remain the same, except, now we have additional parameters that heuristically bound the number of quorums in each level. Let $(\delta_j)_{j \in [l]}$ be a weakly increasing list of positive integers such that there exists δ_j distinct d_j -dimensional projective subspaces of $PG(k, q)$ with a nonempty intersection. The construction of the j th level committee intersection system is now defined as follows. Define a random

³ Two subsets of 60% of a ground set S must intersect in 20% of the nodes in S .

map $N_j : \text{PG}(k, q) \rightarrow 2^{\text{PG}_{d_j}(k, q)}$ by setting $N_j(A)$ to be a set of δ_j distinct d_j -dimensional projective subspaces of $\text{PG}(k, q)$ that contain A , chosen uniformly at random⁴. Then, the j th level committee intersection system will be

$$\mathbb{Q}'_j = \left\{ \text{com}(S) : S \in \bigcup_{A \in \text{PG}(k, q)} N_j(A) \right\},$$

and the corresponding j th level process intersection system is still

$$\mathcal{Q}'_j = \mathbb{P}_{r_j}(\mathbb{Q}'_j).$$

With this construction, each committee is in *approximately* the same number of quorums in \mathbb{Q}'_j with high probability, and we always have that the size of each quorum in \mathbb{Q}'_j equals the size of each quorum in \mathbb{Q}_j . Thus the load of \mathbb{Q}'_j is approximately equal to the load of \mathbb{Q}_j , and the message complexities of \mathbb{Q}'_j and \mathbb{Q}_j are the same (refer to Definitions 6 and 5, and Observation 8). We also know that $\mathbb{Q}'_j \subseteq \mathbb{Q}_j$ by construction, which immediately implies that the slashability of \mathbb{Q}'_j (resp. \mathcal{Q}'_j) is at least that of \mathbb{Q}_j (resp. \mathcal{Q}_j), and equal with very high probability. The availability results we prove in Section 5 only depend on \mathcal{C} , so we obtain the same results with \mathcal{Q}_j and with \mathcal{Q}'_j . However, the number of quorums using this method is significantly reduced. In particular, by Proposition 14,

$$\begin{aligned} |\mathbb{Q}'_j| &= \left| \bigcup_{A \in \text{PG}(k, q)} N_j(A) \right| \leq \sum_{A \in \text{PG}(k, q)} |N_j(A)| = \delta_j |\mathcal{C}| \\ |\mathbb{Q}_j| &= \binom{k+1}{d_j+1}_q \approx q^{(k-d_j)(d_j+1)} \geq q^{kd_j-d_j} \approx |\mathcal{C}|^{d_j - \frac{d_j}{k}} \geq |\mathcal{C}|^{d_j-1}. \end{aligned}$$

Depending on the application, a more dense or less dense intersection system may be desirable. For example, the density (number of quorums out of all possible) may be related to resilience to an adversary that can target specific processes, instead of a case where Byzantine failures are independent and randomly distributed. However, analysis in different failure models is outside the scope of this paper and relegated to future work.

4 Slashability

In this section, we show that our intersection system \mathcal{Q}_j has asymptotically optimal slashability, over a general class of intersection systems constructed from committees. We prove a formalization of the following: The slashability of the j th-level intersection system $\mathcal{Q}_j = \mathbb{P}_{r_j}(\mathbb{Q}_j)$ is asymptotically optimal over all intersection systems built with the same set of committees and the same threshold (r_j) with at least as good overall message complexity and load, allowing a certain trade off between the two quantities.

The proof of the statement above is given in Theorem 20, following some useful lemmas. First, let us compute the size of the quorums in \mathbb{Q}_j and the slashability of \mathcal{Q}_j .

Consider the system with parameters as above, all viewed as a function of the number of processes n . Fix some $j \in [\ell]$ (also a function of n). For a more formalized treatment of asymptotic notations as in this section, see Section 5. Given functions f and g from \mathbb{N} to \mathbb{R} , we say f and g are *asymptotically equivalent*, written $f \sim g$, if $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1$.

These first two lemmas compute $\text{msg}(\mathbb{Q}_j)$ and $\text{slash}(\mathbb{Q}_j)$.

⁴ Given a set S , 2^S denotes the power set of S .

8:10 Optimal Multilevel Slashing for Blockchains

► **Lemma 15.** *The size of each quorum in \mathbb{Q}_j is precisely $\frac{q^{d_j+1}-1}{q-1}$. If $q = \omega(1)$, then this quantity is asymptotically equivalent to q^{d_j} .*

Proof. Consider any $Q \in \mathbb{Q}_j$. Then there exists some $S \in \text{PG}_{d_j}(k, q)$ such that $\text{com}(S) = Q$. This implies that S is a d_j -dimensional projective subspace of $\text{PG}(k, q)$. Hence S is isomorphic to $\text{PG}(d_j, q)$, which has exactly $\frac{q^{d_j+1}-1}{q-1}$ elements, by Proposition 14. Since com is a bijection, it follows that $|S| = |Q| = \frac{q^{d_j+1}-1}{q-1}$. The second statement follows immediately. ◀

► **Lemma 16.** *The slashability of \mathbb{Q}_j is precisely $\frac{q^{2d_j-k+1}-1}{q-1}$. If $q = \omega(1)$, then this quantity is asymptotically equivalent to q^{2d_j-k} .*

Proof. Suppose $Q, R \in \mathbb{Q}_j$. Then there exist $S, T \in \text{PG}_{d_j}(k, q)$ where $\text{com}(S) = Q$ and $\text{com}(T) = R$. By Proposition 12, $S \cap T$ is a projective subspace of $\text{PG}(k, q)$ of dimension at least $2d_j - k$. By Proposition 14, this implies that $|S \cap T| \geq \frac{q^{2d_j-k+1}-1}{q-1}$. Since com is a bijection, this implies that $\text{slash}(\mathbb{Q}_j) \geq \frac{q^{2d_j-k+1}-1}{q-1}$. However, since the bound in Proposition 12 is sharp, this shows $\text{slash}(\mathbb{Q}_j) = \frac{q^{2d_j-k+1}-1}{q-1}$. The second statement follows immediately. ◀

Then Lemmas 15 and 16 show that if $q = \omega(1)$,

$$\text{slash}(\mathbb{Q}_j) \sim q^{2d_j-k} = (q^{d_j})^{2-\frac{k}{d_j}} \sim \text{msg}(\mathbb{Q}_j)^{2-\frac{k}{d_j}}.$$

Hence $\text{slash}(\mathbb{Q}_j) \sim \text{msg}(\mathbb{Q}_j)^{2-\frac{k}{d_j}}$.

In the following result, we assume each committee has the same size, but loosening this restriction does not change the result and is only for simplification.

► **Proposition 17.** *Suppose every committee in \mathcal{C} contains exactly c processes such that $r_j c \in \mathbb{Z}$. Then*

$$\text{slash}(\mathcal{Q}_j) = (2r_j - 1)c \cdot \text{slash}(\mathbb{Q}_j) = (2r_j - 1)c \frac{q^{2d_j-k+1} - 1}{q - 1} = (2r_j - 1) \frac{q^{2d_j-k+1} - 1}{q^{k+1} - 1} n.$$

If $q = \omega(1)$,

$$\text{slash}(\mathcal{Q}_j) \sim (2r_j - 1)q^{2d_j-2k}n$$

Proof. Suppose $S, T \in \mathbb{Q}_j$. Then there exists $\mathcal{S}, \mathcal{T} \in \mathbb{Q}_j$ such that S contains at least $r_j c$ processes from each committee in \mathcal{S} , and similarly for T . Then for each $C \in \mathcal{S} \cap \mathcal{T}$, we have

$$\begin{aligned} |S \cap T \cap C| &= |(S \cap C) \cap (T \cap C)| \\ &= |S \cap C| + |T \cap C| - |(S \cap C) \cup (T \cap C)| \\ &\geq r_j c + r_j c - c = (2r_j - 1)c. \end{aligned}$$

Also, $|\mathcal{S} \cap \mathcal{T}| \geq \text{slash}(\mathbb{Q}_j)$. Hence

$$\begin{aligned} |S \cap T| &= \sum_{C \in \mathcal{C}} |S \cap T \cap C| \\ &\geq \sum_{C \in \mathcal{S} \cap \mathcal{T}} |S \cap T \cap C| \\ &\geq (2r_j - 1)c \cdot \text{slash}(\mathbb{Q}_j). \end{aligned}$$

It follows that $\text{slash}(\mathcal{Q}_j) \geq (2r_j - 1)c \cdot \text{slash}(\mathbb{Q}_j)$.

To prove equality, consider $\mathcal{S}, \mathcal{T} \in \mathcal{Q}_j$ such that $|\mathcal{S} \cap \mathcal{T}| = \text{slash}(\mathcal{Q}_j)$. In the following, we say that the *identity* of process $P_i \in \mathbb{P}$ is i ; the set of process identities is thus $[n]$. Let $S \subseteq \mathbb{P}$ be defined by the following: for all $X \in \mathcal{S}$, pick each of the $r_j c$ processes in X with the least identities to be in S . Define $T \subseteq \mathbb{P}$ similarly, by selecting the $r_j c$ processes of each $X \in \mathcal{T}$ with the greatest identities to be in T . Since $r_j > \frac{1}{2}$, for all $X \in \mathcal{S} \cap \mathcal{T}$, we have $(S \cap X) \cup (T \cap X) = X$, so that

$$|S \cap T \cap X| = |S \cap C| + |T \cap C| - |(S \cap C) \cup (T \cap C)| = r_j c + r_j c - c = (2r_j - 1)c.$$

Hence

$$|S \cap T| = \sum_{X \in \mathcal{S} \cap \mathcal{T}} |S \cap T \cap X| = (2r_j - 1)c \cdot \text{slash}(\mathcal{Q}_j).$$

It follows that $\text{slash}(\mathcal{Q}_j) = (2r_j - 1)c \cdot \text{slash}(\mathcal{Q}_j)$.

Since \mathcal{C} is a partition of \mathbb{P} , we know $c|\mathcal{C}| = n$, so

$$c = \frac{n}{|\mathcal{C}|} = \frac{n}{|\text{PG}(k, q)|} = \frac{n(q-1)}{q^{k+1}-1},$$

by Proposition 14. Substituting this expression for c applying Lemma 16, we obtain

$$\text{slash}(\mathcal{Q}_j) = (2r_j - 1)c \frac{q^{2d_j - k + 1} - 1}{q - 1} = (2r_j - 1) \frac{q^{2d_j - k + 1} - 1}{q^{k+1} - 1} n.$$

The simplified asymptotic expression for $\text{slash}(\mathcal{Q}_j)$ when $q = \omega(1)$ follows immediately. ◀

Let us now show the asymptotic optimality of our system. Denote $m = |\mathcal{C}|$, so m is the number of committees. Recall Definitions 5 and 6. The following defines the set of intersection systems over which we prove optimality.

► **Definition 18.** *Given $r \in (\frac{1}{2}, 1)$, $\lambda \in (0, 1)$, and a positive integer $1 \leq \mu \leq m$, let $\mathbb{S}(\mu, \lambda, r)$ be the set of all intersection systems of the form $\mathbb{P}_r(\mathcal{Q})$, where \mathcal{Q} is an intersection system with ground set \mathcal{C} such that the product of its message complexity and its load is at most $\mu \cdot \lambda$; that is, $\text{msg}(\mathcal{Q}) \cdot \text{load}(\mathcal{Q}) \leq \mu \cdot \lambda$.*

Since we will prove the optimality of \mathcal{Q}_j , we are interested in the maximum slashability of any intersection system in $\mathbb{S}(\text{msg}(\mathcal{Q}_j), \text{load}(\mathcal{Q}_j), r_j)$. A special case of an intersection system in this set is $\mathbb{P}_{r_j}(\mathcal{Q})$, where $\bigcup \mathcal{Q} = \mathcal{C}$ and $\text{msg}(\mathcal{Q}) \leq \text{msg}(\mathcal{Q}_j)$ and $\text{load}(\mathcal{Q}) \leq \text{load}(\mathcal{Q}_j)$, although we prove optimality in a more general setting.

► **Lemma 19.** *Suppose each committee has size c and rc is an integer. Let $1 \leq \mu \leq m$ and $\lambda \in (0, 1)$. For all $\mathcal{Q} \in \mathbb{S}(\mu, \lambda, r)$, we have $\text{slash}(\mathcal{Q}) \leq (2r - 1)c \cdot \mu \cdot \lambda$.*

Proof. Let $\mathcal{Q} \in \mathbb{S}(\mu, \lambda, r)$; let \mathcal{Q} be an intersection system such that $\mathcal{Q} = \mathbb{P}_r(\mathcal{Q})$. Then $\text{msg}(\mathcal{Q}) \cdot \text{load}(\mathcal{Q}) \leq \mu \cdot \lambda$. Select $\mathcal{A}, \mathcal{B} \in \mathcal{Q}$ independently and uniformly at random. Then $\text{slash}(\mathcal{Q}) \leq |\mathcal{A} \cap \mathcal{B}|$, so $\text{slash}(\mathcal{Q}) \leq \mathbf{E}|\mathcal{A} \cap \mathcal{B}|$. For each committee $C \in \bigcup \mathcal{Q}$, let X_C be the indicator random variable for $C \in \mathcal{A}$ and $C \in \mathcal{B}$, so that $|\mathcal{A} \cap \mathcal{B}| = \sum_{C \in \mathcal{C}} X_C$. By linearity of expectation, $\mathbf{E}|\mathcal{A} \cap \mathcal{B}| = \sum_{C \in \mathcal{C}} \mathbf{E}X_C$. A simple double counting argument shows that $\sum_{C \in \mathcal{C}} \deg(C) = \sum_{S \in \mathcal{Q}} |S| \leq \text{msg}(\mathcal{Q}) \cdot |\mathcal{Q}|$, therefore, by Observation 8, we have $\sum_{C \in \mathcal{C}} \text{load}(C) \leq \text{msg}(\mathcal{Q})$. Also, $\text{load}(C) \leq \text{load}(\mathcal{Q})$ for all $C \in \mathcal{C}$. Since for all $C \in \mathcal{C}$, the events $C \in \mathcal{A}$ and $C \in \mathcal{B}$ are independent with probability $\text{load}(C)$, it follows that $\mathbf{E}X_C = \mathbf{P}(C \in \mathcal{A}, C \in \mathcal{B}) = \mathbf{P}(C \in \mathcal{A}) \cdot \mathbf{P}(C \in \mathcal{B}) = \text{load}(C)^2$. Hence,

$$\text{slash}(\mathcal{Q}) \leq \mathbf{E}|\mathcal{A} \cap \mathcal{B}| = \sum_{C \in \mathcal{C}} \text{load}(C)^2 \leq \sum_{C \in \mathcal{C}} \text{load}(\mathcal{Q}) \cdot \text{load}(C) \leq \text{load}(\mathcal{Q}) \cdot \text{msg}(\mathcal{Q}) \leq \mu \cdot \lambda.$$

By the proof of Proposition 17, $\text{slash}(\mathcal{Q}) = (2r - 1)c \cdot \text{slash}(\mathcal{Q}) \leq (2r - 1)c \cdot \mu \cdot \lambda$. ◀

8:12 Optimal Multilevel Slashing for Blockchains

We now come to our main result with respect to slashability, which is a formalization of the informal statements outlined in the beginning of this section.

► **Theorem 20.** *Suppose each committee in \mathcal{C} contains c processes and $r_j c \in \mathbb{Z}$. Then $\mathcal{Q}_j \in \mathbb{S}(\text{msg}(\mathcal{Q}_j), \text{load}(\mathcal{Q}_j), r_j)$ and if $q = \omega(1)$, then*

$$\text{slash}(\mathcal{Q}_j) \sim \max \{ \text{slash}(\mathcal{Q}) : \mathcal{Q} \in \mathbb{S}(\text{msg}(\mathcal{Q}_j), \text{load}(\mathcal{Q}_j), r_j) \}.$$

Proof. It is clear by definition that $\mathcal{Q}_j \in \mathbb{S}(\text{msg}(\mathcal{Q}_j), \text{load}(\mathcal{Q}_j), r_j)$. Suppose $q = \omega(1)$. It is also easy to see that

$$\lim_{n \rightarrow \infty} \frac{\text{slash}(\mathcal{Q}_j)}{\max \{ \text{slash}(\mathcal{Q}) : \mathcal{Q} \in \mathbb{S}(\text{msg}(\mathcal{Q}_j), \text{load}(\mathcal{Q}_j), r_j) \}} \leq 1.$$

By Proposition 14, $\text{msg}(\mathcal{Q}_j) = \frac{q^{d_j+1}-1}{q-1}$. Since every committee in \mathcal{C} has the degree $\Delta(\mathcal{Q}_j)$ in \mathcal{Q}_j and every quorum has size $\text{msg}(\mathcal{Q}_j)$, we also have

$$m \cdot \Delta(\mathcal{Q}_j) = \sum_{C \in \mathcal{C}} \deg_{\mathcal{Q}_j}(C) = \sum_{Q \in \mathcal{Q}_j} |Q| = |\mathcal{Q}_j| \cdot \text{msg}(\mathcal{Q}_j).$$

It follows by Observation 8 that $\text{load}(\mathcal{Q}_j) = \frac{\Delta(\mathcal{Q}_j)}{|\mathcal{Q}_j|} = \frac{\text{msg}(\mathcal{Q}_j)}{m}$.

Also, by Proposition 14, $m = |\mathcal{C}| = |\text{PG}(k, q)| = \frac{q^{k+1}-1}{q-1}$. By Proposition 17, $\text{slash}(\mathcal{Q}_j) = (2r_j - 1)c \frac{q^{2d_j-k+1}-1}{q-1}$. By Lemma 19, we then have

$$\begin{aligned} \frac{\text{slash}(\mathcal{Q}_j)}{\max \{ \text{slash}(\mathcal{Q}) : \mathcal{Q} \in \mathbb{S}(\text{msg}(\mathcal{Q}_j), \text{load}(\mathcal{Q}_j), r_j) \}} &\geq \frac{(2r_j - 1)c \frac{q^{2d_j-k+1}-1}{q-1}}{(2r_j - 1)c \cdot \text{msg}(\mathcal{Q}_j) \cdot \text{load}(\mathcal{Q}_j)} \\ &= \frac{\frac{q^{2d_j-k+1}-1}{q-1}}{\frac{\text{msg}(\mathcal{Q}_j)^2}{m}} \\ &= \frac{q^{2d_j-k+1} - 1}{q - 1} \cdot \frac{\frac{q^{k+1}-1}{q-1}}{\left(\frac{q^{d_j+1}-1}{q-1}\right)^2} \\ &= \frac{(q^{2d_j-k+1} - 1)(q^{k+1} - 1)}{(q^{d_j+1} - 1)^2} \sim 1, \end{aligned}$$

as $n \rightarrow \infty$, where the last 1 denotes the constant function on \mathbb{N} that is always equal to the number 1. It follows that $\text{slash}(\mathcal{Q}_j) \sim \max \{ \text{slash}(\mathcal{Q}) : \mathcal{Q} \in \mathbb{S}(\text{msg}(\mathcal{Q}_j), \text{load}(\mathcal{Q}_j), r_j) \}$. ◀

5 Availability

In this section we demonstrate that under only mild assumptions, *any* multilevel intersection system that generalizes our approach has an asymptotically high availability (including the case when the number of levels is 1). We consider the results of this section relevant on their own, thus the results will be presented with a more general notation (from Section 2), rather than relying on specific notation from our concrete construction in Section 3. With inspiration from some proof ideas seen in [32], we prove a formalized version of the following: Suppose processes are available with probability p . Suppose \mathcal{C} is a partition of the processes into committees and \mathcal{Q} is any intersection system with ground set \mathcal{C} . If $\frac{1}{2} < r < p$ and the smallest committee has size $\Omega(\log n)$, then the intersection system $\mathbb{P}_r(\mathcal{Q})$ has availability converging to 1.

The formal statement and proof of this statement is concluded in Corollary 24, following some useful lemmas below. Lets start by formalizing our assumptions. For each $n \geq 1$, let \mathcal{C}^n be a partition of an n -element set of processes, and let \mathbb{Q}^n be an intersection system with ground set \mathcal{C}^n . Let p be the probability a process is available in its steady-state, with $p > 1/2$. For each $n \geq 1$, let $r_n \in (\frac{1}{2}, p)$ so that $(r_n)_{n \in \mathbb{N}}$ is weakly increasing and $r := \sup_{n \in \mathbb{N}} r_n < p$.⁵ Finally, define $c_n = \min_{C \in \mathcal{C}^n} |C|$; that is, c_n the smallest size of any set within the partition \mathcal{C}^n (that, more generally, models committees).

Now let us bound the probability that a committee of size c does not have at least rc available processes. Denote this probability by $F_p(c; r)$. Define the function $a_1(x) = \frac{(p-x)^2}{2-p-x}$ for $x \in [0, r]$.

► **Lemma 21.** *Suppose $0 < r < p < 1$. We have $F_p(c; r) \leq e^{-a_1(r) \cdot c}$.*

Proof. Let Z be the number of unavailable processes in a committee of size c . It is easy to see that Z is a binomial random variable with parameters c and $q = 1 - p$. Then we have

$$F_p(c; r) = \mathbf{P}(Z > c - rc) \leq \mathbf{P}\left(Z \geq \frac{1-r}{q} \cdot cq\right) = \mathbf{P}\left(Z \geq \left(1 + \frac{1-r-q}{q}\right) \cdot cq\right).$$

Using $\delta = \frac{1-r-q}{q} = \frac{p-r}{q}$ in Lemma 25 from Appendix B ($\delta > 0$ since $r < p$), we obtain

$$F_p(c; r) \leq \exp\left(-\frac{1}{2 + \frac{p-r}{q}} \left(\frac{p-r}{q}\right)^2 cq\right) = \exp\left(-\frac{(p-r)^2}{2-p-r} \cdot c\right) \quad \blacktriangleleft$$

► **Lemma 22.** *For all positive integers n , we have*

$$A_p(\mathbb{P}_{r_n}(\mathbb{Q}^n)) \geq 1 - \frac{n}{c_n} \cdot e^{-a_1(r) \cdot c_n}.$$

Proof. Recall $r = \sup_{n \in \mathbb{N}} r_n < p$. Fix some positive integer n . For each $C \in \mathcal{C}^n$, let E_C be the event that at least $r_n|C|$ of the processes in C are available. Then each E_C is independent since the sets in \mathcal{C}^n are pairwise disjoint. Hence, by Lemma 21,

$$\begin{aligned} A_p(\mathbb{P}_{r_n}(\mathbb{Q}^n)) &\geq \mathbf{P}\left(\bigcap_{C \in \mathcal{C}^n} E_C\right) = \prod_{C \in \mathcal{C}^n} \mathbf{P}(E_C) \\ &= \prod_{C \in \mathcal{C}^n} (1 - F_p(|C|; r_n)) \geq \prod_{C \in \mathcal{C}^n} (1 - \exp(-a_1(r_n) \cdot |C|)). \end{aligned}$$

A straightforward computation shows that for $x \in [0, r]$, we have $a_1'(x) = \frac{(x-p)(4-3p-x)}{(2-p-x)^2}$; since $0 \leq x \leq r < p$, we have $x - p < 0$, $4 - 3p - x > 0$, and $(2 - p - x)^2 > 0$, which implies $a_1'(x) < 0$ on $[0, r]$. Thus $a_1(x)$ is decreasing on $[0, r]$. Since $r_n \leq r < p$, this shows $a_1(r_n) \geq a_1(r)$. Also, $|C| \geq c_n$ for all $C \in \mathcal{C}^n$ implies $|C^n| \leq \frac{n}{c_n}$. Putting these results together, we obtain

$$A_p(\mathbb{P}_{r_n}(\mathbb{Q}^n)) \geq \prod_{C \in \mathcal{C}^n} (1 - \exp(-a_1(r) \cdot c_n)) = (1 - e^{-a_1(r) \cdot c_n})^{|C^n|} \geq (1 - e^{-a_1(r) \cdot c_n})^{\frac{n}{c_n}}.$$

Recall that for all $k \geq 1$ and $0 \leq y \leq x \leq 1$, the following known inequality $k(x-y) \geq x^k - y^k$ holds. Using $k = \frac{n}{c_n}$ (which is at least 1 since $c_n \leq n$), $x = 1$, $y = 1 - e^{-a_1(r) \cdot c_n}$ yields $\frac{n}{c_n}(1 - (1 - e^{-a_1(r) \cdot c_n})) \geq 1 - (1 - e^{-a_1(r) \cdot c_n})^{\frac{n}{c_n}}$, which, following algebraic manipulations, implies the desired bound. \blacktriangleleft

⁵ Note that the r 's presented in this section are not related to the ℓ different r -values used to define our multilevel intersection system in Section 3.

8:14 Optimal Multilevel Slashing for Blockchains

Now we are ready to prove our main theorem. We say a property $R(n)$ that is true or false for every positive integer n holds *for sufficiently large n* if there exists a positive integer N such that $n \geq N$ implies $R(n)$ is true.

► **Theorem 23.** *Suppose $a > 0$ such that for all sufficiently large n , we have $c_n \geq a \log n$. If $a = \frac{b+1}{a_1(r)} = \frac{2-p-r}{(p-r)^2}(b+1)$ for some constant b , then*

$$A_p(\mathbb{P}_{r_n}(\mathbb{Q}^n)) \geq 1 - \frac{1}{a \cdot n^b \cdot \log n} = 1 - O\left(\frac{1}{n^b \cdot \log n}\right).$$

Proof. Using the bound in Lemma 22, for sufficiently large n , we obtain

$$\begin{aligned} A_p(\mathbb{P}_{r_n}(\mathbb{Q}^n)) &\geq 1 - \frac{n}{c_n} \cdot e^{-a_1(r) \cdot c_n} \geq 1 - \frac{n}{a \log n} \cdot e^{-a_1(r) \cdot a \log n} \\ &= 1 - \frac{n}{a \log n} \cdot n^{-a_1(r)a} = 1 - \frac{1}{a \cdot n^{a_1(r)a-1} \cdot \log n} = 1 - \frac{1}{a \cdot n^b \cdot \log n}. \quad \blacktriangleleft \end{aligned}$$

This immediately leads to the following corollary.

► **Corollary 24.** *Suppose a and b are constants satisfying the hypotheses of Theorem 23. If $b \geq 0$, then $\lim_{n \rightarrow \infty} A_p(\mathbb{P}_{r_n}(\mathbb{Q}^n)) = 1$.*

6 Related Work

It is common practice [6, 33, 16, 2, 19] for proof-of-stake blockchains to employ fork-choice rules in order to define an available chain, and rely in an additional “finalization gadget” to execute global consensus. In particular, we note the role of Casper in Ethereum [5], and point to a well-documented conceptual separation of available vs. final chains in [29]. The later paper also points to the fact that this separation, besides practical, is also more general in the sense that a diverse set of consensus algorithms [9, 40, 10] (among others) could be applicable. Reorg attacks [35, 30] are also related to our work as our multilevel construction can precisely quantify a lower bound on slashing if a reorg were to take place – so our work could be seen as a tool to mitigate the functional effects of reorgs.

As noted in the introduction, our work is reminiscent to (and certainly it is inspired by) previous work in *quorum systems*. Quorum systems are set structures whose elements are typically distributed processes (say, servers in a distributed system), so that any two such sets (called *quorums*) intersect in at least one process. The idea is that applications can obtain an acknowledgment of an operation from all members of a chosen quorum, so that any two such acknowledgements are consistent, due to the intersecting property. Quorum systems have been studied extensively [27, 24, 15], with applications to consensus [25], database synchronization [1], finite-state-machine replication [17], mutual exclusion [22], among many others.

We are interested in some crucial metrics, such as server load and system availability, that are also found in the extensive literature of quorum systems, for example in [27, 31]. We note that quorum system design is also quite diverse, being highly impacted by its target applications. For abstract-data-type replication, ADT-specific information can play a role [17, 18]. Another interesting application showing high impact on quorum design is federated Byzantine consensus: in this case, non-uniform quorum systems, where each participant has its own notion of membership, are studied in [15], with ideas originating from practical systems such as the Stellar consensus protocol [25]. An additional important considerations for quorum design are whether the system allows dynamic participation [28, 8].

Block designs are well-studied structures in combinatorics [12] which provide a rich family of parameterized quorum systems [13]. A well-known family of block designs corresponds to finite projective planes and their respective quorum systems. One of its earliest applications in distributed systems is an algorithm that uses $O(\sqrt{n})$ messages to create mutual exclusion in a network, where n is the number of nodes [22]. Although finite projective planes that yield small quorums are desirable due to reduction in communication complexity, asymptotically their availability goes to zero [32, 20]. However, a construction with finite projective planes where points are disjoint subsets that cover the ground set and quorums are achieved on an r fraction of elements in the subsets yields asymptotic high availability [32]. A novel way of leveraging projective spaces in quorum systems for the in-network data storage paradigm is introduced and explored in [34, 21], where a 2D network is “lifted” onto a sphere using a projective map to enable more flexible quorums.

7 Conclusion

In this work, we propose an application of combinatorial designs using projective spaces over finite fields to provide gradual levels of consensus – that is, getting gradual assurance that a certain block is bound to be finalized in a global consensus procedure. We combine our design with a committee-based approach, in order to provide high availability. In fact, we prove not only that our construction is subject to high availability, but we show that *any* approach that uses our “sliding window” of obtaining attestations in the range strictly between $\frac{1}{2}$ and the individual process availability also forms a highly available intersection system. In addition, we demonstrate that our construction has optimal slashability – the extent we penalize an adversary’s assets upon protocol non-compliance – compared to other systems that operate under similar load and message complexity.

We consider the potential applicability of our system very exciting, as we envision that highly-connected participant nodes can even buy and sell “trust certificates” associated with assurances that a block is bound to be finalized. That is, nodes obtain collections of individual and threshold signatures that form multiple quorums, and then offer these trust certificates as *insurance* to blockchain applications.

Having a more gradual consensus – a “sliding window” of trust – can enable large improvements in blockchain usability, providing services such as early slashing-based finalization, reorg tolerance, and even supporting transaction insurance for reorgs. We are excited to spearhead initial theoretical work in this direction.

References

- 1 A. El Abbadi and S. Toueg. Maintaining availability in partitioned replicated databases. *ACM Trans. Database Syst.*, 14(2):264–290, June 1989. doi:10.1145/63500.63501.
- 2 Lacramioara Astefanoaei, Pierre Chambart, Antonella Del Pozzo, Edward Tate, Sara Tucci Piergiovanni, and Eugen Zălinescu. Tenderbake - classical BFT style consensus for public blockchains. *CoRR*, abs/2001.11965, 2020. arXiv:2001.11965.
- 3 Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In Yvo G. Desmedt, editor, *Public Key Cryptography — PKC 2003*, pages 31–46, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
- 4 G. Bracha. Asynchronous Byzantine agreement protocols. *Information and Computation*, 75(2):130–143, November 1987. doi:10.1016/0890-5401(87)90054-X.
- 5 Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. *CoRR*, abs/1710.09437, 2017. arXiv:1710.09437.

- 6 Vitalik Buterin, Diego Hernandez, Thor Kampefner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Ying Wang, and Yan X. Zhang. Combining GHOST and casper. *CoRR*, abs/2003.03052, 2020. [arXiv:2003.03052](https://arxiv.org/abs/2003.03052).
- 7 Christian Cachin, Rachid Guerraoui, and Luís Rodrigues. *Introduction to Reliable and Secure Distributed Programming*. Springer, 2 edition, February 2011.
- 8 Christian Cachin, Giuliano Losa, and Luca Zanolini. Quorum Systems in Permissionless Networks. In Eshcar Hillel, Roberto Palmieri, and Etienne Rivière, editors, *26th International Conference on Principles of Distributed Systems (OPODIS 2022)*, volume 253 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:22, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.OPODIS.2022.17.
- 9 Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, pages 173–186, USA, 1999. USENIX Association. URL: <https://dl.acm.org/citation.cfm?id=296824>.
- 10 Benjamin Y. Chan and Elaine Shi. Streamlet: Textbook streamlined blockchains. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, AFT '20, pages 1–11, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3419614.3423256.
- 11 David C Clark. *Applications of finite geometries to designs and codes*. Michigan Technological University, 2012.
- 12 Charles J Colbourn. *CRC handbook of combinatorial designs*. CRC press, 2010.
- 13 Charles J Colbourn, Jeffrey H Dinitz, and Douglas R Stinson. Quorum systems constructed from combinatorial designs. *Information and Computation*, 169(2):160–173, 2001. doi:10.1006/INCO.2001.3044.
- 14 Hector Garcia-Molina and Daniel Barbara. How to assign votes in a distributed system. *J. ACM*, 32(4):841–860, October 1985. doi:10.1145/4221.4223.
- 15 Álvaro García-Pérez and Alexey Gotsman. Federated Byzantine Quorum Systems. In Jiannong Cao, Faith Ellen, Luis Rodrigues, and Bernardo Ferreira, editors, *22nd International Conference on Principles of Distributed Systems (OPODIS 2018)*, volume 125 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:16, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.OPODIS.2018.17.
- 16 Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, pages 51–68, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3132747.3132757.
- 17 Maurice Herlihy. A quorum-consensus replication method for abstract data types. *ACM Trans. Comput. Syst.*, 4(1):32–53, February 1986. doi:10.1145/6306.6308.
- 18 Maurice Herlihy. Dynamic quorum adjustment for partitioned data. *ACM Trans. Database Syst.*, 12(2):170–194, June 1987. doi:10.1145/22952.22953.
- 19 Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 357–388, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-63688-7_12.
- 20 Akhil Kumar and Shun Yan Cheung. A high availability \sqrt{N} hierarchical grid algorithm for replicated data. *Information Processing Letters*, 40(6):311–316, 1991.
- 21 Jun Luo and Ying He. Geoquorum: Load balancing and energy efficient data access in wireless sensor networks. In *2011 Proceedings IEEE INFOCOM*, pages 616–620. IEEE, 2011. doi:10.1109/INFOCOM.2011.5935238.
- 22 Mamoru Maekawa. A \sqrt{N} algorithm for mutual exclusion in decentralized systems. *ACM Transactions on Computer Systems (TOCS)*, 3(2):145–159, 1985.
- 23 Dahlia Malkhi and Michael Reiter. Byzantine quorum systems. *Distributed computing*, 11(4):203–213, 1998. doi:10.1007/S004460050050.
- 24 Dahlia Malkhi, Michael Reiter, and Rebecca Wright. Probabilistic quorum systems. In *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '97, pages 267–273, New York, NY, USA, 1997. Association for Computing Machinery. doi:10.1145/259380.259458.

- 25 David Mazières. The stellar consensus protocol: A federated model for internet-level consensus. Technical report, StellarDevelopmentFoundation, 2016.
- 26 Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf>, 2008. Accessed: 2024-09-05.
- 27 M. Naor and A. Wool. The load, capacity and availability of quorum systems. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 214–225, 1994. doi:10.1109/SFCS.1994.365692.
- 28 Moni Naor and Udi Wieder. Scalable and dynamic quorum systems. *Distributed Computing*, 17(4):311–322, 2005. doi:10.1007/s00446-004-0114-3.
- 29 Joachim Neu, Ertem Nusret Tas, and David Tse. Ebb-and-flow protocols: A resolution of the availability-finality dilemma. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 446–465, 2021. doi:10.1109/SP40001.2021.00045.
- 30 Joachim Neu, Ertem Nusret Tas, and David Tse. Two more attacks on proof-of-stake ghost/ethereum. In *Proceedings of the 2022 ACM Workshop on Developments in Consensus*, pages 43–52, 2022. doi:10.1145/3560829.3563560.
- 31 David Peleg and Avishai Wool. The availability of quorum systems. *Information and Computation*, 123(2):210–223, 1995. doi:10.1006/INCO.1995.1169.
- 32 Sampath Rangarajan, Sanjeev Setia, and Satish K Tripathi. A fault-tolerant algorithm for replicated data management. *IEEE Transactions on parallel and distributed systems*, 6(12):1271–1282, 1995. doi:10.1109/71.476168.
- 33 Team Rocket, Maofan Yin, Kevin Sekniqi, Robbert van Renesse, and Emin Gün Sirer. Scalable and probabilistic leaderless BFT consensus through metastability. *CoRR*, abs/1906.08936, 2019. arXiv:1906.08936.
- 34 Rik Sarkar, Xianjin Zhu, and Jie Gao. Double rulings for information brokerage in sensor networks. In *Proceedings of the 12th annual international conference on mobile computing and networking*, pages 286–297, 2006. doi:10.1145/1161089.1161122.
- 35 Caspar Schwarz-Schilling, Joachim Neu, Barnabé Monnot, Aditya Asgaonkar, Ertem Nusret Tas, and David Tse. Three attacks on proof-of-stake ethereum. In *International Conference on Financial Cryptography and Data Security*, pages 560–576. Springer, 2022. doi:10.1007/978-3-031-18283-9_28.
- 36 Victor Shoup. Practical threshold signatures. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, pages 207–220, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. doi:10.1007/3-540-45539-6_15.
- 37 T. Srikanth and S. Toueg. Simulating authenticated broadcasts to derive simple fault-tolerant algorithms. *Distributed Computing*, 2(2):80–94, June 1987. doi:10.1007/BF01667080.
- 38 Alistair Stewart and Eleftherios Kokoris-Kogia. GRANDPA: a byzantine finality gadget. *CoRR*, abs/2007.01560, 2020. arXiv:2007.01560.
- 39 Dimitris Vyzovitis, Yusef Napora, Dirk McCormick, David Dias, and Yiannis Psaras. Gossipsub: Attack-resilient message propagation in the filecoin and eth2.0 networks, 2020. arXiv:2007.02754.
- 40 Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham. Hotstuff: Bft consensus with linearity and responsiveness. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, PODC '19, pages 347–356, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3293611.3331591.

A Projective Geometry Proofs

Proof of Proposition 12. Suppose $S, T \in \text{PG}_d(k, q)$. Then there exists $(d + 1)$ -dimensional vector subspaces $U, W \subseteq \mathbb{F}_q^{k+1}$ such that $S = \text{PG}(U)$ and $T = \text{PG}(W)$. It follows that $S \cap T = \text{PG}(U \cap W)$, so $S \cap T$ is indeed a projective subspace of $\text{PG}(k, q)$. Since $U + W$ is a subspace of \mathbb{F}_q^{k+1} , $U + W$ has dimension at most $k + 1$, which shows

8:18 Optimal Multilevel Slashing for Blockchains

$$\begin{aligned}\dim(S \cap T) &= \dim \text{PG}(U \cap W) = \dim(U \cap W) - 1 \\ &= \dim U + \dim W - \dim(U + W) - 1 \\ &\geq \dim U + \dim W - (k + 1) - 1 \\ &= 2(d + 1) - k - 2 = 2d - k.\end{aligned}$$

To show that this bound is sharp for some choice of S and T , suppose $2d \geq k$ and consider any basis v_1, \dots, v_{k+1} of \mathbb{F}_q^{k+1} . Let

$$U = \text{span}(v_1, \dots, v_{d+1}) \quad \text{and} \quad W = \text{span}(v_{k+1}, v_k, \dots, v_{k-d+1}).$$

Clearly both U and W are vector spaces of dimension $d + 1$. Also,

$$U \cap W = \text{span}(v_{k-d+1}, \dots, v_{d+1}),$$

which is a vector space of dimension $(d + 1) - (k - d + 1) + 1 = 2d - k + 1$. Hence, $\text{PG}(U), \text{PG}(W) \in \text{PG}_d(k, q)$ and $\text{PG}(U) \cap \text{PG}(W) = \text{PG}(U \cap W)$ is a projective subspace of $\text{PG}(k, q)$ with projective dimension $\dim(U \cap W) - 1 = 2d - k$, as desired. ◀

B Chernoff Bound

► **Lemma 25** (Chernoff Bound). *Let Z be a binomial random variable with parameters n and q . If $\delta > 0$, then*

$$\mathbf{P}(Z \geq (1 + \delta)nq) \leq \exp\left(-\frac{\delta^2 nq}{2 + \delta}\right).$$